Solarflare® Server Adapter User Guide

Information in this document is subject to change without notice.

SF-103837-CD

Last revised: December 2015

Issue 16

Table of Contents

1 Introduction

This is the User Guide for Solarflare® Server Adapters. This chapter covers the following topics:

NOTE: Throughout this guide the term Onload refers to both OpenOnload® and EnterpriseOnload® unless otherwise stated. Users of Onload should refer to the *Onload User Guide*, SF-104474-CD, which describes procedures for download and installation of the Onload distribution, accelerating and tuning the application using Onload to achieve minimum latency and maximum throughput.

1.1 Virtual NIC Interface

Solarflare's VNIC architecture provides the key to efficient server I/O and is flexible enough to be applied to multiple server deployment scenarios. These deployment scenarios include:

- **Kernel Driver** – This deployment uses an instance of a VNIC per CPU core for standard operating system drivers. This allows network processing to continue over multiple CPU cores in parallel. The virtual interface provides a performance-optimized path for the kernel TCP/IP stack and contention-free access from the driver, resulting in extremely low latency and reduced CPU utilization.
- **Accelerated Virtual I/O** – The second deployment scenario greatly improves I/O for virtualized platforms. The VNIC architecture can provide a VNIC per Virtual Machine, giving over a thousand protected interfaces to the host system, granting any virtualized (guest) operating system direct access to the network hardware. Solarflare's hybrid SR-IOV technology, unique to Solarflare

Ethernet controllers, is the only way to provide bare-metal I/O performance to virtualized guest operating systems whilst retaining the ability to live migrate virtual machines.

- **OpenOnload**™ – The third deployment scenario aims to leverage the host CPU(s) to full capacity, minimizing software overheads by using a VNIC per application to provide a kernel bypass solution. Solarflare has created both an open-source and Enterprise class high-performance application accelerator that delivers lower and more predictable latency and higher message rates for TCP and UDP-based applications, all with no need to modify applications or change the network infrastructure. To learn more about the open source OpenOnload project or EnterpriseOnload, download the Onload user guide (SF-104474-CD) or contact your reseller.

Advanced Features and Benefits

Virtual NIC support	The core of Solarflare technology. Protected VNIC interfaces can be instantiated for each running guest operating system or application, giving it a direct pipeline to the Ethernet network. This architecture provides the most efficient way to maximize network and CPU efficiency. The Solarflare Ethernet controller supports up to 1024 vNIC interfaces per port. On IBM System p servers equipped with Solarflare adapters, each adapter is assigned to a single Logical Partition (LPAR) where all VNICS are available to the LPAR.
PCI Express	Implements PCI Express 3.0.
High Performance	Support for 40G Ethernet interfaces and a new internal datapath micro architecture.
Hardware Switch Fabric	Full hardware switch fabric in silicon capable of steering any flow based on Layer 2, Layer 3 or application level protocols between physical and virtual interfaces. Supporting an open software defined network control plane with full PCI-IOV virtualization acceleration for high performance guest operating systems and virtual applications.
Improved flow processing	The addition of dedicated parsing, filtering, traffic shaping and flow steering engines which are capable of operating flexibly and with an optimal combination of a full hardware data plane with software based control plane.

TX PIO	Transmit Programmed input/output is the direct transfer of data to the adapter without CPU involvement. As an alternative to the usual bus master DMA method, TX PIO improves latency and is especially useful for smaller packets.
Multicast Replication	Received multicast packets are replicated in hardware and delivered to multiple receive queues.
Sideband management	NCSI RMII interface for base board management integration. SMBus interface for legacy base board management integration.
PCI Single-Root-IOV, SR-IOV, capable	16 Physical functions and up to 240 Virtual functions per adapter. Flexible deployment of 1024 channels between Virtual and Physical Functions. Support Alternate Routing ID (ARI). SR-IOV is not supported for Solarflare adapters on IBM System p servers.
10 Gigabit Ethernet	Supports the ability to design a cost effective, high performance 10 Gigabit Ethernet solution.
Receive Side Scaling (RSS)	IPv4 and IPv6 RSS raises the utilization levels of multi-core servers dramatically by distributing I/O load across all CPUs and cores.
Stateless offloads	Through the addition of hardware based TCP segmentation and reassembly offloads, VLAN, VxLAN and FCOE offloads.
Transmit rate pacing (per queue)	Provides a mechanism for enforcing bandwidth quotas across all guest operating systems. Software re-programmable on the fly to allow for adjustment as congestion increases on the network.
Jumbo frame support	Support for up to 9216 byte jumbo frames.
MSI-X support	2048 MSI-X interrupt support enables higher levels of performance. Can also work with MSI or legacy line based interrupts.
Ultra low latency	Cut through architecture. < 7µs end to end latency with standard kernel drivers, < 3µs with Onload drivers.

Remote boot	Support for PXE boot 2.1 and iSCSI Boot provides flexibility in cluster design and diskless servers (see Solarflare Boot ROM Agent on page 435). Network boot is not supported for Solarflare adapters on IBM System p servers.
MAC address filtering	Enables the hardware to steer packets based on the MAC address to a VNIC.
Hardware timestamps	The Solarflare Flareon™ SFN7000 series adapters can support hardware timestamping for all packets , sent and received - including PTP. The SFN5322F and SFN6322F adapters can generate hardware timestamps of PTP packets.

1.2 Product Specifications

Solarflare Flareon™ Network Adapters

Solarflare Flareon™ Ultra SFN7124F Quad-Port 10GbE SFP+ PCIe 3.0 Server I/O Adapter

Part number	SFN7124F
Controller silicon	SFC9140
Power	13W typical
PCI Express	8 lanes Gen 3 (8.0GT/s), x8 edge connector (usable in x8 and x16 slots)
PCIe features support	Per adapter: 16 PF, 240 VF, 2048 VI, 2048 MSI-X Interrupts
Supports OpenOnload	Yes (factory enabled)
PTP and hardware timestamps	Enabled by installing AppFlex license
1PPS	Optional bracket and cable assembly - not factory installed
SR-IOV	Yes
Network ports	4 x SFP+ (10G/1G)

Solarflare Flareon™ Ultra SFN7004F Quad-Port 10GbE SFP+ PCIe 3.0 Server I/O Adapter

Part number	SFN7004F
Controller silicon	SFC9140
Power	13W typical
PCI Express	8 lanes Gen 3 (8.0GT/s), x8 edge connector (usable in x8 and x16 slots)
PCIe features support	Per adapter: 16 PF, 240 VF, 2048 VI, 2048 MSI-X Interrupts
Supports OpenOnload	Enabled by installing AppFlex license
PTP and hardware timestamps	Enabled by installing AppFlex license
1PPS	No
SR-IOV	Yes
Network ports	4 x SFP+ (10G/1G)

Solarflare Flareon™ Ultra SFN7142Q Dual-Port 40GbE QSFP+ PCIe 3.0 Server I/O Adapter

Part number	SFN7142Q
Controller silicon	SFC9140
Power	13W typical
PCI Express	8 lanes Gen 3 (8.0GT/s), x8 edge connector (usable in x8 and x16 slots)
PCIe features support	Per adapter: 16 PF, 240 VF, 2048 VI, 2048 MSI-X Interrupts
Supports OpenOnload	Yes (factory enabled)
PTP and hardware timestamps	Enabled by installing AppFlex license
1PPS	Optional bracket and cable assembly - not factory installed
SR-IOV	Yes
Network ports	2 x QSFP+ (40G/10G)

Solarflare Flareon™ SFN7042Q Dual-Port 40GbE QSFP+ PCIe 3.0 Server I/O Adapter

Part number	SFN7042Q
Controller silicon	SFC9140
Power	13W typical
PCI Express	8 lanes Gen 3 (8.0GT/s), x8 edge connector (usable in x8 and x16 slots)
PCIe features support	Per adapter: 16 PF, 240 VF, 2048 VI, 2048 MSI-X Interrupts
Supports OpenOnload	Enabled by installing AppFlex license
PTP and hardware timestamps	Enabled by installing AppFlex license
1PPS	Optional bracket and cable assembly - not factory installed
SR-IOV	Yes
Network ports	2 x QSFP+ (40G/10G)

Solarflare Flareon™ Ultra SFN7322F Dual-Port 10GbE PCIe 3.0 Server I/O Adapter

Part number	SFN7322F
Controller silicon	SFC9120
Power	5.9W typical
PCI Express	8 lanes Gen 3 (8.0GT/s), x8 edge connector (usable in x8 and x16 slots)
PCIe features support	Per adapter: 16 PF, 240 VF, 2048 VI, 2048 MSI-X Interrupts
Supports OpenOnload	Yes (factory enabled)
PTP and hardware timestamps	Yes (factory enabled)
1PPS	Optional bracket and cable assembly - not factory installed
SR-IOV	Yes
Network ports	2 x SFP+ (10G/1G)

Solarflare Flareon™ Ultra SFN7122F Dual-Port 10GbE PCIe 3.0 Server I/O Adapter

Part number	SFN7122F
Controller silicon	SFC9120
Power	5.9W typical
PCI Express	8 lanes Gen 3 (8.0GT/s), x8 edge connector (usable in x8 and x16 slots)
PCIe features support	1Per adapter: 16 PF, 240 VF, 2048 VI, 2048 MSI-X Interrupts.
Supports OpenOnload	Yes (factory enabled)
PTP and hardware timestamps	AppFlex™ license required
1PPS	Optional bracket and cable assembly - not factory installed.
SR-IOV	Yes
Network ports	2 x SFP+ (10G/1G)

Solarflare Flareon™ SFN7002F Dual-Port 10GbE PCIe 3.0 Server I/O Adapter

Part number	SFN7002F
Controller silicon	SFC9120
Power	5.9W typical
PCI Express	8 lanes Gen 3 (8.0GT/s), x8 edge connector (usable in x8 and x16 slots)
PCIe features support	Per adapter: 16 PF, 240 VF, 2048 VI, 2048 MSI-X Interrupts.
Supports OpenOnload	AppFlex™ license required
PTP and hardware timestamps	AppFlex™ license required
1PPS	Optional bracket and cable assembly - not factory installed.
SR-IOV	Yes
Network ports	2 x SFP+ (10G/1G)

Solarflare Onload Network Adapters

Solarflare SFN5121T Dual-Port 10GBASE-T Server Adapter

Part number	SFN5121T
Controller silicon	SFL9021
Power	12.9W typical
PCI Express	8 lanes Gen2 (5.0GT/s), x8 edge connector (usable in x8 and x16 slots), 127 SR-IOV virtual functions per port
Virtual NIC support	1024 vNIC interfaces per port
Supports OpenOnload	Yes
SR-IOV	Yes
Network ports	2 x 10GBASE-T (10G/1G/100M)

Solarflare SFN5122F Dual-Port 10G SFP+ Server Adapter

Part number	SFN5122F
Controller silicon	SFC9020
Power	4.9W typical
PCI Express	8 lanes Gen2 (5.0GT/s), x8 edge connector (usable in x8 and x16 slots), 127 SR-IOV virtual functions per port
Virtual NIC support	1024 vNIC interfaces per port
Supports OpenOnload	Yes
SR-IOV	Yes
Network ports	2 x SFP+ (10G/1G)

Solarflare SFN6122F Dual-Port 10GbE SFP+ Server Adapter

Part number	SFN6122F
Controller silicon	SFC9020
Power	5.9W typical
PCI Express	8 lanes Gen2 (5.0GT/s), x8 edge connector (usable in x8 and x16 slots), 127 SR-IOV virtual functions per port
Virtual NIC support	1024 vNIC interfaces per port
Supports OpenOnload	Yes
SR-IOV	Yes[1]
Network ports	2 x SFP+ (10G/1G)
Regulatory Product Code	S6102

1. SR-IOV is not supported for Solarflare adapters on IBM System p servers.

Solarflare SFN6322F Dual-Port 10GbE SFP+ Server Adapter

Part number	SFN6122F
Controller silicon	SFC9020
Power	5.9W typical
PCI Express	8 lanes Gen2 (5.0GT/s), x8 edge connector (usable in x8 and x16 slots), 127 SR-IOV virtual functions per port
Virtual NIC support	1024 vNIC interfaces per port
Supports OpenOnload	Yes
SR-IOV	Yes
Network ports	2 x SFP+ (10G/1G)

Solarflare SFA6902F Dual-Port 10GbE SFP+ ApplicationOnload™ Engine

Part number	SFA6902F
Controller silicon	SFC9020
Power	25W typical
PCI Express	8 lanes Gen2 (5.0GT/s), x8 edge connector (usable in x8 and x16 slots), 127 SR-IOV virtual functions per port
Virtual NIC support	1024 vNIC interfaces per port
Supports OpenOnload	Yes
SR-IOV	Yes
Network ports	2 x SFP+ (10G/1G)

Solarflare Performant Network Adapters

Solarflare SFN5161T Dual-Port 10GBASE-T Server Adapter

Part number	SFN5161T
Controller silicon	SFL9021
Power	12.9W typical
PCI Express	8 lanes Gen2 (5.0GT/s), x8 edge connector (usable in x8 and x16 slots)
Virtual NIC support	1024 vNIC interfaces per port
Supports OpenOnload	No
SR-IOV	Yes
Network ports	2 x 10GBASE-T (10G/1G/100M)

Solarflare SFN5162F Dual-Port 10G SFP+ Server Adapter

Part number	SFN5162F
Controller silicon	SFC9020
Power	4.9W typical
PCI Express	8 lanes Gen2 (5.0GT/s), x8 edge connector (usable in x8 and x16 slots)
Virtual NIC support	1024 vNIC interfaces per port
Supports OpenOnload	No
SR-IOV	Yes[1]
Network ports	2 x SFP+ (10G/1G)

1. SR-IOV is not supported for Solarflare adapters on IBM System p servers.

Solarflare Mezzanine Adapters

Solarflare SFN5812H Dual-Port 10G Ethernet Mezzanine Adapter

Part number	SFN5812H
Controller silicon	SFC9020
Power	3.9W typical
PCI Express	8 lanes Gen2 (5.0GT/s), x8 edge connector (usable in x8 and x16 slots), 127 SR-IOV virtual functions per port
Virtual NIC support	1024 vNIC interfaces per port
Supports OpenOnload	Yes
SR-IOV	Yes
Ports	2 x 10GBASE-KX4 backplane transmission

Solarflare SFN5814H Quad-Port 10G Ethernet Mezzanine Adapter

Part number	SFN5814H
Controller silicon	2 x SFC9020
Power	7.9W typical
PCI Express	8 lanes Gen2 (5.0GT/s), x8 edge connector (usable in x8 and x16 slots), 127 SR-IOV virtual functions per port
Virtual NIC support	1024 vNIC interfaces per port
Supports OpenOnload	Yes
SR-IOV	Yes
Ports	4 x 10GBASE-KX4 backplane transmission

Solarflare SFN6832F Dual-Port 10GbE SFP+ Mezzanine Adapter

Part number	SFN6832F-C61 for DELL PowerEdge C6100 series SFN6832F-C62 for DELL PowerEdge C6200 series
Controller silicon	SFC9020
Power	5.9W typical
PCI Express	8 lanes Gen2 (5.0GT/s), x8 edge connector (usable in x8 and x16 slots), 127 SR-IOV virtual functions per port
Virtual NIC support	1024 vNIC interfaces per port
Supports OpenOnload	Yes
SR-IOV	Yes
Ports	2 x SFP+ (10G/1G)
Regulatory Product Code	S6930

Solarflare SFN6822F Dual-Port 10GbE SFP+ FlexibleLOM Onload Server Adapter

Part number	SFN6822F
Controller silicon	SFC9020
Power	5.9W typical
PCI Express	8 lanes Gen2 (5.0GT/s), x8 edge connector (usable in x8 and x16 slots), 127 SR-IOV virtual functions per port
Virtual NIC support	1024 vNIC interfaces per port
Supports OpenOnload	Yes
SR-IOV	Yes
Ports	2 x SFP+ (10G/1G)

1.3 Software Driver Support

The software driver is currently supported on the following distributions:

- Windows® Server 2008, R2 only.
- Windows® Server 2012, including R2.
- Red Hat Enterprise Linux 6 (6.5 or later)
- Red Hat Messaging Realtime and Grid 2 update 5
- Red Hat Enterprise Linux 7.x
- Red Hat Enterprise Linux for Realtime 7.x
- SUSE Linux Enterprise Server 11 (SP3 or later), and 12
- SUSE Linux Enterprise Real Time 11 (SP3 or later).
- Ubuntu 14.04 LTS, 14.10 and 15.04.
- Debian 7.x and 8.x.
- FreeBSD 10.x.
- VMware® ESX™ 5.0, ESXi™ 5.1, 5.5 and 6.0.
- Linux® KVM.
- Solaris™ 10 updates 8, 9 and 10 and Solaris™ 11 (GLDv3).
- Mac OS X Snow Leopard 10.6.8 (32 bit and 64 bit), OS X Lion 10.7.0 and later releases, OS X Mountain Lion 10.8.0 and later, OS X Mavericks 10.9.

Support includes all minor updates/releases/service packs of the above major releases, for which the distributor has not yet declared end of life/support.

Solarflare are not aware of any issues preventing building and installing the driver on other Linux variants that use kernel versions 2.6.18 - 4.1 inclusive.

Solarflare SFN5162F and SFN6122F adapters are supported on the IBM POWER architecture (PPC64) running RHEL 6.4 on IBM System p servers.

The Solarflare accelerated network middleware, OpenOnload and EnterpriseOnload, is supported on all Linux, Ubuntu, and Debian variants listed above, and is available for all Solarflare Onload network adapters. Solarflare are not aware of any issues preventing OpenOnload installation on other Linux variants such as Centos, Gentoo and Fedora.

1.4 Solarflare AppFlex™ Technology Licensing.

Solarflare AppFlex technology allows Solarflare server adapters to be selectively configured to enable on-board applications. AppFlex licenses are required to enable selected functionality on the Solarflare Flareon™ adapters and the AOE ApplicationOnload™ Engine.

Customers can obtain access to AppFlex applications via their Solarflare sales channel by obtaining the corresponding AppFlex authorization code. The authorization code allows the customer to generate licenses at the MyAppFlex page at https://support.solarflare.com/myappflex.

The sfkey utility application is used to install the generated license key file on selected adapters. For detailed instructions for sfkey and license installation refer to License Install with sfkey on page 90.

1.5 Open Source Licenses

Solarflare Boot Manager

The Solarflare Boot Manager is installed in the adapter's flash memory. This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

Controller Firmware

The firmware running on the SFC9xxx controller includes a modified version of libcoroutine. This software is free software published under a BSD license reproduced below:

Copyright (c) 2002, 2003 Steve Dekorte

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the author nor the names of other contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.6 Support and Download

Solarflare network drivers, RPM packages and documentation are available for download from https://support.solarflare.com/.

Software and documentation for OpenOnload is available from www.openonload.org.

1.7 Regulatory Information

Warnings

D*o not* install the Solarflare network adapter in hazardous areas where highly combustible or explosive products are stored or used without taking additional safety precautions. Do not expose the Solarflare network adapter to rain or moisture.

The Solarflare network adapter is a Class III SELV product intended only to be powered by a certified limited power source.

The equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. The equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation.

If the equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Changes or modifications not expressly approved by Solarflare Communications, the party responsible for FCC compliance, could void the user's authority to operate the equipment.

This Class B digital apparatus complies with Canadian ICES-003.

Cet appareil numérique de la classe B est conforme à la norme NMB-003 du Canada.

Underwriters Laboratory Inc ('UL') has not tested the performance or reliability of the security or signaling aspects of this product. UL has only tested for fire, shock or casualty hazards as outlined in the UL's Standard for Safety UL 60950-1. ***UL Certification does not cover the performance or reliability of the security or signaling aspects of this product. UL makes no representations, warranties or certifications whatsoever regarding the performance or reliability of any security or signaling related functions of this product.***

Laser Devices

The laser safety of the equipment has been verified using the following certified laser device module (LDM):

Manufacturer	Model	CDRH Accession No	Mark of conformity	File No
Avago Technologies	AFBR-703SDZ	9720151-072	TUV	R72071411
Finisar Corporation	FTLX8571D3BCL	9210176-094	TUV	R72080250

When installed in a 10Gb Ethernet network interface card from the Solarflare SFN5000, SFN6000 or SFN7000 SERIES, the laser emission levels remain under Class I limits as specified in the FDA regulations for lasers, 21 CFR Part 1040.

The decision on what LDMs to use is made by the installer. For example, equipment may use one of a multiple of different LDMs depending on path length of the laser communication signal. This equipment is not basic consumer ITE.

The equipment is installed and maintained by qualified staff from the end user communications company or subcontractor of the end user organization. The end product user and/or installer are solely responsible for ensuring that the correct devices are utilized in the equipment and the equipment with LDMs installed complies with applicable laser safety requirements.

1.8 Regulatory Approval

The information in this section is applicable to SFN5121T, and SFN5162F Solarflare network adapters:

Category	Specification	Details
EMC	Europe	BS EN 55022:2006
		BS EN 55024:1998 +A1:2001 +A2:2003
	US	FCC Part 15 Class B
	Canada	ICES 003/NMB-003 Class B
Safety[1]	Europe	BS EN 60950-1:2006 +A11:2009
	US	UL 60950-1 2nd Ed.
	Canada	CSA C22.2 60950-1-07 2nd Ed.
	CB	IEC 60950-1:2005 2nd Ed.
RoHS	Europe	Complies with EU directive 2002/95/EC

1. The safety assessment has been concluded on this product as a component/sub-assembly only.

Additional Regulatory Information for SFN5122F, SFN6122F, SFN6322F, SFA6902F, SFN7002F, SFN7122F, SFN7322F, SFN7042Q and SFN7142Q adapters

これは情報処理装置等電波障害自主規制協議会（VCCI）の標準に基づくクラスＡ情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。そのような障害が発生した際、使用者は適切な対応が必要となる場合があります

警告使用者：

這是甲類的資訊產品，在居住的環境中使用時，可能會造成射頻 干擾，在這種情況下，使用者會被要求採取某些適當的對策

A 급 기기 (업무용 방송통신기기): 이 기기는 업무용 (A 급) 으로 전자파적합등록을 한 기기이오니 판매자 또는 사용자는 이 점을 주의하시기 바라며 , 가정외의 지역에서 사용하는 것을 목적으로 합니다

Category	Specification	Details
EMC	Europe	BS EN 55022:2010 + A1:2007
		BS EN 55024:1998 +A1:2001 +A2:2003
	US	FCC Part 15 Class B
	Canada	ICES 003/NMB-003 Class B
	Taiwan	CNS 13438:2006 Class B
	Japan	VCCI Regulations V-3:2010 Class B
	South Korea	KCC KN-22, KN-24
	Australia	AS/NZS CISPR 22:2009
Safety[1]	Europe	BS EN 60950-1:2006 +A11:2009
	US	UL 60950-1 2nd Ed.
	Canada	CSA C22.2 60950-1-07 2nd Ed.
	CB	IEC 60950-1:2005 2nd Ed.
RoHS	Europe	Complies with EU directive 2011/65/EU

1. The safety assessment has been concluded on this product as a component/sub-assembly only.

Additional Regulatory Information for SFN7004F and SFN7124F adapters

この装置は、クラス B 情報技術装置です。　この装置は、家庭環境で使用することを目的としていますが、この装置がラジオやテレビジョン受信機に近接して使用されると、受信障害を引き起こすことがあります。取扱説明書に従って正しい取り扱いをして下さい。　VCCI-B

警告用戶：

這是一個 B 類 產品，在居住環境中使用時可能會導致無線電干擾，在這種情況下，用戶可能需要採取適當的措施

B 급 기기 (가정용 방송통신기기): 이 기기는 가정용 (B 급) 으로 전자파적합등록을 한 기기로서 주로 가정에서 사용하는 것을 목적으로 하며 , 모든 지역에서 사용할 수 있습니다 .

Category	Specification	Details
EMC	Europe	BS EN 55022:2010, 55032:2012
		BS EN 55024:2010
	US	FCC CFR 47 Part 15 Class B
	Canada	ICES 003/NMB-003 Class B
	Taiwan	CNS 13438:2006 Class B
	Japan	VCCI Regulations V-3:2014.04 Class B
	South Korea	KCC KN-32, KN-24
	Australia	AS/NZS CISPR 22:2009 +A1:2010
Safety[1]	Europe	BS EN 60950-1:2006 +A12:2011
	US	UL 60950-1 2nd Ed.
	Canada	CSA C22.2 60950-1-07 2nd Ed.
	CB	IEC 60950-1:2005 2nd Ed.+AMI:2009
RoHS	Europe	Complies with EU directive 2011/65/EU

1. The safety assessment has been concluded on this product as a component/sub-assembly only.

Additional Regulatory Information for SFN5812H, SFN5814H and SFN6832F adapters

これは情報処理装置等電波障害自主規制協議会（VCCI）の標準に基づくクラスＡ情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。そのような障害が発生した際、使用者は適切な対応が必要となる場合があります

警告使用者：

這是甲類的資訊產品，在居住的環境中使用時，可能會造成射頻 干擾，在這種情況下，使用者會被要求採取某些適當的對策

Category	Specification	Details
EMC	Europe	BS EN 55022:2006
		BS EN 55024:1998 +A1:2001 +A2:2003
	US	FCC Part 15 Class B
	Canada	ICES 003/NMB-003 Class B
	Taiwan	CNS 13438:2006 Class A
	Japan	VCCI Regulations V-3:2010 Class A
	Australia	AS/NZS CISPR 22:2009
Safety[1]	Europe	BS EN 60950-1:2006 +A11:2009
	US	UL 60950-1 2nd Ed.
	Canada	CSA C22.2 60950-1-07 2nd Ed.
	CB	IEC 60950-1:2005 2nd Ed.
RoHS	Europe	Complies with EU directive 2002/95/EC

1. The safety assessment has been concluded on this product as a component/sub-assembly only.

2 Installation

This chapter covers the following topics:

CAUTION: Servers contain high voltage electrical components. Before removing the server cover, disconnect the mains power supply to avoid the risk of electrocution.

CAUTION: Static electricity can damage computer components. Before handling computer components, discharge static electricity from yourself by touching a metal surface, or wear a correctly fitted anti-static wrist band.

2.1 Solarflare Network Adapter Products

Solarflare Flareon™ adapters

- Solarflare Flareon Ultra SFN7024F Quad-Port 10GbE PCIe 3.0 SFP+ Server Adapter
- Solarflare Flareon SFN7004F Quad-Port 10GbE PCIe 3.0 SFP+ Server Adapter
- Solarflare Flareon Ultra SFN7142Q Dual-Port 40GbE PCIe 3.0 QSFP+ Server Adapter
- Solarflare Flareon SFN7042Q Dual-Port 40GbE PCIe 3.0 QSFP+ Server Adapter
- Solarflare Flareon Ultra SFN7322F Dual-Port 10GbE PCIe 3.0 Server I/O Adapter
- Solarflare Flareon Ultra SFN7122F Dual-Port 10GbE PCIe 3.0 Server I/O Adapter
- Solarflare Flareon SFN7002F Dual-Port 10GbE PCIe 3.0 Server I/O Adapter

Solarflare Onload adapters

- Solarflare SFN6322F Dual-Port 10GbE Precision Time Stamping Server Adapter
- Solarflare SFN6122F Dual-Port 10GbE SFP+ Server Adapter
- Solarflare SFA6902F Dual-Port 10GbE ApplicationOnload™ Engine
- Solarflare SFN5122F Dual-Port 10G SFP+ Server Adapter
- Solarflare SFN5121T Dual-Port 10GBASE-T Server Adapter

Solarflare Performant network adapters

- Solarflare SFN5161T Dual-Port 10GBASE-T Server Adapter
- Solarflare SFN5162F Dual-Port 10G SFP+ Server Adapter

Solarflare Mezzanine adapters

- Solarflare SFN5812H Dual-Port 10G Ethernet Mezzanine Adapter for IBM BladeCenter
- Solarflare SFN5814H Quad-Port 10G Ethernet Mezzanine Adapter for IBM BladeCenter
- Solarflare SFN6832F-C61 Dual-Port 10GbE SFP+ Mezzanine Adapter for DELL PowerEdge C6100 series servers.
- Solarflare SFN6832F-C62 Dual-Port 10GbE SFP+ Mezzanine Adapter for DELL PowerEdge C6200 series servers.
- Solarflare SFN6822F Dual-Port 10GbE SFP+ FlexibleLOM Onload Server Adapter

Solarflare network adapters can be installed on Intel/AMD x86 based 32 bit or 64 bit servers. The network adapter must be inserted into a PCIe x8 OR PCIe x 16 slot for maximum performance. Refer to PCI Express Lane Configurations on page 233 for details.

Solarflare SFN5162F and SFN6122F adapters are supported on the IBM POWER architecture (PPC64) running RHEL 6.4 on IBM System p servers.

2.2 Fitting a Full Height Bracket (optional)

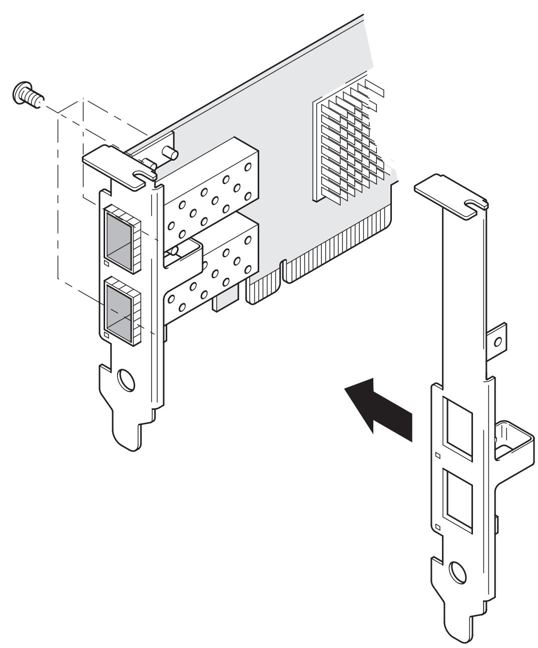
Solarflare adapters are supplied with a low-profile bracket fitted to the adapter. A full height bracket has also been supplied for PCIe slots that require this type of bracket.

To fit a full height bracket to the Solarflare adapter:

1 From the back of the adapter, remove the screws securing the bracket.

2 Slide the bracket away from the adapter.

3 Taking care not the overtighten the screws, attach the full height bracket to the adapter.



2.3 Inserting the Adapter in a PCI Express (PCIe) Slot

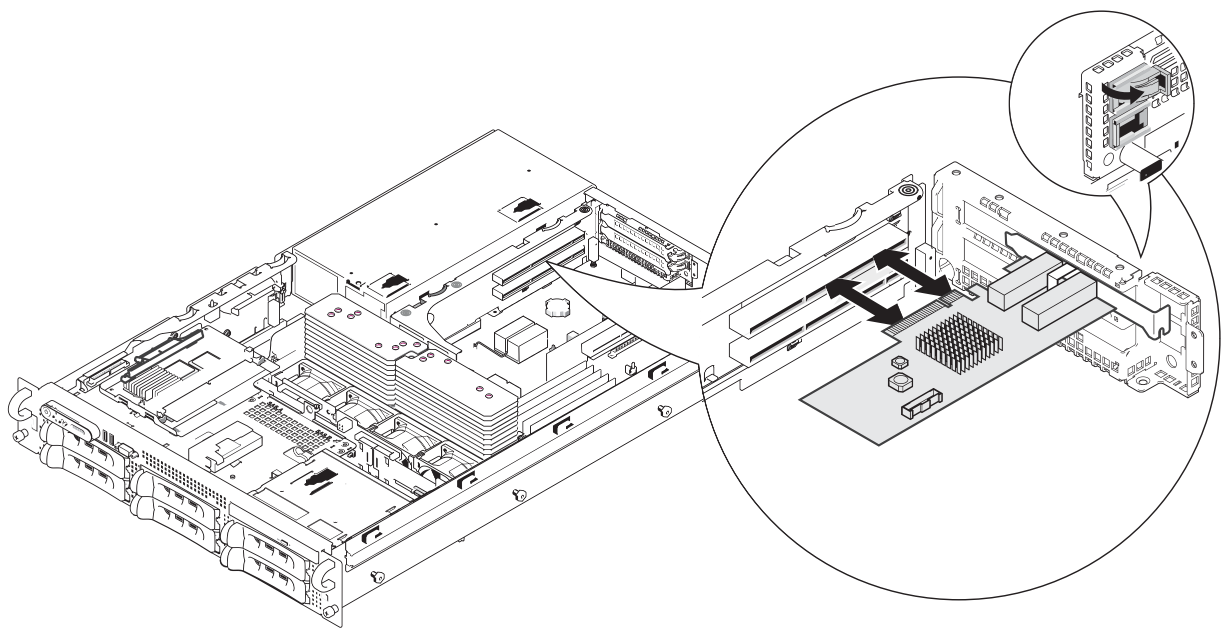
To insert the adapter in a PCI Express (PCIe) slot:

1 Shut down the server and unplug it from the mains. Remove the server cover to access the PCIe slots in the server.

2 Locate an 8-lane or 16-lane PCIe slot (refer to the server manual if necessary) and insert the Solarflare card.

3 Secure the adapter bracket in the slot.

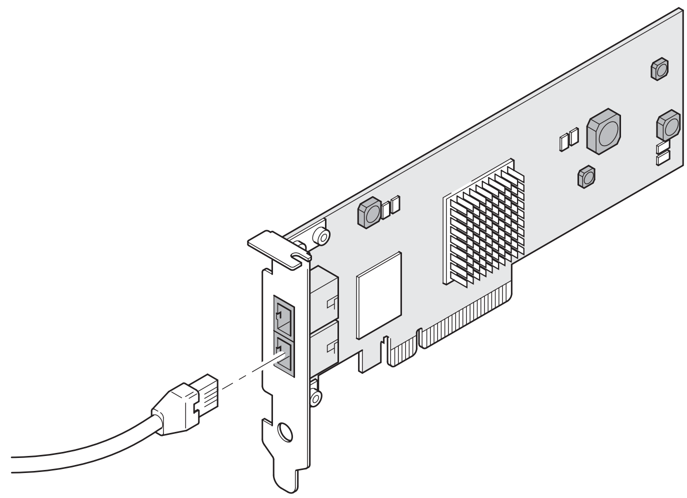
4 Replace the cover and restart the server.



5 After restarting the server, the host operating system may prompt you to install drivers for the new hardware. Click Cancel or abort the installation and refer to the relevant chapter in this manual for how to install the Solarflare adapter drivers for your operating system.

2.4 Attaching a Cable (RJ-45)

Solarflare 10GBASE-T Server Adapters connect to the Ethernet network using a copper cable fitted with an RJ-45 connector (shown below).



RJ-45 Cable Specifications

Table 1 below lists the recommended cable specifications for various Ethernet port types. Depending on the intended use, attach a suitable cable. For example, to achieve 10 Gb/s performance, use a Category 6 cable. To achieve the desired performance, the adapter must be connected to a compliant link partner, such as an IEEE 802.3an-compliant gigabit switch.

Table 1: RJ-45 Cable Specification

Port type	Connector	Media Type	Maximum Distance
10GBASE-T	RJ-45	Category 6A	100m (328 ft.)
		Category 6 unshielded twisted pairs (UTP)	55m (180 ft.)
		Category 5E	55m (180 ft.)
1000BASE-T	RJ-45	Category 5E, 6, 6A UTP	100m (328 ft.)
100BASE-TX	RJ-45	Category 5E, 6, 6A UTP	100m (328 ft.)

2.5 Attaching a Cable (SFP+)

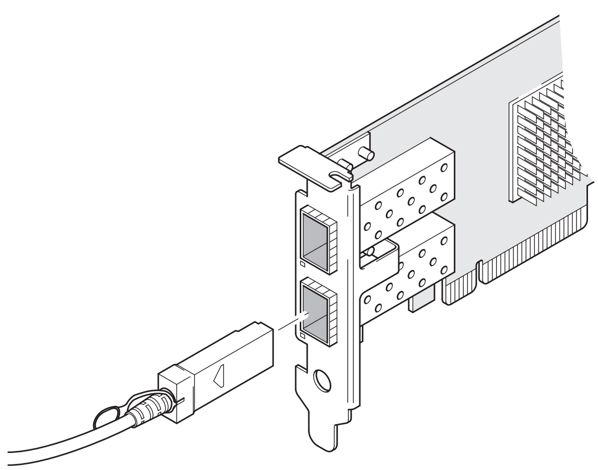
Solarflare SFP+ Server Adapters can be connected to the network using either an SFP+ Direct Attach cable or a fiber optic cable.

Attaching an SFP+ Direct Attach Cable

To attach an SFP+ Direct Attach cable:

1 Turn the cable so that the connector retention tab and gold fingers are on the same side as the network adapter retention clip.

2 Push the cable connector straight in to the adapter socket until it clicks into place.



Removing an SFP+ Direct Attach Cable

To remove an SFP+ Direct Attach cable:

1 Pull straight back on the release ring to release the cable retention tab. Alternatively, you can lift the retention clip on the adapter to free the cable if necessary.

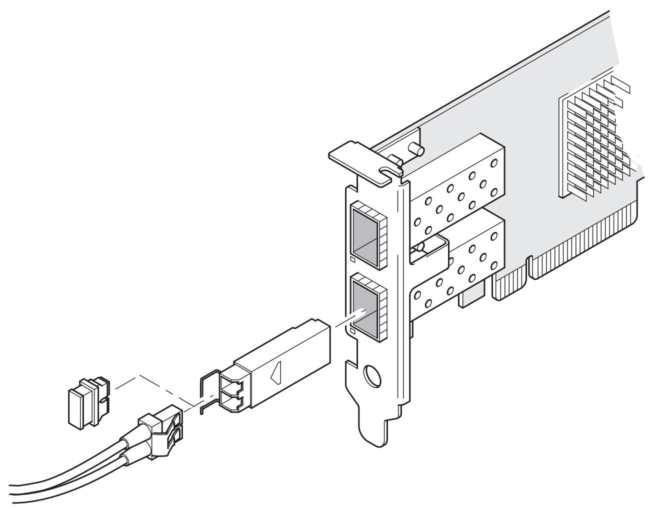
2 Slide the cable free from the adapter socket.

Attaching a fiber optic cable

WARNING: *Do not* look directly into the fiber transceiver or cables as the laser beams can damage your eyesight.

To attach a fiber optic cable:

1. Remove and save the fiber optic connector cover.
2. 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).



Removing a fiber optic cable

WARNING: *Do not* look directly into the fiber transceiver or cables as the laser beams can damage your eyesight.

To remove a fiber optic cable:

1. Remove the cable from the adapter bracket and replace the fiber optic connector cover.
2. Pull the plastic or wire tab to release the adapter bracket.
3. Hold the main body of the adapter bracket and remove it from the adapter.

2.6 Supported SFP+ Cables

Table 2 is a list of supported SFP+ cables that have been tested by Solarflare. Solarflare is not aware of any issues preventing the use of other brands of SFP+ cables (of up to 5m in length) with Solarflare network adapters. However, only cables in the table below have been fully verified and are therefore supported.

Table 2: Supported SFP+ Direct Attach Cables

Manufacturer	Product Code	Cable Length	Notes
Arista	CAB-SFP-SFP-1M	1m	
Arista	CAB-SFP-SFP-3M	3m	
Cisco	SFP-H10GB-CU1M	1m	
Cisco	SFP-H10GB-CU3M	3m	
Cisco	SFP-H10GB-CU5M	5m	
HP	J9283A/B Procurve	3m	
Juniper	EX-SFP-10GE-DAC-1m	1m	
Juniper	EX-SFP-10GE-DAC-3m	3m	
Molex	74752-1101	1m	
Molex	74752-2301	3m	
Molex	74752-3501	5m	
Molex	74752-9093	1m	37-0960-01 / 0K585N
Molex	74752-9094	3m	37-0961-01 / 0J564N
Molex	74752-9096	5m	37-0962-01 / 0H603N
Panduit	PSF1PXA1M	1m	
Panduit	PSF1PXA3M	3m	
Panduit	PSF1PXD5MBU	5m	
Siemon	SFPP30-01	1m	
Siemon	SFPP30-02	2m	
Siemon	SFPP30-03	3m	
Siemon	SFPP24-05	5m	
Tyco	2032237-2 D	1m	
Tyco	2032237-4	3m	

The Solarflare SFA6902F adapter has been tested and certified with direct attach cables up to 3m in length.

2.7 Supported SFP+ 10G SR Optical Transceivers

Table 3 is a list of supported SFP+10G SR optical transceivers that have been tested by Solarflare. Solarflare is not aware of any issues preventing the use of other brands of 10G SR transceivers with Solarflare network adapters. However, only transceivers in the table below have been fully verified and are therefore supported.

Table 3: Supported SFP+ 10G Optical SR Transceivers

Manufacturer	Product Code	Notes
Avago	AFBR-703SDZ	10G
Avago	AFBR-703SDDZ	Dual speed 1G/10G optic.
Avago	AFBR-703SMZ	10G
Arista	SFP-10G-SR	10G
Finisar	FTLX8571D3BCL	10G
Finisar	FTLX8571D3BCV	Dual speed 1G/10G optic.
HP	456096-001	Also labeled as 455883-B21 and 455885-001
Intel	AFBR-703SDZ	10G
JDSU	PLRXPL-SC-S43-22-N	10G
Juniper	AFBR-700SDZ-JU1	10G
MergeOptics	TRX10GVP2010	10G
Solarflare	SFM-10G-SR	10G
Vorboss	VBO-PXG-SR-300	10G

2.8 Supported SFP+ 10G LR Optical Transceivers

Table 4 is a list of supported SFP+10G LR optical transceivers that have been tested by Solarflare. Solarflare is not aware of any issues preventing the use of other brands of 10G LR transceivers with Solarflare network adapters. However, only transceivers in the table below have been fully verified and are therefore supported.

Table 4: Supported SFP+ 10G LR Optical Transceivers

Manufacturer	Product Code	Notes
Avago	AFCT-701SDZ	10G single mode fiber
Finisar	FTLX1471D3BCL	10G single mode fiber

2.9 QSFP+ Transceivers and Cables

The following tables identify QSFP+ transceiver modules and cables tested by Solarflare with the SFN7000 QSP+ adapters. Solarflare are not aware of any issues preventing the use of other brands of QSFP+ 40G transceivers and cables with Solarflare SFN7000 QSFP+ adapters. However, only products listed in the tables below have been fully verified and are therefore supported

Supported QSFP+ 40GBASE-SR4 Transceivers

The Solarflare Flareon Ultra SFN7000Q QSFP+ adapters have been tested with the following QSFP+ 40GBASE-SR4 optical transceiver modules.

Table 5: Supported QSFP+ SR4 Transceivers

Manufacturer	Product Code	Notes
Arista	AFBR-79E4Z	Standard 100m (OM3 Multimode fiber) range.
Avago	AFBR-79EADZ	
Avago	AFBR-79EIDZ	
Avago	AFBR-79EQDZ	
Avago	AFBR-79EQPZ	
Finisar	FTL410QE2C	
JDSU	JQP-04SWAA1	
JDSU	JDSU-04SRAB1	
Solarflare	SFM-40G-SR4	

Supported QSFP+ 40G Active Optical Cables (AOC)

The Solarflare Flareon Ultra SFN7000Q QSFP+ adapters have been tested with the following QSFP+ Active Optical Cables (AOC).

Table 6: Supported QSFP+ Active Optical Cables

Manufacturer	Product Code	Notes
Avago	AFBR-7QER05Z	3m
Finisar	FCBG410QB1C03	3m
Finisar	FCBN410QB1C05	5m

Supported QSFP+ 40G Direct Attach Cables

The Solarflare Flareon Ultra SFN7000Q QSFP+ adapters have been tested with the following QSFP+ Direct Attach Cables (DAC). QSFP cables may not work with all switches.

Table 7: Supported QSFP+ Direct Attach Cables

Manufacturer	Product Code	Notes
Arista	CAB-Q-Q-3M	3m
Arista	CAB-Q-Q-5M	5m
FCI	10093084-3030LF	3m
Molex	74757-1101	1m QSFP cable
Molex	74757-2301	3m QSFP cable
Siemon	QSFP30-01	1m
Siemon	QSFP30-03	3m
Siemon	QSFP26-05	5m

Supported QSFP+ to SFP+ Breakout Cables

Solarflare QSFP+ to SFP+ breakout cables enable users to connect Solarflare SFN7142Q, SFN7042Q dual-port QSFP+ server I/O adapters to work as a quad-port SFP+ server I/O adapters. The breakout cables offer a cost-effective option to support connectivity flexibility in high-speed data center applications.

These high performance direct-attach assemblies support 2 lanes of 10 Gb/s per QSFP+ port and are available in lengths of 1 meters and 3 meters. The SOLR-QSFP2SFP-1M, -3M copper DAC cables are fully tested and compatible with the Solarflare SFN7142Q server I/O adapter. These cables are compliant with the SFF-8431, SFF-8432, SFF-8436, SFF-8472 and IBTA Volume 2 Revision 1.3 specifications.

Table 8: Supported QSFP+ to SFP+ Breakout Cables

Manufacturer	Product Code	Notes
Solarflare	SOLR-QSFP2SFP-1M	
Solarflare	SOLR-QSFP2SFP-3M	

2.10 Supported SFP 1000BASE-T Transceivers

Table 9 is a list of supported SFP 1000BASE-T transceivers that have been tested by Solarflare. Solarflare is not aware of any issues preventing the use of other brands of 1000BASE-T transceivers with the Solarflare network adapters. However, only transceivers in the table below have been fully verified and are therefore supported.

Table 9: Supported SFP 1000BASE-T Transceivers

Manufacturer	Product Code	Notes
Arista	SFP-1G-BT	
Avago	ABCU-5710RZ	
Cisco	30-1410-03	
Dell	FCMJ-8521-3-(DL)	
Finisar	FCLF-8521-3	
Finisar	FCMJ-8521-3	
HP	453156-001 453154-B21	
3COM	3CSFP93	

2.11 Supported 1G Optical Transceivers

Table 10 is a list of supported 1G transceivers that have been tested by Solarflare. Solarflare is not aware of any issues preventing the use of other brands of 1G transceivers with Solarflare network adapters. However, only transceivers in the table below have been fully verified and are therefore supported.

Table 10: Supported 1G Transceivers

Manufacturer	Product Code	Type
Avago	AFBR-5710PZ	1000Base-SX
Cisco	GLC-LH-SM	1000Base-LX/LH
Finisar	FTLF8519P2BCL	1000Base-SX
Finisar	FTLF8519P3BNL	1000Base-SX
Finisar	FTLF1318P2BCL	1000Base-LX
Finisar	FTLF1318P3BTL	1000Base-LX
HP	453153-001 453151-B21	1000Base-SX

2.12 Supported Speed and Mode

Solarflare network adapters support either QSFP+, SFP, SFP+ or Base-T standards.

On Base-T adapters three speeds are supported 100Mbps, 1Gbps and 10Gbps. The adapters use auto negotiation to automatically select the highest speed supported in common with the link partner.

On SFP+ adapters the currently inserted SFP module (transceiver) determines the supported speeds, typically SFP modules only support a single speed. Some Solarflare SFP+ adapters support dual speed optical modules that can operate at either 1Gbps or 10Gbps. However, these modules do not auto-negotiate link speed and operate at the maximum (10G) link speed unless explicitly configured to operate at a lower speed (1G).

The tables below summarizes the speeds supported by Solarflare network adapters.

Table 11: SFN5xxx,SFN6xxx and SFN7xxx SFP+ QSFP+ Adapters

Supported Modes	Auto neg speed	Speed	Comment
QSFP+ direct attach cables	No	10G or 40G	SFN7142Q, SFN7042Q
QSFP+ optical cables	No	10G or 40G	SFN7142Q, SFN7042Q
SFP+ direct attach cable	No	10G	
SFP+ optical module (10G)	No	10G	
SFP optical module (1G)	No	1G	
SFP+ optical module (10G/1G)	No	10G or 1G	Dual speed modules run at the maximum speed (10G) unless explicitly configured to the lower speed (1G)
SFP 1000BASE-T module	No	1G	These modules support only 1G and will not link up at 100Mbps

Table 12: SFN5121T, SFN5151T, SFN5161T 10GBASE-T Adapters

Supported Modes	Auto neg speed	Speed	Comment
100Base-T	Yes	100Mbps	Typically the interface is set to auto negotiation speed and automatically selects the highest speed supported in common with its link partner. If the link partner is set to 100Mbps, with no autoneg, the adapter will use “parallel detection” to detect and select 100Mbps speed. If needed any of the three speeds can be explicitly configured
1000Base-TX	Yes	1Gbps	
10GBase-T	Yes	10Gbps	

100Base-T in a Solarflare adapter back-to-back (no intervening switch) configuration will not work and is not supported.

2.13 LED States

There are two LEDs on the Solarflare network adapter transceiver module. LED states are as follows

Table 13: LED States

Adapter Type	LED Description	State
QSFP+, SFP/SFP+	Speed	Green (solid) at all speeds
	Activity	Flashing green when network traffic is present LEDs are OFF when there is no link present
BASE-T	Speed	Green (solid) 10Gbps Yellow (solid) 100/1000Mbps
	Activity	Flashing green when network traffic is present LEDs are OFF when there is no link present

2.14 Configure QSFP+ Adapter

QSFP+ adapters can operate as 2 x 10Gbps per QSFP+ port or as 1 x 40Gbps per QSFP+ port. A configuration of 1 x 40G and 2 x 10G ports is not supported.

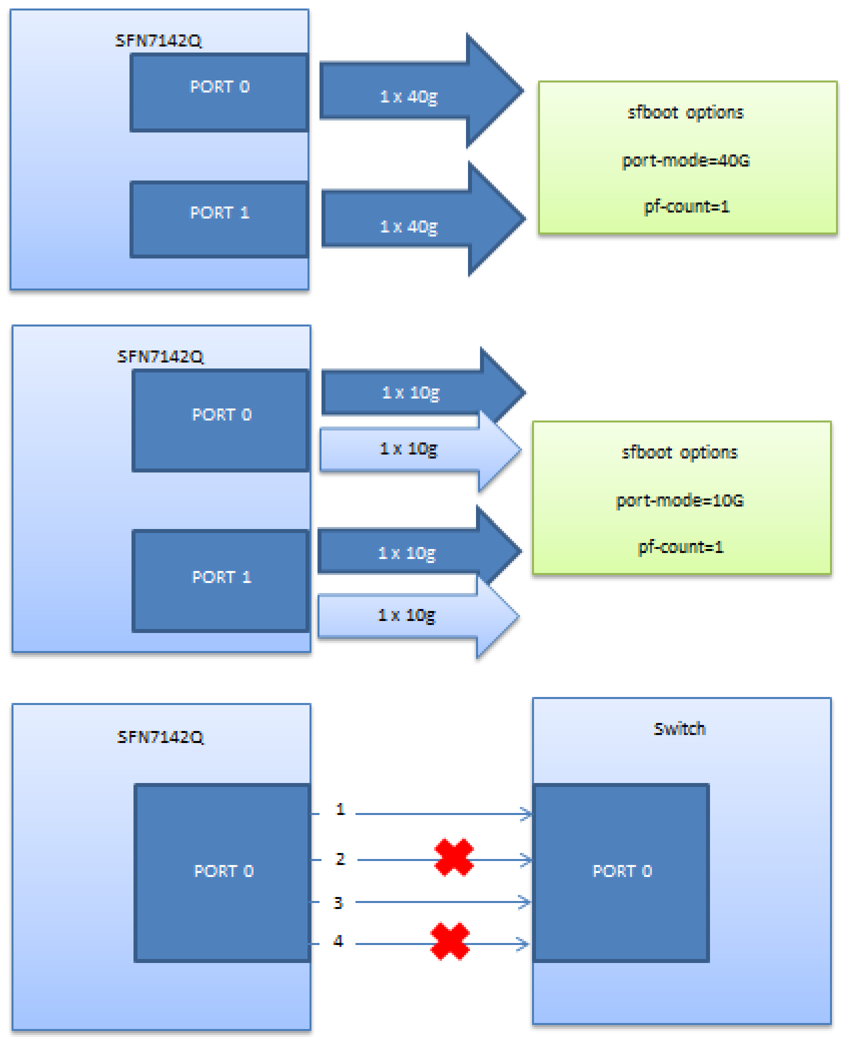


Figure 1: QSFP+ Port Configuration

The Solarflare 40G breakout cables have only 2 physical cables - for details refer to Supported QSFP+ to SFP+ Breakout Cables on page 32. Breakout cables from other suppliers may have 4 physical cables. When connecting a third party breakout cable into the Solarflare 40G QSFP+ cage (in 10G mode), only cables 1 and 3 will be active.

The sfboot utility from the Solarflare Linux Utilities package (SF-107601-LS) is used to configure the adapter for 10G or 40G operation.

```
# sfboot port-mode=2x40G
```

2.15 Single Optical Fiber - RX Configuration

The Solarflare adapter will support a receive (RX) only fiber cable configuration when the adapter is required only to receive traffic, but have no transmit link. This can be used, for example, when the adapter is to receive traffic from a fiber tap device.

Solarflare have successfully tested this configuration on a 10G link on SFN5000, SFN6000 and SFN7000 series adapters when the link partner is configured to be TX only (this will always be the case with a fiber tap). Some experimentation might be required when splitting the light signal to achieve a ratio that will deliver sufficient signal strength to all endpoints.

Solarflare adapters do not support a receive only configuration on 1G links.

2.16 Solarflare Mezzanine Adapters: SFN5812H and SFN5814H

The Solarflare SFN5812H Dual-Port and SFN5814H Quad-Port are 10G Ethernet Mezzanine Adapters for the IBM BladeCenter.

Solarflare mezzanine adapters are supported on the IBM BladeCenter E, H and S chassis, HS22, HS22V and HX5 servers. The IBM BladeCenter blade supports a single Solarflare mezzanine adapter.

1 The blade should be extracted from the BladeCenter in order to install the mezzanine adapter.

2 Remove the blade top cover and locate the two retaining posts towards the rear of the blade - (Figure 2). Refer to the BladeCenter manual if necessary

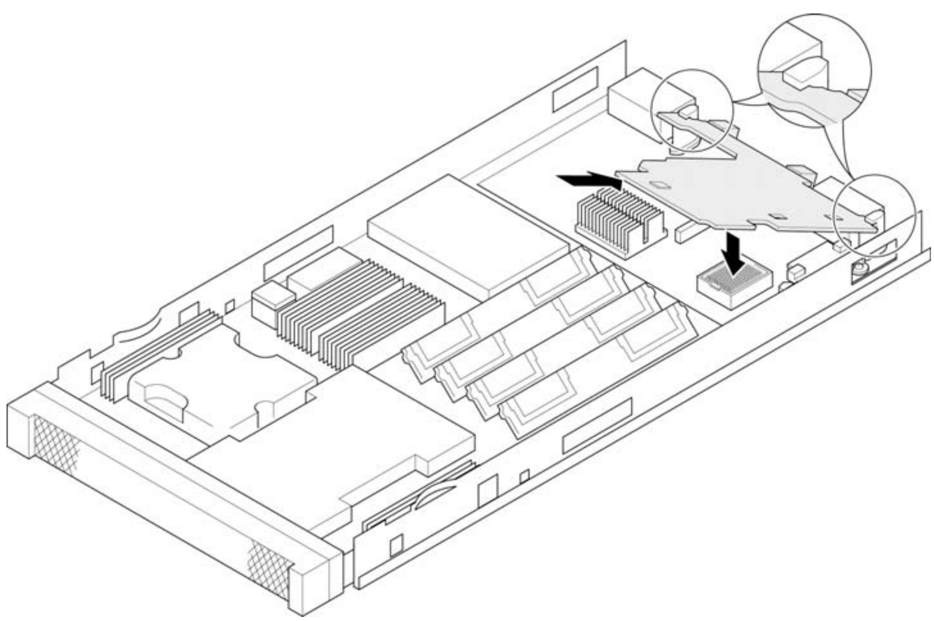


Figure 2: Installing the Mezzanine Adapter

3 Hinge the adapter under the retaining posts, as illustrated, and align the mezzanine port connector with the backplane connector block.

4 Lower the adapter, taking care to align the side positioning/retaining posts with the recesses in the adapter. See Figure 3.

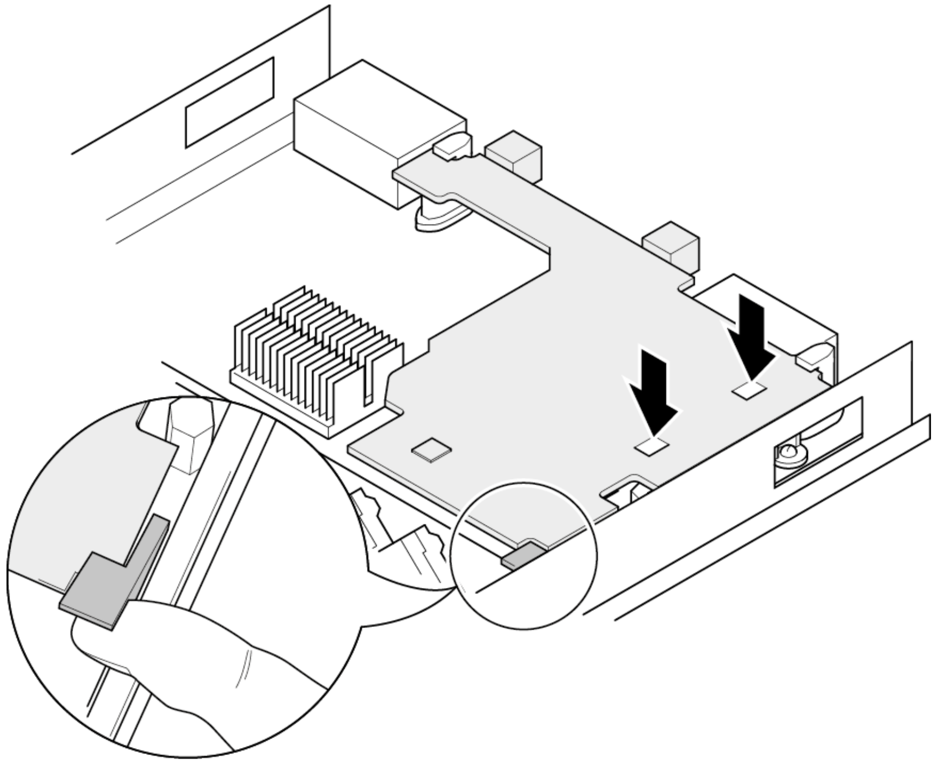


Figure 3: In position mezzanine adapter

5 Press the port connector gently into the connector block ensuring that the adapter is firmly and correctly seated in the connector block.

6 Replace the blade top cover.

7 When removing the adapter raise the release handle (shown on <Blue Text>Figure <Blue Text>3) to ease the adapter upwards until it can be freed from the connector block.

2.17 Solarflare Mezzanine Adapter SFN6832F-C61

The Solarflare SFN6832F-C61 is a Dual-Port SFP+ are 10GbE Mezzanine Adapters for the DELL PowerEdge C6100 series rack server. Each DELL PowerEdge node supports a single Solarflare mezzanine adapter.

1 The node should be extracted from the rack server in order to install the mezzanine adapter. Refer to the PowerEdge rack server manual if necessary.

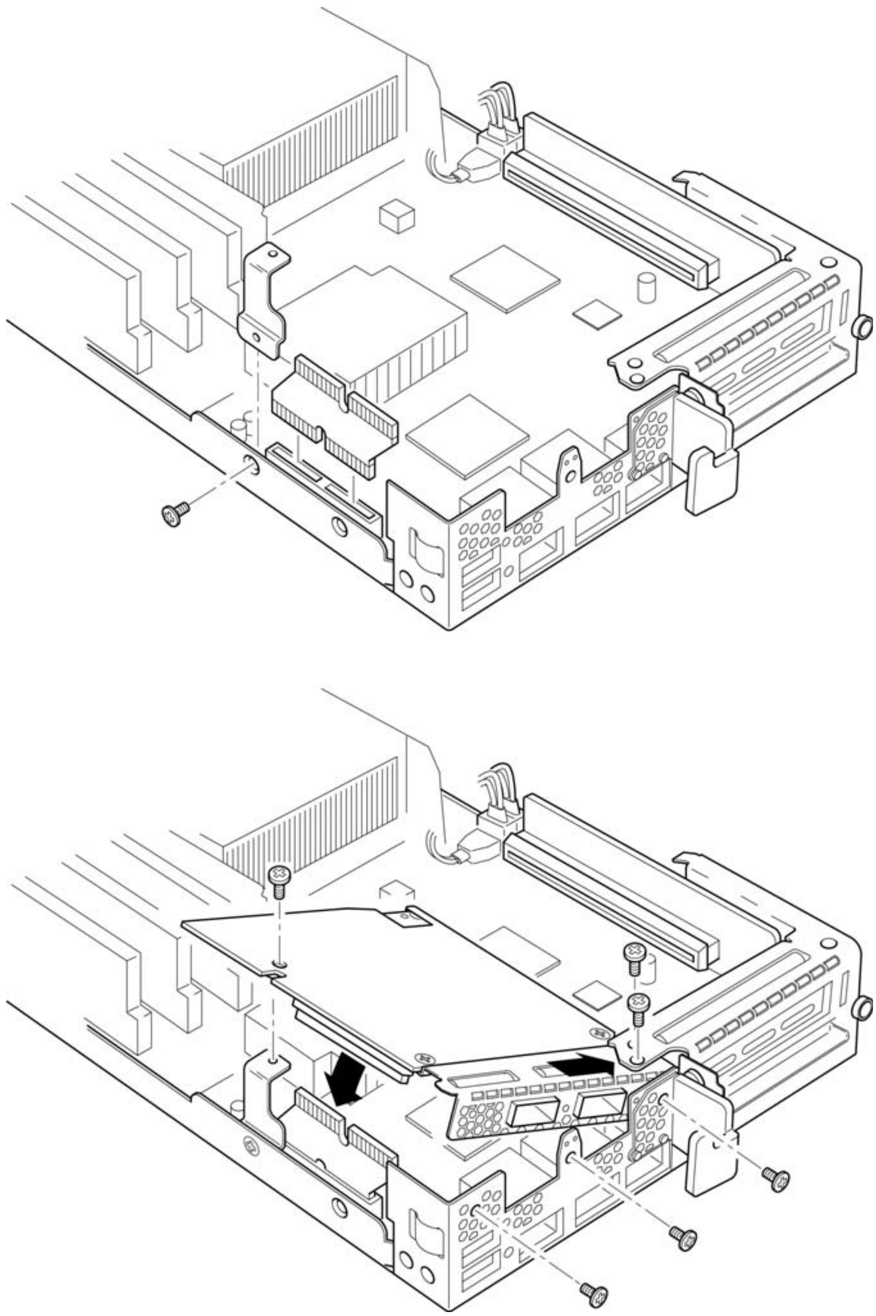


Figure 4: SFN6832F-C61 - Installing into the rack server node

2 Secure the side retaining bracket as shown in Figure 5 (top diagram)

3 Fit riser PCB card into the slot as shown in Figure 5 (top diagram). Note that the riser card only fits one way.

4 Offer the adapter to the node and ensure it lies underneath the chassis cover.

5 Lower the adapter into position making sure to connect the adapter slot with the to of the PCB riser card.

6 Secure the adapter using the supplied screws at the positions shown in the diagram.

2.18 Solarflare Mezzanine Adapter SFN6832F-C62

The Solarflare SFN6832F-C61 is a Dual-Port SFP+ are 10GbE Mezzanine Adapters for the DELL PowerEdge C6200 series rack server. Each DELL PowerEdge node supports a single Solarflare mezzanine adapter.

1 The node should be extracted from the rack server in order to install the mezzanine adapter. Refer to the PowerEdge rack server manual if necessary.

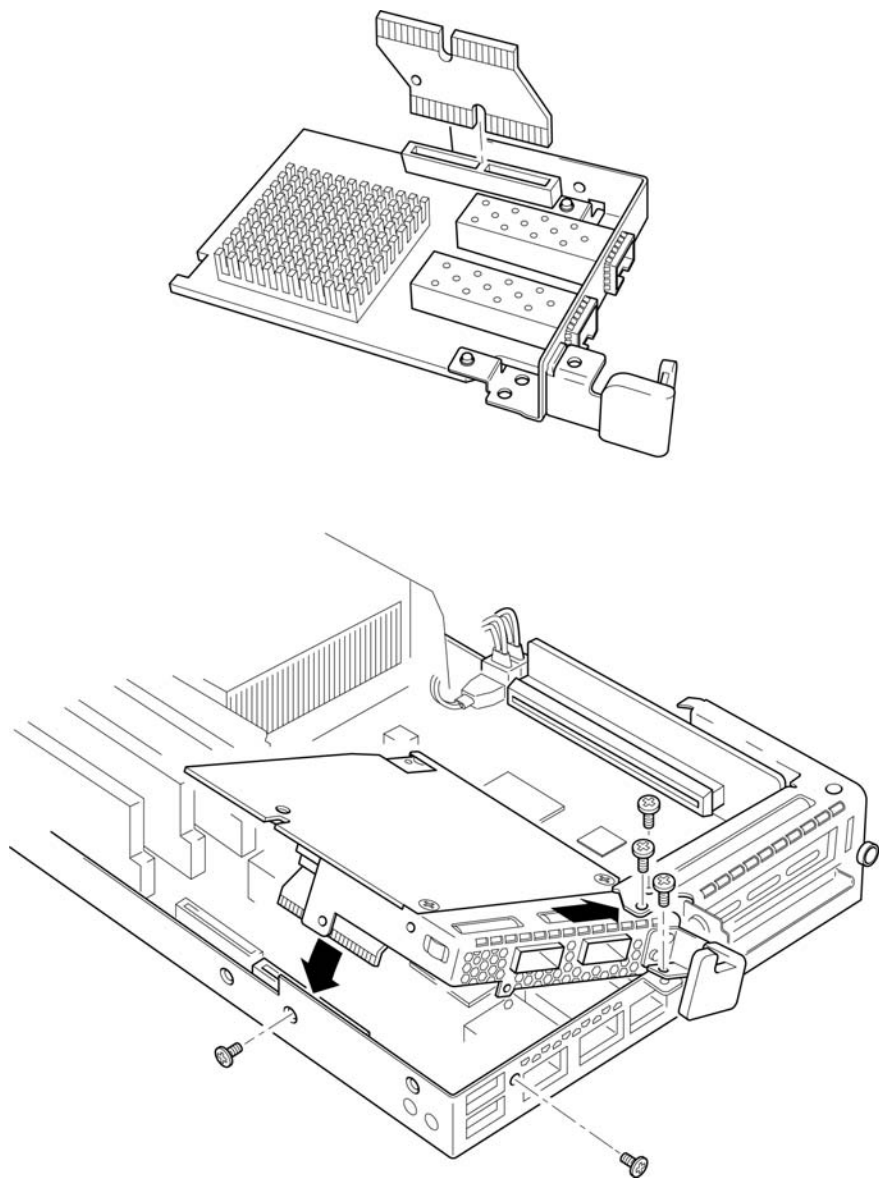


Figure 5: SFN6832F-C62 - Installing into the rack server node

2 Fit the PCB riser card to the underside connector on the adapter.

3 Offer the adapter to the rack server node ensuring it lies underneath the chassis cover.

4 Lower to adapter to connect the riser PCB card into the slot in the node.

5 Secure the adapter with the supplied screws at the points shown in the diagram.

2.19 Solarflare Precision Time Synchronization Adapters

The Solarflare SFN7042Q[1], SFN7142Q[1], SFN7122F[1], SFN7322F and SFN6322F adapters can generate hardware timestamps for PTP packets in support of a network precision time protocol deployment compliant with the IEEE 1588-2008 specification.

Customers requiring configuration instructions for these adapters and Solarflare PTP in a PTP deployment should refer to the *Solarflare Enhanced PTP User Guide* (SF-109110-CD).

2.20 Solarflare ApplicationOnload™ Engine

The ApplicationOnload™ Engine (AOE) SFA7942Q is a half-length, full-height PCIe form factor adapter combining the ultra-low latency dual-port 40GbE adapter with an Altera Stratix V FPGA.For details of the SFA7942Q adapter refer to the Solarflare ApplicationOnload Users Guide (SF-115020-CD).

The ApplicationOnload™ Engine (AOE) SFA6902F is a full-length PCIe form factor adapter that combines an ultra-low latency adapter with a tightly coupled ‘bump-in-the-wire’ FPGA.For details of installation and configuring applications that run on the SFA6902F AOE refer to the *Solarflare AOE User’s Guide* (SF-108389-CD). For details on developing custom applications to run on the FPGA refer to the *AOE Firmware Development Kit User Guide* (SF-108390-CD).

1. Requires an AppFlex™ license - refer to Solarflare AppFlex™ Technology Licensing. on page 14.

3 Solarflare Adapters on Linux

This chapter covers the following topics on the Linux® platform:

3.1 System Requirements

Refer to Software Driver Support on page 13 for supported Linux Distributions.

NOTE: SUSE Linux Enterprise Server 11 includes a version of the Solarflare network adapter Driver. This driver does not support the SFN512x family of adapters. To update the supplied driver, see SUSE Linux Enterprise Server Distributions on page 49

NOTE: Red Hat Enterprise Linux versions 5.5 and 6.0 include a version of the Solarflare adapter driver. This driver does not support the SFN512x family of adapters. Red Hat Enterprise Linux 5.6 and 6.1 includes a version of the Solarflare network driver for the SFN512x family of adapters. To update the supplied driver, see Installing Solarflare Drivers and Utilities on Linux on page 48

3.2 Linux Platform Feature Set

Table 14 lists the features supported by Solarflare adapters on Red Hat and SUSE Linux distributions.

Table 14: Linux Feature Set

Fault diagnostics	Support for comprehensive adapter and cable fault diagnostics and system reports. • See Linux Utilities RPM on page 71
Firmware updates	Support for Boot ROM, Phy transceiver and adapter firmware upgrades. • See Upgrading Adapter Firmware with sfupdate on page 86
Hardware Timestamps	Solarflare Flareon SFN7122F[1] SFN7142Q[1]and SFN7322F adapters support the hardware timestamping of all received packets - including PTP packets. The Linux kernel must support the SO_TIMESTAMPING socket option (2.6.30+) to allow the driver to support hardware packet timestamping. Therefore hardware packet timestamping is not available in RHEL 5.
Jumbo frames	Support for MTUs (Maximum Transmission Units) from 1500 bytes to 9216 bytes. • See Configuring Jumbo Frames on page 57

Table 14: Linux Feature Set

PXE and iSCSI booting	Support for diskless booting to a target operating system via PXE or iSCSI boot. • See Configuring the Boot ROM with sfboot on page 72 • See Solarflare Boot ROM Agent on page 435 PXE or iSCSI boot are not supported for Solarflare adapters on IBM System p servers.
Receive Side Scaling (RSS)	Support for RSS multi-core load distribution technology. • See Receive Side Scaling (RSS) on page 65.
ARFS	Linux Accelerated Receive Flow Steering. Improve latency and reduce jitter by steering packets to the core where a receiving application is running. See Receive Flow Steering (RFS) on page 67.
SARFS	Solarflare Accelerated RFS. See Solarflare Accelerated RFS (SARFS) on page 68.
Transmit Packet Steering (XPS)	Supported on Linux 2.6.38 and later kernels. Selects the transmit queue when transmitting on multi-queue devices. See Transmit Packet Steering (XPS) on page 69.
NIC Partitioning	Each physical port on the SFN7000 series adapter can be exposed as up to 8 PCIe Physical Functions (PF). See NIC Partitioning on page 58.
SR-IOV	Support for Linux KVM SR-IOV. • See SR-IOV Virtualization Using KVM on page 377 SR-IOV is not supported for Solarflare adapters on IBM System p servers.
Task offloads	Support for TCP Segmentation Offload (TSO), Large Receive Offload (LRO), and TCP/UDP/IP checksum offload for improved adapter performance and reduced CPU processing requirements. • See Configuring Task Offloading on page 56

Table 14: Linux Feature Set

Teaming	Improve server reliability and bandwidth by combining physical ports, from one or more Solarflare adapters, into a team, having a single MAC address and which function as a single port providing redundancy against a single point of failure. • See Setting Up Teams on page 58
Virtual LANs (VLANs)	Support for multiple VLANs per adapter. • See Setting Up VLANs on page 57

1. Requires an AppFlex license - for details refer to Solarflare AppFlex™ Technology Licensing. on page 14.

3.3 Solarflare RPMs

Solarflare supply RPM packages in the following formats:

- DKMS
- Source RPM

DKMS RPM

Dynamic Kernel Module Support (DKMS) is a framework where device driver source can reside outside the kernel source tree. It supports an easy method to rebuild modules when kernels are upgraded.

Execute the command `dkms --version` to determine whether DKMS is installed.

To install the Solarflare driver DKMS package execute the following command:

```
rpm -i sfc-dkms-<version>.noarch.rpm
```

Load the driver:

```
modprobe sfc
```


Building the Source RPM

These instructions may be used to build a source RPM package for use with Linux distributions or kernel versions where DKMS packages are not suitable.

NOTE: RPMs can be installed for multiple kernel versions.

1. Kernel headers for the running kernel must be installed at `/lib/modules/<kernel-version>/build`. On Red Hat systems, install the appropriate `kernel-smp-devel` or `kernel-devel` package. On SUSE systems install the `kernel-source` package.

2. To build a source RPM for the running kernel version from the source RPM, enter the following at the command-line:

   ```
   rpmbuild --rebuild <package_name>
   ```

   where `package_name` is the full path to the source RPM.

3. To build for a different kernel to the running system, enter the following command:

   ```
   rpmbuild --define 'kernel <kernel version>' --rebuild <package_name>
   ```

4. Install the resulting RPM binary package, as described in Installing Solarflare Drivers and Utilities on Linux

5. Load the driver:

   ```
   modprobe sfc
   ```

NOTE: The location of the generated RPM is dependent on the distribution and often the version of the distribution and the RPM build tools.

The RPM build process should print out the location of the RPM towards the end of the build process, but it can be hard to find amongst the other output.

Typically the RPM will be placed in `/usr/src/<dir>/RPMS/<arch>/`, where `<dir>` is distribution specific. Possible folders include Red Hat, **packages** or **extra**. The RPM file will be named using the same convention as the Solarflare provided pre-built binary RPMs.

The command: `find /usr/src -name "*sfc*.rpm`" will list the locations of all Solarflare RPMs.

3.4 Installing Solarflare Drivers and Utilities on Linux

- Red Hat Enterprise Linux Distributions on page 48
- SUSE Linux Enterprise Server Distributions on page 49

Linux drivers for Solarflare are available in DKMS and source RPM packages. The source RPM can be used to build binary RPMs for a wide selection of distributions and kernel variants. This section details how to install the resultant binary RPM.

Solarflare recommend using DKMS RPMs if the DKMS framework is available. See DKMS RPM on page 46 for more details.

NOTE: The Solarflare adapter should be physically installed in the host computer before installing the driver. The user must have root permissions to install the adapter drivers.

3.5 Red Hat Enterprise Linux Distributions

These instructions cover installation and configuration of the Solarflare network adapter drivers on Red Hat Enterprise Linux Server. Refer to Software Driver Support on page 13 for details of supported Linux distributions.

Refer to Building the Source RPM on page 47 for directions on creating the binary RPM.

1 Install the RPMs:

```
# rpm -ivh kernel-module-sfc-RHEL6-2.6.32-279.el6.x86_64-3.3.0.6262-1.x86_64.rpm
```

2 There are various tools that can be used for configuring the Solarflare Server Adapter:

a) The NetworkManager service and associated GUI tools. For more information about his refer to https://wiki.gnome.org/NetworkManager.

b) Solarflare recommend using the Network Administration Tool (**NEAT**) to configure the new network interface. **NEAT** is a GUI based application and therefore requires an X server to run.

c) Alternatively the command line program **Kudzu** can be used. However, you may find when kudzu is run that you are NOT presented with an option to configure the new network interface. If this occurs, carefully clear details of the Solarflare Server Adapter from the hardware database by removing all entries with “vendor id: 1924” in the `/etc/sysconfig/hwconf` file. Running kudzu again should now provide an option to configure the newly added network interface.

3 Apply the new network settings:

a) **NEAT** provides an option to **Activate** the new interface. The new network interface can then be used immediately (there is no need to reboot or restart the network service).

b) If you are not using **NEAT** you will need to reboot, or alternatively restart the networking service, by typing the following before the new Solarflare interface can be used:

```
# service network restart
```

3.6 SUSE Linux Enterprise Server Distributions

These instructions cover installation and configuration of the Solarflare Network Adapter drivers on SUSE Linux Enterprise Server. Refer to Software Driver Support on page 13 for details of supported distributions.

Refer to Building the Source RPM on page 47 for directions on creating the binary RPM.

1 The Solarflare drivers are currently classified as 'unsupported' by SUSE Enterprise Linux 10 (SLES10). To allow unsupported drivers to load in SLES10, edit the following file:

```
/etc/sysconfig/hardware/config
```

Find the line:

```
LOAD_UNSUPPORTED_MODULES_AUTOMATICALLY=no
```

and change no to `yes`.

For SLES 11, edit the last line in `/etc/modprobe.d/unsupported-modules` to:

```
allow_unsupported_modules 1
```

2 Install the RPMs:

```
# rpm -ivh kernel-module-sfc-2.6.5-7.244-smp-2.1.0111-0.sf.1.SLES9.i586.rpm
```

3 Run **YaST** to configure the Solarflare Network Adapter. When you select the Ethernet Controller, the **Configuration Name** will take one of the following forms:

- `eth-bus-pci-`*`dddd`*`:`*`dd`*`:`*`dd`*`.`*`N`* where *N* is either `0` or `1`.
- `eth-id-00:0F:53:`*`XX`*`:`*`XX`*`:`*`XX`*

Once configured, the **Configuration Name** for the correct Ethernet Controller will change to the second form, and an ethX interface will appear on the host. If the incorrect Ethernet Controller is chosen and configured, then the **Configuration Name** will remain as `eth-bus-pci-`*`dddd`*`:`*`dd`*`:`*`dd`*`.1` after configuration by YaST, and an ethX interface will not appear on the system. In this case, you should remove the configuration for this Ethernet Controller, and configure the other Ethernet Controller of the pair.

3.7 Installing DKMS Driver and Utilities on Ubuntu/Debian Servers

Net Driver DKMS

The Solarflare net driver DKMS package (SF-104979-LS) is available from:

https://support.solarflare.com/

1 Download the DKMS source package SF-104979-LS and unzip on the target server.

2 Create the .deb file:

```
sudo alien -c sfc-dkms-<version>.sf.1.noarch.rpm
```

This command generates the sfc-dkms_<version>_all.deb file.

The -c option is required to convert source scripts and build the driver.

3 Install the deb file:

```
sudo dpkg –i -dkms_<version>_all.deb
```

4 Reload the sfc driver:

```
modprobe –r sfc
modprobe sfc
```

Utilities

The Solarflare Linux Utilities package (SF-107601-LS) is available from:

https://support.solarflare.com/

1 Download and unzip the package on the target server.

2 Create the .deb file:

```
sudo alien sfutils-<version>.x86_64.rpm
```

This command generates the sfutils_<version>_amd64.deb file.

3 Install the deb file:

```
sudo dpkg -i sfutils_<version>_amd64.deb
```

4 Utilities sfupdate, sfkey, sfctool and sfboot are available on the server.

3.8 Unattended Installations

Building Drivers and RPMs for Unattended Installation

Linux unattended installation requires building two drivers:

- A minimal installation Solarflare driver that only provides networking support. This driver is used for network access during the installation process.
- An RPM that includes full driver support. This RPM is used to install drivers in the resultant Linux installation.

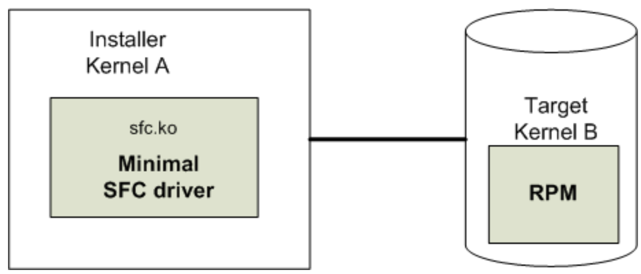


Figure 6: Unattended Installation RPM

Figure 6 shows how the unattended installation process works.

1 Build a minimal Solarflare driver needed for use in the installation kernel (Kernel A in the diagram above). This is achieved by defining “sfc_minimal” to rpmbuild. This macro disables hardware monitoring, MTD support (used for access to the adapters flash), I2C and `debugfs`. This results in a driver with no dependencies on other modules and allows networking support from the driver during installation.

```
# as normal user
$ mkdir -p /tmp/rpm/BUILD
$ rpm -i sfc-<ver>-1.src.rpm
$ rpmbuild -bc -D 'sfc_minimal=1' -D 'kernel=<installer kernel>' \
   /tmp/rpm/SPECS/sfc.spec
```

2 The Solarflare minimal driver `sfc.ko` can be found in `/tmp/rpm/BUILD/sfc-<ver>/linux_net/sfc.ko.` Integrate this minimal driver into your installer kernel, either by creating a driver disk incorporating this minimal driver or by integrating this minimal driver into initrd.

3 Build a full binary RPM for your Target kernel and integrate this RPM into your Target (Kernel B).

Driver Disks for Unattended Installations

Table 15 below identifies the various stages of an unattended installation process:

Table 15: Installation Stages

In Control	Stages of Boot	Setup needed
BIOS	PXE code on the adapter runs.	Adapter must be in PXE boot mode. See PXE Support on page 436.
SF Boot ROM (PXE)	DHCP request from PXE (SF Boot ROM).	DHCP server filename and next-server options.
SF Boot ROM (PXE)	TFTP request for filename to next-server, e.g. pxelinux.0	TFTP server.
pxelinux	TFTP retrieval of pxelinux configuration.	pxelinux configuration on TFTP server.
pxelinux	TFTP menu retrieval of Linux kernel image initrd.	pxelinux configuration Kernel, kernel command, initrd
Linux kernel/installer	Installer retrieves kickstart configuration, e.g. via HTTP.	Kickstart/AutoYaST configuration.
Target Linux kernel	kernel reconfigures network adapters.	DHCP server.

3.9 Unattended Installation - Red Hat Enterprise Linux

Documentation for preparing for a Red Hat Enterprise Linux network installation can be found at:

http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5/html/Installation_Guide/s1-begininstall-perform-nfs-x86.html

http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Installation_Guide/index.html

The prerequisites for a Network Kickstart installation are:

- Red Hat Enterprise Linux installation media.
- A Web server and/or FTP Server for delivery of the RPMs that are to be installed.
- A DHCP server for IP address assignments and to launch PXE Boot.

- A TFTP server for download of PXE Boot components to the machines being kickstarted.
- The BIOS on the computers to be Kickstarted must be configured to allow a network boot.
- A Boot CD-ROM or flash memory that contains the kickstart file or a network location where the kickstart file can be accessed.
- A Solarflare driver disk.

Unattended Red Hat Enterprise Linux installations are configured with Kickstart. The documentation for Kickstart can be found at:

http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5/html/Installation_Guide/ch-redhat-config-kickstart.html

http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Installation_Guide/ch-kickstart2.html

To install Red Hat Enterprise you need the following:

1 A modified `initrd.img` file with amended `modules.alias` and `modules.dep` which incorporates the Solarflare minimal driver for the installation kernel.

Find current aliases with the modinfo command:

```
modinfo sfc | grep alias
```

Then add the aliases found to the `modules.alias` file:

```
pci:v00001924d00001923sv*sd*bc*sc*i*
pci:v00001924d00000923sv*sd*bc*sc*i*
pci:v00001924d00001903sv*sd*bc*sc*i*
pci:v00001924d00000903sv*sd*bc*sc*i*
pci:v00001924d00000813sv*sd*bc*sc*i*
pci:v00001924d00000803sv*sd*bc*sc*i*
pci:v00001924d00000710sv*sd*bc*sc*i*
pci:v00001924d00000a23sv*sd*bc*sc*i*
pci:v00001924d00001a23sv*sd*bc*sc*i*
```

2 Identify the driver dependencies using the modinfo command:

```
modinfo ./sfc.ko | grep depends
depends:    i2c-core,mii,hwmon,hwmon-vid,i2c-algo-bit mtdcore mtdpart
```

All modules listed as depends must be present in the initrd file image. In addition the user should be aware of further dependencies which can be resolved by adding the following lines to the modules.dep file:

```
sfc: i2c-core mii  hwmon  hwmon-vid  i2c-algo-bit  mtdcore  mtdpart[1]
i2c-algo-bit:  i2c-core
mtdpart:  mtdcore
```

1. For Red Hat Enterprise Linux from version 5.5 add `mdio` to this line.

3. A configured kickstart file with the Solarflare Driver RPM manually added to the `%Post` section. For example:

```
%post

/bin/mount -o ro <IP Address of Installation server>:/<path to
location directory containing Solarflare RPM> /mnt
/bin/rpm -Uvh /mnt/<filename of Solarflare RPM>
/bin/umount /mnt
```

3.10 Unattended Installation - SUSE Linux Enterprise Server

Unattended SUSE Linux Enterprise Server installations are configured with AutoYaST. The documentation for AutoYaST can be found at:

http://www.suse.com/~ug/autoyast_doc/index.html

The prerequisites for a Network AutoYaST installation are:

- SUSE Linux Enterprise installation media.
- A DHCP server for IP address assignments and to launch PXE Boot.
- A NFS or FTP server to provide the installation source.
- A TFTP server for the download of the kernel boot images needed to PXE Boot.
- A boot server on the same Ethernet segment.
- An install server with the SUSE Linux Enterprise Server OS.
- An AutoYaST configuration server that defines rules and profiles.
- A configured AutoYaST Profile (control file).

Further Reading

- SUSE Linux Enterprise Server remote installation:
  http://www.novell.com/documentation/sles10/sles_admin/?page=/documentation/sles10/sles_admin/data/cha_deployment_remoteinst.html
- SUSE install with PXE Boot:
  http://en.opensuse.org/SuSE_install_with_PXE_boot

3.11 Configuring the Solarflare Adapter

Ethtool is a standard Linux tool that you can use to query and change Ethernet adapter settings. Ethtool can be downloaded from http://sourceforge.net/projects/gkernel/files/ethtool/.

The general command for ethtool is as follows:

```
ethtool <-option> <ethX>
```

where X is the identifier of the interface. Root access is required to configure adapter settings.

Hardware Timestamps

The Solarflare Flareon SFN7000 series adapters can support hardware timestamping for all received network packets.

The Linux kernel must support the SO_TIMESTAMPING socket option (2.6.30+) therefore hardware packet timestamping is not supported on RHEL 5.

For more information about using the kernel timestamping API, users should refer to the Linux documentation: http://lxr.linux.no/linux/Documentation/networking/timestamping.txt

Configuring Speed and Modes

Solarflare adapters by default automatically negotiate the connection speed to the maximum supported by the link partner. On the 10GBASE-T adapters "auto" instructs the adapter to negotiate the highest speed supported in common with its link partner. On SFP+ adapters, "auto" instructs the adapter to use the highest link speed supported by the inserted SFP+ module. On 10GBASE-T and SFP+ adapters, any other value specified will fix the link at that speed, regardless of the capabilities of the link partner, which may result in an inability to establish the link. Dual speed SFP+ modules operate at their maximum (10G) link speed unless explicitly configured to operate at a lower speed (1G).

The following commands demonstrate ethtool to configure the network adapter Ethernet settings.

- Identify interface configuration settings:

  ```
  ethtool ethX
  ```

- Set link speed:

  ```
  ethtool -s ethX speed 1000|100
  ```

- To return the connection speed to the default auto-negotiate, enter:

  ```
  ethtool -s <ethX> autoneg on
  ```

- Configure auto negotiation:

  ```
  ethtool -s ethX autoneg [on|off]
  ```


- Set auto negotiation advertised speed 1G:

  `ethtool -s ethX advertise 0x20`

- Set autonegotiation advertised speed 10G:

  `ethtool -s ethX advertise 0x1000`

- Set autonegotiation advertised speeds 1G and 10G:

  `ethtool -s ethX advertise 0x1020`

- Identify interface auto negotiation pause frame setting:

  `ethtool -a ethX`

- Configure auto negotiation of pause frames:

  `ethtool -A ethX autoneg on [rx on|off] [tx on|off]`

NOTE: Due to a limitation in ethtool, when auto-negotiation is enabled, the user must specify both speed and duplex mode or speed and set an advertise mask otherwise speed configuration will not function.

Configuring Task Offloading

Solarflare adapters support transmit (Tx) and receive (Rx) checksum offload, as well as TCP segmentation offload. To ensure maximum performance from the adapter, all task offloads should be enabled, which is the default setting on the adapter. For more information, see Performance Tuning on Linux on page 93.

To change offload settings for Tx and Rx, use the ethtool command:

```
ethtool --offload <ethX> [rx on|off] [tx on|off]
```

Configuring Receive/Transmit Ring Buffer Size

By default receive and transmit ring buffers on the Solarflare adapter support 1024 descriptors. The user can identify and reconfigure ring buffer sizes using the ethtool command.

To identify the current ring size:

```
ethtool -g ethX
```

To set the new transmit or receive ring size to value N

```
ethtool -G ethX [rx N| tx N]
```

The ring buffer size must be a value between 128 and 4096. On the SFN7000 series adapters the maximum TX buffer size is restricted to 2048. Buffer size can also be set directly in the `modprobe.conf` file or add the options line to a file under the `/etc/modprobe.d` directory e.g.

```
options sfc rx_ring=4096
```

Using the modprobe method sets the value for all Solarflare interfaces. Then reload the driver for the option to become effective:

```
modprobe -r sfc
modprobe sfc
```


Configuring Jumbo Frames

Solarflare adapters support frame sizes from 1500 bytes to 9216 bytes. For example, to set a new frame size (MTU) of 9000 bytes, enter the following command:

```
ifconfig <ethX> mtu 9000
```

To make the changes permanent, edit the network configuration file for `<ethX>`; for example, `/etc/sysconfig/network-scripts/ifcfg-eth1` and append the following configuration directive, which specifies the size of the frame in bytes:

```
MTU=9000
```

3.12 Setting Up VLANs

VLANs offer a method of dividing one physical network into multiple broadcast domains. In enterprise networks, these broadcast domains usually match with IP subnet boundaries, so that each subnet has its own VLAN. The advantages of VLANs include:

- Performance
- Ease of management
- Security
- Trunks
- You don't have to configure any hardware device, when physically moving your server to another location.

To set up VLANs, consult the following documentation:

- To configure VLANs on SUSE Linux Enterprise Server, see:

  http://www.novell.com/support/viewContent.do?externalId=3864609
- To configure tagged VLAN traffic only on Red Hat Enterprise Linux, see:

  http://kbase.redhat.com/faq/docs/DOC-8062
- To configure mixed VLAN tagged and untagged traffic on Red Hat Enterprise Linux, see:

  http://kbase.redhat.com/faq/docs/DOC-8064

3.13 Setting Up Teams

Teaming network adapters (network bonding) allows a number of physical adapters to act as one, virtual adapter. Teaming network interfaces, from the same adapter or from multiple adapters, creates a single virtual interface with a single MAC address.

The virtual adapter or virtual interface can assist in load balancing and providing failover in the event of physical adapter or port failure.

Teaming configuration support provided by the Linux bonding driver includes:

- 802.3ad Dynamic link aggregation
- Static link aggregation
- Fault Tolerant

To set up an adapter team, consult the following documentation:

- General:
  http://www.kernel.org/doc/Documentation/networking/bonding.txt
- RHEL 5:
  http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.4/html/Deployment_Guide/s2-modules-bonding.html
- RHEL6:
  http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Deployment_Guide/s2-networkscripts-interfaces-chan.html
- SLES:
  http://www.novell.com/documentation/sles11/book_sle_admin/data/sec_basicnet_yast.html#sec_basicnet_yast_netcard_man

3.14 NIC Partitioning

NIC Partitioning is a feature supported on Solarflare adapters starting with the SFN7000 series. By partitioning the NIC, each physical network port can be exposed to the host as multiple PCIe Physical Functions (PF) with each having a unique interface name and unique MAC address.

When the Solarflare NET driver (sfc.ko) is loaded in the host, each PF is backed by a virtual adapter connected to a virtual port. A switching function supports the transport of network traffic between virtual ports (vport) and the physical port. Partitioning is particularly useful when, for example, splitting a single 40GbE interface into multiple PFs.

- Up to 16 PFs and 16 MAC addresses are support PER ADAPTER.
- On a 10GbE dual-port adapter each physical port can be exposed as a maximum 8 PFs.

- On a 40GbE dual-port adapter (in 2*40G mode) each physical port can be exposed as a maximum 8 PFs.
- On a 40GbE dual-port adapter (in 4*10G mode) each physical port can be exposed as a maximum 4 PFs.

NIC Partitioning Without VLANs

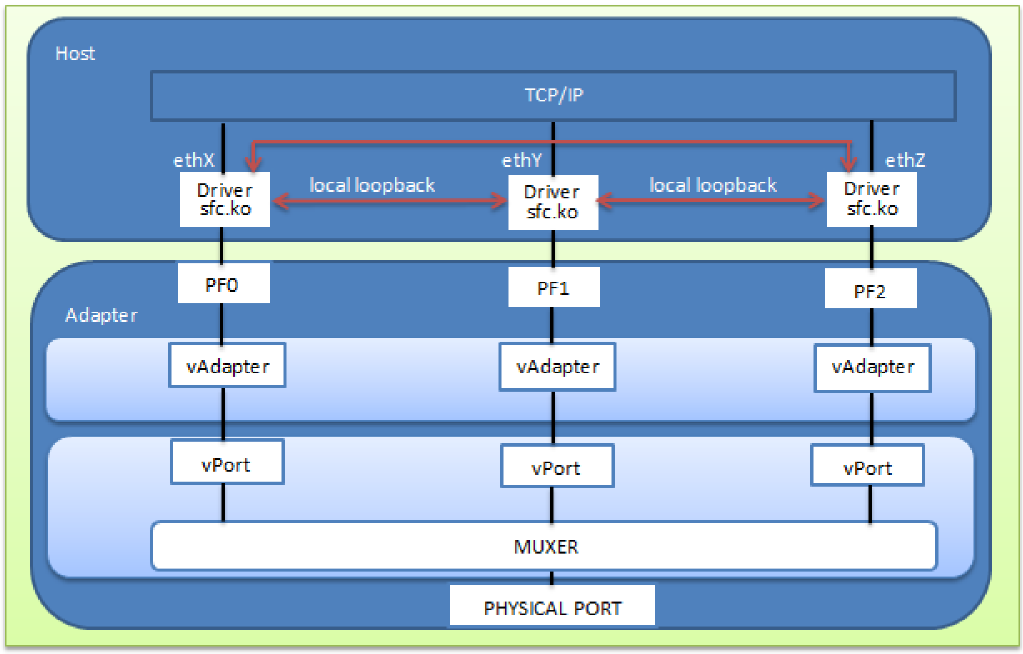


Figure 7: NIC Partitioning - without VLANs

- Configured without VLANs, all PFs are in the same Ethernet layer 2 broadcast domain i.e. a packet broadcast from any one PF would be received by all other PFs.
- Transmitted packets go directly to the wire. Packets sent between PFs are routed through the local TCP/IP stack loopback interface without touching the sfc driver.
- Received broadcast packets are replicated to all PFs.
- Received multicast packets are delivered to each subscriber.
- Received unicast packets are delivered to the PF with a matching MAC address. Because the TCP/IP stack has multiple network interfaces on the same broadcast domain, there is always the possibility that any interface could respond to an ARP request. To avoid this the user should use `arp_ignore=2` to avoid ARP cache pollution ensuring that ARP responses are only sent from an interface if the target IP address in the ARP request matches the interface address with both sender/receiver IP addresses in the same subnet.
- To set `arp_ignore` for the current session:

```
echo 2 >/proc/sys/net/ipv4/conf/all/arp_ignore
```


- To set `arp_ignore` permanently (does not affect the current session), add the following line to the `/etc/sysctl.conf` file:

  ```
  net.ipv4.conf.all.arp_ignore = 2
  ```

- The MUXER function is a layer2 switching function for received traffic enabled in adapter firmware. When the OS delivers traffic to local interfaces via the loopback interface, the MUXER acts as a layer2 switch for both transmit and receive.

VLAN Support

When PFs are configured with VLAN tags each PF must be in a different VLAN. The MUXER function acts as a VLAN aggregator such that transmitted packets are sent to the wire and received packets are demultiplexed based on the VLAN tags. VLAN tags are added/stripped by the adapter firmware transparent to the OS and driver. VLAN tags can be assigned when PFs are enabled using the sfboot command. A single PF can be assigned VLAN tag 0 allowing it to receive untagged traffic.

```
# sfboot switch-mode=partitioning pf-count=3 pf-vlan=0,200,300
```

The first VLAN ID in the pf-vlan comma separated list is assigned to the first PF of the physical port and thereafter tags are assigned to PFs in lowest MAC address order.

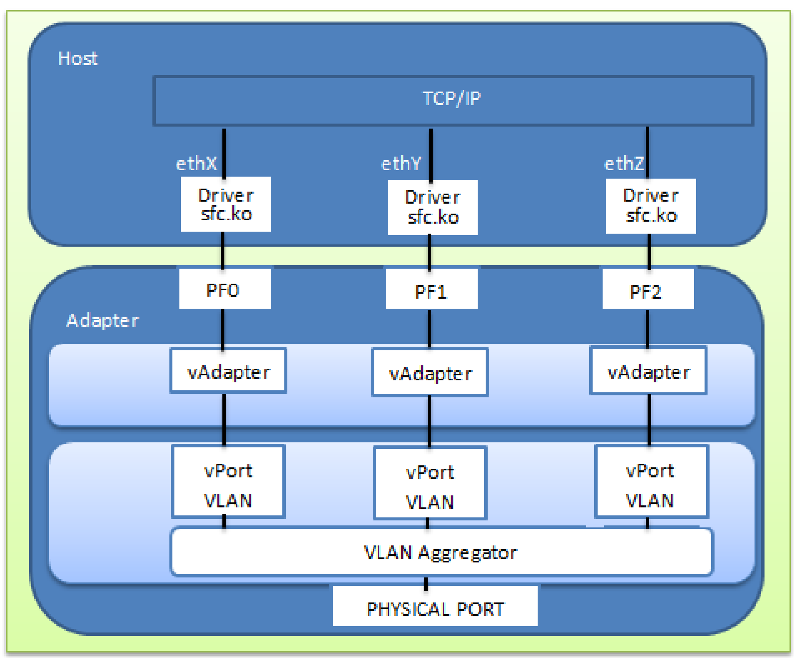


Figure 8: NIC Partitioning - VLAN Support

NIC Partitioning Configuration

Up to 16 PFs and 16 MAC addresses are supported per adapter. The PF count value applies to all physical ports. Ports cannot be configured individually.

1 Ensure the Solarflare adapter driver (sfc.ko) is installed on the host.

2 The `sfboot` utility (`pf-count`) from the Solarflare Linux Utilities package (SF-107601-LS) is used to partition physical interfaces to the required number of PFs.

3 To partition all ports (example configures 4 PFs per port):

```
# sfboot switch-mode=partitioning pf-count=4
Solarflare boot configuration utility [v4.5.0]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-2005

eth2:
  Boot image                             Option ROM only
    Link speed                           Negotiated automatically
    Link-up delay time                   5 seconds
    Banner delay time                    2 seconds
    Boot skip delay time                 5 seconds
    Boot type                            Disabled
  Physical Functions per port            4
  MSI-X interrupt limit                  32
  Number of Virtual Functions            0
  VF MSI-X interrupt limit               8
  Firmware variant                       full feature / virtualization
  Insecure filters                       Disabled
  MAC spoofing                           Disabled
  VLAN tags                              None
  Switch mode                            Partitioning
```

A cold reboot of the server is required for sfboot changes to be effective.

4 Following reboot each PF will be visible using the `lspci` command:

```
# lspci -d 1924:

07:00.0 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.1 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.2 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.3 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.4 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.5 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.6 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.7 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
```

- If more than 8 functions are required the server must support ARI - see Alternative Routing-ID Interpretation (ARI) on page 379.
- Solarflare also recommend setting `pci=realloc` in the kernel configuration grub file - refer to Kernel Configuration on page 379 for details.

5 To identify which physical port a given network interface is using:

```
# cat /sys/class/net/eth<N>/device/physical_port
```


6 If the Solarflare driver is loaded, PFs will also be visible using the `ifconfig` command where each PF is listed with a unique MAC address.

Software Requirements

The server must have the following (minimum) net driver and firmware versions to enable NIC Partitioning:

```
# ethtool -i eth<N>
driver: sfc
version: 4.4.1.1017
firmware-version: 4.4.2.1011 rx0 tx0
```

The adapter must be using the *full-feature* firmware variant which can be selected using the `sfboot` utility and confirmed with `rx0 tx0` appearing after the version number in the output from ethtool as shown above.

The firmware update utility (`sfupdate`) and bootROM configuration tool (`sfboot`) are available in the Solarflare Linux Utilities package (SF-107601-LS issue 28 or later).

3.15 NIC Partitioning with SR-IOV

When combining NIC partitioning with SR-IOV, every partition (PF) must be in a separate VLAN. The user is able to create a number of PFs per physical port and associate a number of VFs with each PF. Within this layer2 broadcast domain there is switching between a PF and its associated VFs.

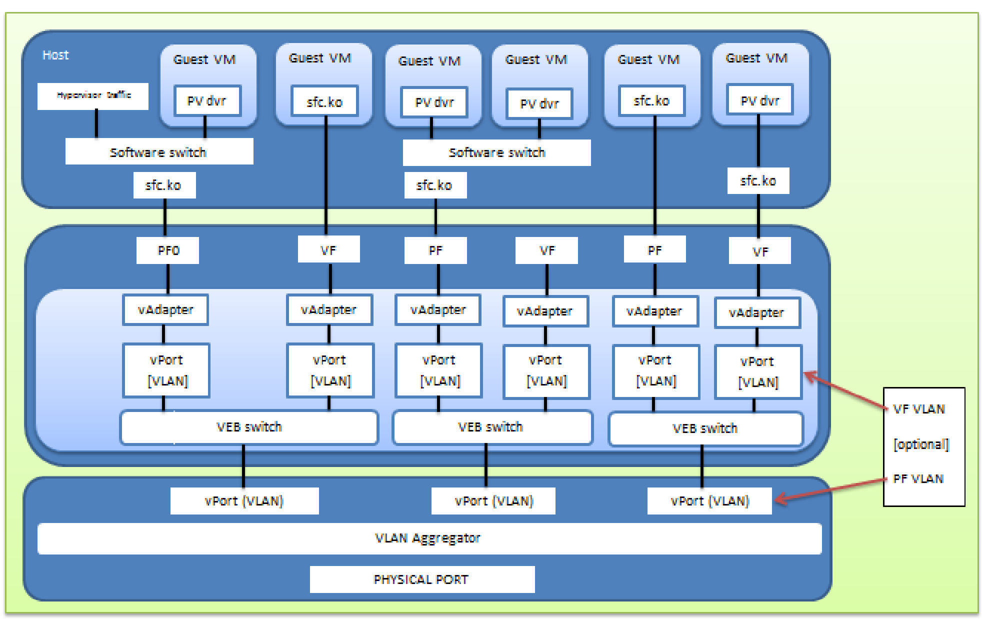


Figure 9: NIC Partitioning with SR-IOV

Configuration

1 Use the sfboot utility to set the firmware switch-mode, create PFs, assign unique VLAN ID to each PF and assign a number of VFs for each PF.

In the following example 4 PFs are configured per physical port and 2 VFs per PF:

```
# sfboot switch-mode=partitioning-with-sriov pf-count=4 /
pf-vlans=0,100,110,120 vf-count=2

eth10:
  Interface-specific boot options are not available. Adapter-wide
  options are available via eth4 (00-0F-53-21-00-60).

eth11:
  Interface-specific boot options are not available. Adapter-wide
  options are available via eth4 (00-0F-53-21-00-60).

eth12:
  Interface-specific boot options are not available. Adapter-wide
  options are available via eth4 (00-0F-53-21-00-60).

eth13:
  Interface-specific boot options are not available. Adapter-wide
  options are available via eth4 (00-0F-53-21-00-60).

eth14:
  Interface-specific boot options are not available. Adapter-wide
  options are available via eth4 (00-0F-53-21-00-60).

eth15:
  Interface-specific boot options are not available. Adapter-wide
  options are available via eth4 (00-0F-53-21-00-60).

eth4:
  Boot image                            Option ROM only
    Link speed                          Negotiated automatically
    Link-up delay time                  5 seconds
    Banner delay time                   2 seconds
    Boot skip delay time                5 seconds
    Boot type                           Disabled
  Physical Functions per port           4
  MSI-X interrupt limit                 32
  Number of Virtual Functions           2
  VF MSI-X interrupt limit              8
  Firmware variant                      full feature / virtualization
  Insecure filters                      Disabled
  MAC spoofing                          Disabled
  VLAN tags                             0,100,110,120
  Switch mode                           Partitioning with SRIOV

eth5:
  Boot image                            Option ROM only
    Link speed                          Negotiated automatically
    Link-up delay time                  5 seconds
    Banner delay time                   2 seconds
```



```
  Boot skip delay time                 5 seconds
  Boot type                            Disabled
Physical Functions per port            4
MSI-X interrupt limit                  32
Number of Virtual Functions            2
VF MSI-X interrupt limit               8
Firmware variant                       full feature / virtualization
Insecure filters                       Disabled
MAC spoofing                           Disabled
VLAN tags                              0,100,110,120
Switch mode                            Partitioning with SRIOV
```

2 PF interfaces are visible in the host using the `ifconfig` command:

```
eth4      Link encap:Ethernet  HWaddr 00:0F:53:21:00:60
eth5      Link encap:Ethernet  HWaddr 00:0F:53:21:00:61
eth10     Link encap:Ethernet  HWaddr 00:0F:53:21:00:64
eth11     Link encap:Ethernet  HWaddr 00:0F:53:21:00:65
eth12     Link encap:Ethernet  HWaddr 00:0F:53:21:00:66
eth13     Link encap:Ethernet  HWaddr 00:0F:53:21:00:63
eth14     Link encap:Ethernet  HWaddr 00:0F:53:21:00:62
eth15     Link encap:Ethernet  HWaddr 00:0F:53:21:00:67
```

3 The output from steps 1 and 2 above identifies a server with 2 physical interfaces (eth4/eth5), 4 PFs per physical port and identifies the following PF-VLAN configuration:

Table 16: PF-VLAN Configuration

Interface	MAC Address	PF	VLAN ID
eth4	00:0F:53:21:00:60	PF0	0
eth10	00:0F:53:21:00:64	PF4	110
eth12	00:0F:53:21:00:66	PF6	120
eth14	00:0F:53:21:00:62	PF2	100
eth5	00:0F:53:21:00:61	PF1	0
eth11	00:0F:53:21:00:65	PF5	110
eth13	00:0F:53:21:00:63	PF3	100
eth15	00:0F:53:21:00:67	PF7	120

4 Refer to SR-IOV Configuration on page 383 for procedures to create VMs and VFs.

VLAN Configuration

When using partitioning with SR-IOV, all PFs must have a unique VLAN tag. A single PF from each physical port can use tag 0 (zero) to receive untagged traffic. VLAN tags are transparently inserted/stripped by the adapter firmware.

LACP Bonding

LACP Bonding is not currently supported using the NIC Partitioning configuration mode as the LACP partner i.e. the switch will be unaware of the configured partitions.

Users are advised to refer to the sfc driver release notes for current limitations when using the NIC partitioning features.

3.16 Receive Side Scaling (RSS)

Solarflare adapters support Receive Side Scaling (RSS). RSS enables packet receive-processing to scale with the number of available CPU cores. RSS requires a platform that supports MSI-X interrupts. RSS is enabled by default.

When RSS is enabled the controller uses multiple receive queues to deliver incoming packets. The receive queue selected for an incoming packet is chosen to ensure that packets within a TCP stream are all sent to the same receive queue – this ensures that packet-ordering within each stream is maintained. Each receive queue has its own dedicated MSI-X interrupt which ideally should be tied to a dedicated CPU core. This allows the receive side TCP processing to be distributed amongst the available CPU cores, providing a considerable performance advantage over a conventional adapter architecture in which all received packets for a given interface are processed by just one CPU core. RSS can be restricted to only process receive queues on the NUMA node local to the Solarflare adapter. To configure this the driver module option `rss_numa_local` should be set to 1.

By default the driver enables RSS and configures one RSS Receive queue per CPU core. The number of RSS Receive queues can be controlled via the driver module parameter `rss_cpus`. The following table identifies rss_cpus options.

Table 17: rss_cpus Options

Option	Description	Interrupt Affinity (MSI-X)
<num_cpus>	Indicates the number of RSS queues to create.	A separate MSI-X interrupt for a receive queue is affinitized to each CPU.
packages	An RSS queue will be created for each multi-core CPU package. The first CPU in the package will be chosen.	A separate MSI-X interrupt for a receive queue, is affinitized to each of the designated package CPUs.

Table 17: rss_cpus Options

Option	Description	Interrupt Affinity (MSI-X)
cores	An RSS queue will be created for each CPU. The first hyperthread instance (If CPU has hyperthreading) will be chosen. The default option.	A separate MSI-X interrupt for a receive queue, is affinitized to each of the CPUs.
hyperthreads	An RSS queue will be created for each CPU hyperthread (hyperthreading must be enabled).	A separate MSI-X interrupt for a receive queue, is affinitized to each of the hyperthreads.

Add the following line to `/etc/modprobe.conf file` or add the options line to a user created file under the `/etc/modprobe.d` directory. The file should have a `.conf` extension:

```
options sfc rss_cpus=<option>
```

To set `rss_cpus` equal to the number of CPU cores:

```
options sfc rss_cpus=cores
```

Sometimes, it can be desirable to disable RSS when running single stream applications, since all interface processing may benefit from taking place on a single CPU:

```
options sfc rss_cpus=1
```

The driver must be reloaded to enable option changes:

NOTE: The association of RSS receive queues to a CPU is governed by the receive queue's MSI-X interrupt affinity. See Interrupt Affinity on page 103 for more details.

```
rmmod sfc
modprobe sfc
```

NOTE: The `rss_cpus` parameter controls the number of MSI-X interrupts used by each Solarflare port. Unfortunately, some older Linux version have a bug whereby the maximum number of MSI-X interrupts used by a PCI function is fixed at the first driver load. For instance, if the drivers are first loaded with `rss_cpus=1`, all subsequent driver loads will always use `rss_cpus=1`.

Red Hat Enterprise Linux 5 update 2 (and above), and SUSE Enterprise Linux 11 are not affected by this issue.

To workaround this issue, you must reboot the host after modifying `rss_cpus`.

NOTE: RSS also works for UDP packets. For UDP traffic the Solarflare adapter will select the Receive CPU based on IP source and destination addresses. Solarflare adapters support IPv4 and IPv6 RSS.

3.17 Receive Flow Steering (RFS)

RFS will attempt to steer packets to the core where a receiving application is running. This reduces the need to move data between processor caches and can significantly reduce latency and jitter. Modern NUMA systems, in particular, can benefit substantially from RFS where packets are delivered into memory local to the receiving thread.

Unlike RSS which selects a CPU from a CPU affinity mask set by an administrator or user, RFS will store the application's CPU core identifier when the application process calls `recvmsg()` or `sendmsg()`.

- A hash is calculated from a packet's addresses or ports (2-tuple or 4-tuple) and serves as the consistent hash for the flow associated with the packet.
- Each receive queue has an associated list of CPUs to which RFS may enqueue the received packets for processing.
- For each received packet, an index into the CPU list is computed from the flow hash modulo the size of the CPU list.

There are two types of RFS implementation; Soft RFS and Hardware (or Accelerated) RFS.

Soft RFS is a software feature supported since Linux 2.6.35 that attempts to schedule protocol processing of incoming packets on the same processor as the user thread that will consume the packets.

Accelerated RFS requires Linux kernel version 2.6.39 or later, with the Linux sfc driver or Solarflare v3.2 network adapter driver.

RFS can dynamically change the allowed CPUs that can be assigned to a packet or packet stream and this introduces the possibility of out of order packets. To prevent out of order data, two tables are created that hold state information used in the CPU selection.

- **Global_flow_table**: Identifies the number of simultaneous flows that are managed by RFS.
- **Per_queue_table**: Identifies the number of flows that can be steered to a queue. This holds state as to when a packet was last received.

The tables support the steering of incoming packets from the network adapter to a receive queue affinitized to a CPU where the application is waiting to receive them. The Solarflare accelerated RFS implementation requires configuration through the two tables and the ethtool -K command.

The following sub-sections identify the RFS configuration procedures:

Kernel Configuration

Before using RFS the kernel must be compiled with the kconfig symbol `CONFIG_RPS` enabled. Accelerated RFS is only available if the kernel is complied with the kconfig symbol `CONFIG_RFS_ACCEL` enabled.

Global Flow Count

Configure the number of simultaneous flows that will be managed by RFS. The suggested flow count will depend on the expected number of active connections at any given time and may be less than the number of open connections. The value is rounded up to the nearest power of two.

```
# echo 32768 > /proc/sys/net/core/rps_sock_flow_entries
```

Per Queue Flow Count

For each adapter interface there will exist a ‘queue’ directory containing one ‘rx’ or ‘tx’ subdirectory for each queue associated with the interface. For RFS only the receive queues are relevant.

```
# cd /sys/class/net/eth3/queue
```

Within each ‘rx’ subdirectory, the `rps_flow_cnt` file holds the number of entries in the per- queue flow table. If only a single queue is used then `rps_flow_cnt` will be the same as `rps_sock_flow_entries`. When multiple queues are configured the count will be equal to `rps_sock_flow_entries`/*N* where *N* is the number of queues, for example:

`rps_sock_flow_entries` = 32768 and there are 16 queues then `rps_flow_cnt` for each queue will be configured as 2048.

```
# echo 2048 > /sys/class/net/eth3/queues/rx-0/rps_flow_cnt
# echo 2048 > /sys/class/net/eth3/queues/rx-1/rps_flow_cnt
```

Disable RFS

To turn off RFS using the following command:

```
# ethtool -K <devname> ntuple off
```

3.18 Solarflare Accelerated RFS (SARFS)

The Solarflare Accelerated RFS feature directs TCP flows to queues processed on the same CPU core as the user process which is consuming the flow. By querying the CPU when a TCP packet is sent, the transmit queue can be selected from the interrupt associated with the correct CPU core. A hardware filter directs the receive flow to the same queue.

SARFS is provided for servers that do not support standard Linux ARFS. For details of Linux ARFS, refer to the previous section. Additional information can be found at the following link:

https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Performance_Tuning_Guide/network-acc-rfs.html

Overall SARFS can improve bandwidth, especially for smaller packets and because core assignment is not subject to the semi-random selection of transmit and receive queues, both bandwidth and latency become more consistent.

The SARFS feature is disabled by default and can be enabled using net driver module parameters. Driver module parameters can be specified in a user created file (e.g. sfc.conf) in the `/etc/modprobe.d` directory:

```
sxps_enabled
sarfs_table_size
sarfs_global_holdoff_ms
sarfs_sample_rate
```

If the kernel supports XPS, this should be enabled when using the SARFS feature. When the kernel does not supports XPS, the `sxps_enabled` parameter should be enabled when using SARFS.

NOTE: sxps_enabled is known to work on RHEL version up to and including RHEL6.5, but does not function on RHEL7 due to changes in the interrupt hint policy.

Refer to Module Parameters on page 113 for a description of the SARFS driver module parameters.

3.19 Transmit Packet Steering (XPS)

Transmit Packet Steering (XPS) is supported in Linux 2.6.38 and later. XPS is a mechanism for selecting which transmit queue to use when transmitting a packet on a multi-queue device.

XPS is configured on a per transmit queue basis where a bitmap of CPUs identifies the CPUs that may use the queue to transmit.

Kernel Configuration

Before using XPS the kernel must be compiled with the kconfig symbol `CONFIG_XPS` enabled.

Configure CPU/Hyperthreads

Within in each `/sys/class/net/eth3/queues/tx-N` directory there exists an `xps_cpus` file which contains a bitmap of CPUs that can use the queue to transmit. In the following example transmit queue 0 can be used by the first two CPUs and transmit queue 1 can be used by the following two CPUs:

```
# echo 3 > /sys/class/net/eth3/queues/tx-0/xps_cpus
# echo c > /sys/class/net/eth3/queues/tx-0/xps_cpus
```

If hyperthreading is enabled, each hyperthread is identified as a separate CPU, for example if the system has 16 cores but 32 hyperthreads then the transmit queues should be paired with the hyperthreaded cores:

```
# echo 30003 > /sys/class/net/eth3/queues/tx-0/xps_cpus
# echo c000c > /sys/class/net/eth3/queues/tx-0/xps_cpus
```


XPS - Example Configuration

System Configuration:

- Single Solarflare adapter
- 2 x 8 core processors with hyperthreading enabled to give a total of 32 cores
- rss_cpus=8
- Only 1 interface on the adapter is configured
- The IRQ Balance service is disabled

Identify interrupts for the configured interface:

```
# cat /proc/interrupts | grep 'eth3\ | CPU'
```

```
> cat /proc/irq/132/smp_affinity
00000000,00000000,00000000,00000001
> cat /proc/irq/133/smp_affinity
00000000,00000000,00000000,00000100
> cat /proc/irq/134/smp_affinity
00000000,00000000,00000000,00000002
[...snip...]
> cat /proc/irq/139/smp_affinity
00000000,00000000,00000000,00000800
```

The output identifies that IRQ-132 is the first queue and is routed to CPU0. IRQ-133 is the second queue routed to CPU8, IRQ-134 to CPU2 and so on.

Map TX queue to CPU

Hyperthreaded cores are included with the associated physical core:

```
> echo   110011 > /sys/class/net/eth3/queues/tx-0/xps_cpus
> echo 11001100 > /sys/class/net/eth3/queues/tx-1/xps_cpus
> echo   220022 > /sys/class/net/eth3/queues/tx-2/xps_cpus
> echo 22002200 > /sys/class/net/eth3/queues/tx-3/xps_cpus
> echo   440044 > /sys/class/net/eth3/queues/tx-4/xps_cpus
> echo 44004400 > /sys/class/net/eth3/queues/tx-5/xps_cpus
> echo   880088 > /sys/class/net/eth3/queues/tx-6/xps_cpus
> echo 88008800 > /sys/class/net/eth3/queues/tx-7/xps_cpus
```

Configure Global and Per Queue Tables

- The flow count (number of active connections at any one time) = 32768
- Number of queues = 8 (rss_cpus)
- So the flow count for each queue will be 32768/8

```
> echo 32768 > /proc/sys/net/core/rps_sock_flow_entries
> echo 4096 > /sys/class/net/eth3/queues/rx-0/rps_flow_cnt
> echo 4096 > /sys/class/net/eth3/queues/rx-1/rps_flow_cnt
> echo 4096 > /sys/class/net/eth3/queues/rx-2/rps_flow_cnt
> echo 4096 > /sys/class/net/eth3/queues/rx-3/rps_flow_cnt
> echo 4096 > /sys/class/net/eth3/queues/rx-4/rps_flow_cnt
```

```
> echo 4096 > /sys/class/net/eth3/queues/rx-5/rps_flow_cnt
> echo 4096 > /sys/class/net/eth3/queues/rx-6/rps_flow_cnt
> echo 4096 > /sys/class/net/eth3/queues/rx-7/rps_flow_cnt
```

3.20 Linux Utilities RPM

The Solarflare Linux Utilities RPM contains:

- A boot ROM utility.

  See Configuring the Boot ROM with sfboot on page 72.

- A flash firmware update utility.

  See Upgrading Adapter Firmware with sfupdate on page 86.

- A license key install utility.

  See License Install with sfkey on page 90.

The RPM package, is supplied as 64bit and 32bit binaries compiled to be compatible with GLIBC versions for all supported distributions. The Solarflare utilities RPM file can be downloaded from the following location:

https://support.solarflare.com/

- SF-104451-LS is a 32bit binary RPM package.
- SF-107601-LS is a 64bit binary RPM package.

For example, to install the 64bit package:

1 Download and copy the zipped binary RPM package to the required directory.

2 Unzip the package:

```
# unzip SF-107601-LS-<version>_Solarflare_Linux_Utilities_RPM_64bit.zip
```

3 Install the binary RPM:

```
# rpm -Uvh sfutils-<version>.x86_64.rpm
Preparing...            ########################################### [100%]
1:sfutils                ########################################### [100%]
```

4 Check that the RPM installed correctly:

```
# rpm -q sfutils
sfutils-<version>.x86_64
```

Directions for the use of the utility programs are explained in the following sections.

3.21 Configuring the Boot ROM with sfboot

- Sfboot: Command Usage on page 72.
- Sfboot: Command Line Options on page 73.
- Sfboot: Examples on page 83.

Sfboot is a command line utility for configuring Solarflare adapter Boot ROM options, including PXE and iSCSI booting. Using sfboot is an alternative to using **Ctrl + B** to access the Boot ROM agent during server startup.

See Configuring the Solarflare Boot ROM Agent on page 436 for more information on the Boot Rom agent.

PXE and iSCSI network boot is not supported for Solarflare adapters on IBM System p servers.

Sfboot: SLES 11 Limitation

Due to limitations in SLES 11 using kernel versions prior to 2.6.27.54 it is necessary to reboot the server after running the sfboot utility.

Sfboot: Command Usage

The general usage for sfboot is as follows (as root):

```
sfboot [--adapter=eth<N>] [options] [parameters]
```

When the `--adapter` option is not specified, the `sfboot` command applies to all adapters present in the target host.

The format for the parameters are:

```
<parameter>=<value>
```


Sfboot: Command Line Options

Table 18 lists the options for `sfboot`, Table 19 lists the available global parameters, and Table 20 lists the available per-adapter parameters. Note that command line options are case insensitive and may be abbreviated.

NOTE: Abbreviations in scripts should be avoided, since future updates to the application may render abbreviated scripts invalid.

Table 18: Sfboot Options

Option	Description
-h, `--help`	Displays command line syntax and provides a description of each sfboot option.
-V, `--version`	Shows detailed version information and exits.
-v, `--verbose`	Shows extended output information for the command entered.
-y, `--yes`	Update without prompting.
`-s`, `--quiet` Aliases: `--silent`	Suppresses all output, except errors; no user interaction. The user should query the completion code to determine the outcome of commands when operating silently.
`-l`, `--list`	Lists all available Solarflare adapters. This option shows the ifname and MAC address. Note: this option may not be used in conjunction with any other option. If this option is used with configuration parameters, those parameters will be silently ignored.
`-i`, `--adapter =<ethX>`	Performs the action on the identified Solarflare network adapter. The adapter identifier `ethX` can be the ifname or MAC address, as output by the --`list` option. If `--adapter` is not included, the action will apply to all installed Solarflare adapters.
`-c`, `--clear`	Resets all options to their default values. If an adapter is specified, options for the given adapter are reset, but global options (shown in Table 19) are not reset. Note that `--clear` can also be used with parameters, allowing you to reset to default values, and then apply the parameters specified.

The following global parameters in Table 19 are used to control the configurable parameters for the Boot ROM driver when running prior to the operating system booting.

Table 19: Sfboot Global Parameters

Parameter	Description
`boot-image=` `all|optionrom|uefi|disabled`	Specifies which boot firmware images are served-up to the BIOS during start-up. This parameter can not be used if the `--adapter` option has been specified.
`port-mode=` `default|1x10G|2x10G|4x10G|` `2x40G`	Configure the port mode to use. This is for SFN7000 series adapters only. The values specify the connectors available after using any splitter cables. The usable values are adapter-dependent: • SFN7*xx*2F: 1x10G, 2x10G (default) • SFN7*xx*4F: 2x10G, 4x10G (default) • SFN7*xx*2Q: 2x10G, 4x10G, 2x40G (default) Changes to this setting with `sfboot` require a cold reboot to become effective. MAC address assignments may change after altering this setting.
`firmware-variant=` `full-feature|ultra-low-latency|` `capture-packed-stream|auto`	Configure the firmware variant to use. This is for SFN7000 series adapters only: • the SFN7002F adapter is factory set to full-feature • all other adapters are factory set to auto. Default value = auto - means the driver will select a variant that meets its needs: • the VMware driver always uses full-feature • otherwise, ultra-low-latency is used. The ultra-low-latency variant produces best latency without support for TX VLAN insertion or RX VLAN stripping (not currently used features). It is recommended that Onload customers use the ultra-low-latency variant.
`insecure-filters=` `enabled|disabled`	If enabled bypass filter security on non-privileged functions. This is for SFN7000 series adapters only. This reduces security in virtualized environments. The default is disabled. When enabled a function (PF or VF) can insert filters not qualified by their own permanent MAC address. This is a requirement when using Onload or when using bonded interfaces.

Table 19: Sfboot Global Parameters

Parameter	Description
`mac-spoofing=enabled|disabled`	If enabled, non-privileged functions can create unicast filters for MAC addresses that are not associated with them. This is for SFN7000 series adapters only. The default is disabled. Changes to this setting with `sfboot` require a cold reboot to become effective.
`rx-dc-size=8|16|32|64`	Specifies the size of the descriptor cache for each receive queue. This is for SFN7000 series adapters only. The default is: • 16 if the `port-mode` supports the maximum number of connectors for the adaptor • 32 if the `port-mode` supports a reduced number of connectors.
`tx-dc-size=8|16|32|64`	Specifies the size of the descriptor cache for each transmit queue. This is for SFN7000 series adapters only. The default is: • 32 if the `port-mode` supports the maximum number of connectors for the adaptor • 64 if the `port-mode` supports a reduced number of connectors.
`vi-count=<vi count>`	Sets the total number of virtual interfaces that will be available on the NIC.

The following per-adapter parameters in Table 20 are used to control the configurable parameters for the Boot ROM driver when running prior to the operating system booting.

Table 20: Sfboot Per-adapter Parameters

Parameter	Description
`link-speed=auto|10g|1g|100m`	Specifies the network link speed of the adapter used by the Boot ROM. The default is `auto`. On the 10GBASE-T adapters, `auto` instructs the adapter to negotiate the highest speed supported in common with its link partner. On SFP+ adapters, `auto` instructs the adapter to use the highest link speed supported by the inserted SFP+ module. On 10GBASE-T and SFP+ adapters, any other value specified will fix the link at that speed, regardless of the capabilities of the link partner, which may result in an inability to establish the link. `auto` Auto-negotiate link speed (default) `10G` 10G bit/sec `1G` 1G bit/sec `100M` 100M bit/sec
`linkup-delay=` `<delay time in seconds>`	Specifies the delay (in seconds) the adapter defers its first connection attempt after booting, allowing time for the network to come up following a power failure or other restart. This can be used to wait for spanning tree protocol on a connected switch to unblock the switch port after the physical network link is established. The default is 5 seconds.
`banner-delay=` `<delay time in seconds>`	Specifies the wait period for `Ctrl-B` to be pressed to enter adapter configuration tool. `<delay time in seconds>` = 0-256
`bootskip-delay=` `<delay time in seconds>`	Specifies the time allowed for `Esc` to be pressed to skip adapter booting. `<delay time in seconds>` = 0-256
`boot-type=pxe|iscsi|disabled`	Sets the adapter boot type – effective on next boot. `pxe` – PXE (Preboot eXecution Environment) booting `iscsi` – iSCSI (Internet Small Computer System Interface) booting `disabled` – Disable adapter booting

Table 20: Sfboot Per-adapter Parameters

Parameter	Description

`initiator-dhcp=enabled|disabled`	Enables or disables DHCP address discovery for the adapter by the Boot ROM except for the Initiator IQN (see `initiator-iqn-dhcp`). This option is only valid if iSCSI booting is enabled (`boot-type=iscsi`).
	
	If initiator-DHCP is set to disabled, the following options will need to be specified:
	
	`initiator-ip=<IPv4 address>`
	`netmask=<IPv4 subnet mask>`
	
	The following options may also be needed:
	
	`gateway=<ip_address>`
	`primary-dns=<ip_address>`

`initiator-ip=<IPv4 address>`	Specifies the IPv4 address (in standard "." notation form) to be used by the adapter when `initiator-dhcp` is disabled. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot boot-type=iscsi initiator-dhcp=disabled initiator-ip=<192.168.1.3>`
`netmask=<IPv4 subnet mask>`	Specifies the IPv4 subnet mask (in standard "." notation form) to be used by the adapter when `initiator-dhcp` is disabled. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot boot-type=iscsi initiator-dhcp=disabled netmask=255.255.255.0`
`gateway=<IPv4 address>`	Specifies the IPv4 subnet mask (in standard "." notation form) to be used by the adapter when `initiator-dhcp` is disabled. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot boot-type=iscsi initiator-dhcp=disabled gateway=192.168.0.10`

Table 20: Sfboot Per-adapter Parameters

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>primary-dns=<IPv4 address></td><td>Specifies the IPv4 address (in standard "." notation form) of the Primary DNS to be used by the adapter when initiator-dhcp is disabled.

This option is only valid if iSCSI booting is enabled (boot-type=iscsi).

Example:

sfboot boot-type=iscsi initiator-dhcp=disabled primary-dns=192.168.0.3</td></tr>
<tr><td>initiator-iqn-dhcp=
enabled|disabled</td><td>Enables or disables use of DHCP for the initiator IQN only.</td></tr>
<tr><td>initiator-iqn=<IQN></td><td>Specifies the IQN (iSCSI Qualified Name) to be used by the adapter when initiator-iqn-dhcp is disabled. The IQN is a symbolic name in the "." notation form; for example: iqn.2009.01.com.solarflare, and is a maximum of 223 characters long.

Note that this option is only valid if iSCSI booting is enabled (boot-type=iscsi).

Example:

sfboot initiator-iqn-dhcp=disabled initiator-iqn=iqn.2009.01.com.solarflare adapter=2</td></tr>
<tr><td>lun-retry-count=<retry count></td><td>Specifies the number of times the adapter attempts to access and login to the Logical Unit Number (LUN) on the iSCSI Target before failing. Note that this option is only valid if iSCSI booting is enabled (boot-type=iscsi).

Example:

sfboot lun-retry-count=3</td></tr>
<tr><td>target-dhcp=enabled|disabled</td><td>Enables or disables the use of DHCP to discover iSCSI target parameters on the adapter.

If target-dhcp is disabled, you must specify the following options:

target-server=<DNS name or IPv4 address>
target-iqn=<IQN>
target-port=<port number>
target-lun=<LUN>

Example - Enable the use of DHCP to configure iSCSI Target settings:

sfboot boot-type=iscsi target-dhcp=enabled</td></tr>
</table>

Table 20: Sfboot Per-adapter Parameters

Parameter	Description
`target-server=` `<DNS name or IPv4 address>`	Specifies the iSCSI target's DNS name or IPv4 address to be used by the adapter when `target-dhcp` is disabled. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot boot-type=iscsi target-dhcp=disabled target-server=192.168.2.2`
`target-port=<port number>`	Specifies the Port number to be used by the iSCSI target when `target-dhcp` is disabled. The default Port number is Port 3260. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot boot-type=iscsi target-dhcp=disabled target-port=3262` This option should only be used if your target is using a non-standard TCP Port.
`target-lun=<LUN>`	Specifies the Logical Unit Number (LUN) to be used by the iSCSI target when `target-dhcp` is disabled. The default LUN is 0. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`).
`target-iqn=<IQN>`	Specifies the IQN of the iSCSI target when `target-dhcp` is disabled. Maximum of 223 characters. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Note that if there are spaces contained in `<IQN>`, then the IQN must be wrapped in double quotes (“”). Example: `sfboot target-dhcp=disabled target-iqn=iqn.2009.01.com.solarflare adapter=2`
`vendor-id=<vendor identifier>`	Specifies the device vendor ID to be advertised to the DHCP server. This must match the vendor id configured at the DHCP server when using DHCP option 43 to obtain the iSCSI target.

Table 20: Sfboot Per-adapter Parameters

Parameter	Description

`chap=enabled|disabled`	Enables or disables the use of Challenge Handshake Protocol (CHAP) to authenticate the iSCSI connection.
	
	Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`).
	
	To be valid, this option also requires the following sub-options to be specified:
	
	`username=<initiator username>`
	`secret=<initiator password>`
	
	Example:
	
	`sfboot boot-type=iscsi chap=enabled username=initiatorusername secret=initiatorsecret`

`username=<username>`	Specifies the CHAP initiator username (maximum 64 characters). Note that this option is required if either CHAP or Mutual CHAP is enabled (`chap=enabled, mutual-chap=enabled`). Note that if there are spaces contained in `<username>`, then it must be wrapped in double quotes (“”). Example: `sfboot boot-type=iscsi chap=enabled username=username`
`secret=<secret>`	Specifies the CHAP initiator secret (minimum 12 characters, maximum 20 characters). Note that this option is valid if either CHAP or Mutual CHAP is enabled (chap=enabled, `mutual-chap=enabled`). Note that if there are spaces contained in `<secret>`, then it must be wrapped in double quotes (“”). Example: `sfboot boot-type=iscsi chap=enabled username=username secret=veryverysecret`

Table 20: Sfboot Per-adapter Parameters

Parameter	Description

`mutual-chap=enabled|disabled`	Enables/disables Mutual CHAP authentication when iSCSI booting is enabled.
	
	This option also requires the following sub-options to be specified:
	
	`target-username=<username>`
	`target-secret=<password>`
	`username=<username>`
	`secret=<password>`
	
	Example:
	
	`sfboot boot-type=iscsi mutual-chap=enabled username=username secret=veryverysecret target-username=targetusername target-secret=anothersecret`

`target-username=<username>`	Specifies the username that has been configured on the iSCSI target (maximum 64 characters). Note that this option is necessary if Mutual CHAP is enabled on the adapter (`mutual-chap=enabled`). Note that if there are spaces contained in `<username>`, then it must be wrapped in double quotes (“”).
`target-secret=<secret>`	Specifies the secret that has been configured on the iSCSi target (minimum 12 characters; maximum 20 characters). Note: This option is necessary if Mutual CHAP is enabled on the adapter (`mutual-chap=enabled`). Note that if there are spaces contained in `<secret>`, then it must be wrapped in double quotes (“”).
`mpio-priority=<MPIO priority>`	Specifies the Multipath I/O (MPIO) priority for the adapter. This option is only valid for iSCSI booting over multi-port adapters, where it can be used to establish adapter port priority. The range is 1- 255, with 1 being the highest priority.
`mpio-attempts=<attempt count>`	Specifies the number of times MPIO will try and use each port in turn to login to the iSCSI target before failing.
`pf-count=<pf count>`	This is the number of available PCIe PFs per physical network port. This setting is applied to all ports on the adapter. Changes to this setting with `sfboot` require a cold reboot to become effective. MAC address assignments may change after altering this setting.

Table 20: Sfboot Per-adapter Parameters

Parameter	Description
`msix-limit=` `8|16|32|64|128|256|512|1024`	Specifies the maximum number of MSI-X interrupts that each PF will use. The default is 32. Note: Using the incorrect setting can impact the performance of the adapter. Contact Solarflare technical support before changing this setting.
`sriov=enabled|disabled`	Enable SR-IOV support for operating systems that support this. Not required on SFN7000 series adapters.
`vf-count=<vf count>`	The number of virtual functions (VF) advertised to the operating system for each Physical Function on this physical network port. SFN7000 series adapters have a total limit of 2048 interrupts. Earlier adapters support a total limit of 127 virtual functions per port and a total of 1024 interrupts. Depending on the values of msix-limit and vf-msix-limit, some of these virtual functions may not be configured. Enabling all 127 VFs per port with more than one MSI-X interrupt per VF may not be supported by the host BIOS - in which case you may get 127 VFs on one port and none on others. Contact your BIOS vendor or reduce the VF count. The sriov parameter is implied if vf-count is greater than zero. Changes to this setting with `sfboot` require a cold reboot to become effective.
`vf-msix-limit=` `1|2|4|8|16|32|64|128|256`	The maximum number of interrupts a virtual function may use.

Table 20: Sfboot Per-adapter Parameters

Parameter	Description
`pf-vlans=<tag>[,<tag>[,...]]|none`	Comma separated list of VLAN tags for each PF in the range 0-4094 - see sfboot --help for details.
`switch-mode= default|sriov|partitioning| partitioning-with-sriov|pfiov`	Specifies the mode of operation that the port will be used in: `default` - single PF created, zero VFs created. `sriov` - SR-IOV enabled, single PF created, VFs configured with `vf-count`. `partitioning` - PFs configured with `pf-count`, VFs configured with `vf-count`. See NIC Partitioning on page 58 for details. `partitioning-with-sriov` - SR-IOV enabled, PFs configured with `pf-count`, VFs configured with `vf-count`. See NIC Partitioning on page 58 for details. `pfiov` - PFIOV enabled, PFs configured with `pf-count`, VFs not supported. Changes to this setting with `sfboot` require a cold reboot to become effective.

Sfboot: Examples

- Show the current boot configuration for all adapters:

  `sfboot`

```
# ./sfboot
Solarflare boot configuration utility [v4.3.1]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-2005

eth4:
  Boot image                            Option ROM only
    Link speed                          Negotiated automatically
    Link-up delay time                  5 seconds
    Banner delay time                   2 seconds
    Boot skip delay time                5 seconds
    Boot type                           Disabled
  Physical Functions per port           1
  MSI-X interrupt limit                 32
  Number of Virtual Functions           0
  VF MSI-X interrupt limit              8
  Firmware variant                      full feature / virtualization
  Insecure filters                      Disabled
  VLAN tags                             None
  Switch mode                           Default
```


- List all Solarflare adapters installed on the localhost:

  `sfboot --list`

```
./sfboot -l
Solarflare boot configuration utility [v4.3.1]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-2005
Adapter list:
  eth4
  eth5
```

- Enable iSCSI booting on adapter eth4. Implement default iSCSI settings:

  `sfboot --adapter=eth4 boot-type=iscsi`

```
Solarflare boot configuration utility [v4.3.1]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-2005

eth4:
  Boot image                            Option ROM only
    Link speed                          Negotiated automatically
    Link-up delay time                  5 seconds
    Banner delay time                   2 seconds
    Boot skip delay time                5 seconds
    Boot type                           iSCSI
      Use DHCP for Initiator            Enabled
      Use DHCP for Initiator IQN        Enabled
      LUN busy retries                  2
      Use DHCP for Target               Enabled
      DHCP Vendor Class ID              SFCgPXE
      CHAP authentication               Disabled
      MPIO priority                     0
      MPIO boot attempts                3
  Physical Functions per port           1
  MSI-X interrupt limit                 32
  Number of Virtual Functions           0
  VF MSI-X interrupt limit              8
  Firmware variant                      full feature / virtualization
  Insecure filters                      Disabled
  VLAN tags                             None
  Switch mode                           Default
```

- iSCSI enable adapter eth2. Disable DHCP. Specify adapter IP address and netmask:

  ```
  sfboot boot-type=iscsi --adapter=eth2 initiator-dhcp=disabled
  initiator-ip=192.168.0.1 netmask=255.255.255.0
  ```

```
Solarflare boot configuration utility [v4.3.1]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-2005

eth4:
  Boot image                            Option ROM only
    Link speed                          Negotiated automatically
    Link-up delay time                  5 seconds
    Banner delay time                   2 seconds
    Boot skip delay time                5 seconds
    Boot type                           iSCSI
      Use DHCP for Initiator            Disabled
        Initiator IP address            192.168.0.1
```



```
        Initiator netmask                255.255.255.0
        Initiator default gateway        0.0.0.0
        Initiator primary DNS            0.0.0.0
      Use DHCP for Initiator IQN         Enabled
      LUN busy retries                   2
      Use DHCP for Target                Enabled
      DHCP Vendor Class ID               SFCgPXE
      CHAP authentication                Disabled
      MPIO priority                      0
      MPIO boot attempts                 3
  Physical Functions per port            1
  MSI-X interrupt limit                  32
  Number of Virtual Functions            0
  VF MSI-X interrupt limit               8
  Firmware variant                       full feature / virtualization
  Insecure filters                       Disabled
  VLAN tags                              None
  Switch mode                            Default
```

- Enable SR-IOV (SFN5000 SFN6000 series adapters only)

  `sfboot sriov=enabled vf-count=16 vf-msix-limit=1`

  - SFN7000 Series - Firmware Variant

    `sfboot firmware-variant=full-feature`

```
Solarflare boot configuration utility [v4.3.1]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-2005

eth4:
  Boot image                                Option ROM only
    Link speed                              Negotiated automatically
    Link-up delay time                      7 seconds
    Banner delay time                       3 seconds
    Boot skip delay time                    6 seconds
    Boot type                               PXE
  MSI-X interrupt limit                     32
    Number of Virtual Functions             0
    VF MSI-X interrupt limit                1
    Firmware variant                        full feature / virtualization
```

  - SFN7000 Series - SR-IOV enabled and using Virtual Functions

    `sfboot switch-mode=sriov vf-count=4`

```
Solarflare boot configuration utility [v4.3.1]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-2005

eth4:
  Boot image                             Option ROM only
    Link speed                           Negotiated automatically
    Link-up delay time                   5 seconds
    Banner delay time                    2 seconds
    Boot skip delay time                 5 seconds
    Boot type                            Disabled
  Physical Functions per port            1
  MSI-X interrupt limit                  32
  Number of Virtual Functions            4
  VF MSI-X interrupt limit               8
  Firmware variant                       full feature / virtualization
```



```
Insecure filters                      Disabled
VLAN tags                             None
Switch mode                           SRIOV
```

3.22 Upgrading Adapter Firmware with sfupdate

- Sfupdate: Command Usage on page 86.
- Sfupdate: Command Line Options on page 89.
- Sfupdate: Examples on page 90.

Sfupdate is a command line utility to manage and upgrade the Solarflare adapter Boot ROM, Phy and adapter firmware. Embedded within the sfupdate executable are firmware images for the Solarflare adapter - the exact updates available via sfupdate depend on the specific adapter type.

See Configuring the Solarflare Boot ROM Agent on page 436 for more information on the Boot Rom agent.

CAUTION: All Applications accelerated with OpenOnload should be terminated before updating the firmware with sfupdate.

CAUTION: Solarflare PTP (sfptpd) should be terminated before updated firmware.

Sfupdate: Command Usage

The general usage for sfupdate is as follows (as root):

```
sfupdate [--adapter=eth<N>] [options]
```

where:

- `ethN` is the interface name (ifname) of the Solarflare adapter to be upgraded.
- `option` is one of the command options listed in Table 21.

The format for the options are:

```
<option>=<parameter>
```

Running the command `sfupdate` with no additional parameters will show the current firmware version for all Solarflare adapters and identifies whether the firmware version within sfupdate is more up to date. To update the firmware for all Solarflare adapters run the command `sfupdate --write`

Solarflare recommend the following procedure:

1. Run `sfupdate` to check that the firmware on all adapters is up to date.
2. Run `sfupdate --write` to update the firmware on all adapters.

Sfupdate: Linux MTD Limitations

The driver supplied “inbox” within RedHat and Novell distributions has a limitation on the number of adapters that sfupdate can support. This limitation is removed from RHEL 6.5 onwards. The Solarflare supplied driver is no longer subject to this limitation on any distro/kernel.

Linux kernel versions prior to 2.6.20 support up to 16 MTD (flash) devices. Solarflare adapters are equipped with 6 flash partitions. If more than two adapters are deployed within a system a number of flash partitions will be inaccessible during upgrade.

The limit was raised to 32 in Linux kernel version 2.6.20 and removed altogether in 2.6.35.

If issues are encountered during sfupdate, the user should consider one of the following options when upgrading firmware on systems equipped with more than two Solarflare adapters:

- Upgrade two adapters at a time with the other adapters removed.
- Upgrade the kernel.
- Rebuild the kernel, raising the value of `MAX_MTD_DEVICES` in `include/linux/mtd/mtd.h`.
- Request bootable utilities from support@solarflare.com.

Overcome Linux MTD Limitations

An alternative method is available to upgrade the firmware without removing the adapters.

1 Unbind all interfaces from the drivers:

```
# for bdf in $(lspci -D -d 1924: | awk '{ print $1 }'); do \
  echo -n ${bdf}\  > /sys/bus/pci/devices/${bdf}/driver/unbind; done
```

2 Identify the bus/device/function for all Solarflare interfaces.

Using `ifconfig -a` will not discover any Solarflare interfaces. Use `lspci`:

```
# lspci -D -d 1924:
```

Output similar to the following will be produced (5 NICs installed in this example):

```
# lspci -D -d 1924:
0000:02:00.0 Ethernet controller: Solarflare Communications SFC9020 [Solarstorm]
0000:02:00.1 Ethernet controller: Solarflare Communications SFC9020 [Solarstorm]
0000:03:00.0 Ethernet controller: Solarflare Communications SFC9020 [Solarstorm]
0000:03:00.1 Ethernet controller: Solarflare Communications SFC9020 [Solarstorm]
0000:04:00.0 Ethernet controller: Solarflare Communications SFL9021 [Solarstorm]
0000:04:00.1 Ethernet controller: Solarflare Communications SFL9021 [Solarstorm]
0000:83:00.0 Ethernet controller: Solarflare Communications SFC9020 [Solarstorm]
0000:83:00.1 Ethernet controller: Solarflare Communications SFC9020 [Solarstorm]
0000:84:00.0 Ethernet controller: Solarflare Communications SFC9020 [Solarstorm]
0000:84:00.1 Ethernet controller: Solarflare Communications SFC9020 [Solarstorm]
```


3 There are enough resources to upgrade two NICs at a time, so re-bind interfaces in groups of four (2x2NICs):

```
# echo -n "0000:02:00.0" > /sys/bus/pci/drivers/sfc/bind
# echo -n "0000:02:00.1" > /sys/bus/pci/drivers/sfc/bind
# echo -n "0000:03:00.0" > /sys/bus/pci/drivers/sfc/bind
# echo -n "0000:03:00.1" > /sys/bus/pci/drivers/sfc/bind
```

4 Run sfupdate to update these NICs (command options may vary):

```
# sfupdate --write --yes --force
```

5 Run the command to unbind the interfaces again. There will be failures reported because some of the interfaces are not bound:

```
# for bdf in $(lspci -D -d 1924: | awk '{ print $1 }'); do \
  echo -n ${bdf}\ > /sys/bus/pci/devices/${bdf}/driver/unbind; done
```

6 Repeat the process for the other interfaces (0000:04:00.x; 0000:83:00.x and 0000:84:00.x) doing so in pairs until all the NICs have been upgraded.

7 Rebind all interfaces, doing so en-mass and ignoring errors from those already bound:

```
# for bdf in $(lspci -D -d 1924: | awk '{ print $1 }'); do \
  echo -n ${bdf}\ > /sys/bus/pci/drivers/sfc/bind; done
```

Alternatively reload the sfc driver:

```
# onload_tool reload
```

or:

```
# modprobe -r sfc
# modprobe sfc
```

8 Run `ifconfig -a` again to find that all the interfaces are reported and all have been firmware upgraded without having to physically touch the server or change the kernel.

Sfupdate: SLES 11 Limitation

Due to limitations in SLES 11 using kernel versions prior to 2.6.27.54 it is necessary to reboot the server after running the sfupdate utility to upgrade server firmware.

Sfupdate: Command Line Options

Table 21 lists the options for `sfupdate`.

Table 21: Sfupdate Options

Option	Description
`-h, --help`	Shows help for the available options and command line syntax.
`-i, --adapter=ethX`	Specifies the target adapter when more than one adapter is installed in the localhost. `ethX` = Adapter ifname or MAC address (as obtained with `--list`).
`--list`	Shows the adapter ID, adapter name and MAC address of each adapter installed in the localhost.
`--write`	Re-writes the firmware from the images embedded in the `sfupdate` tool. To re-write using an external image, specify `--image=<filename>` in the command. `--write` fails if the embedded image is the same or a previous version. To force a write in this case, specify `--force` in the command.
`--force`	Force the update of all firmware, even if the installed firmware version is the same as, or more recent then, the firmware embedded in `sfupdate`.
`--backup`	Backup existing firmware image before updating. This option may be used with `--write` and `--force`.
`--image=(filename)`	Update the firmware using the binary image from the given file rather than from those embedded in the utility.
`-y, --yes`	Update without prompting. This option can be used with the `--write` and `--force` options.
`-v, --verbose`	Verbose mode.
`-s, --silent`	Suppress output while the utility is running; useful when the utility is used in a script.
`-V, --version`	Display version information and exit.

Sfupdate: Examples

- Display firmware versions for all adapters:

  `sfupdate`

```
Solarstorm firmware update utility [v4.3.1]
Copyright Solarflare Communications 2006-2013, Level 5 Networks 2002-2005

eth4 - MAC: 00-0F-53-21-00-61
    Controller type:    Solarflare SFC9100-family
    Controller versoin: unknown
    Boot ROM version:   unknown

This utility contains more recent Boot ROM firmware [v4.2.1.1000]
   - run "sfupdate --write" to perform an update
This utility contains more recent controller firmware [v4.2.1.1010]
   - run "sfupdate --write" to perform an update

eth5 - MAC: 00-0F-53-21-00-60
    Controller type:    Solarflare SFC9100-family
    Controller version: unknown
    Boot ROM version:   unknown

This utility contains more recent Boot ROM firmware [v4.2.1.1000]
   - run "sfupdate --write" to perform an update
This utility contains more recent controller firmware [v4.2.1.1010]
   - run "sfupdate --write" to perform an update
```

3.23 License Install with sfkey

The sfkey utility is distributed with the Linux Utilities RPM package. This utility is used to install Solarflare AppFlex™ licenses and enable selected on-board services for Solarflare adapters. For more information about license requirements see Solarflare AppFlex™ Technology Licensing. on page 14.

sfkey: Command Usage

`# sfkey [--adapter=eth<N>] [options]`

If the adapter option is not specified, operations will be applied to all installed adapters.

- To view all sfkey options:

  `# sfkey --help`

- To list (by serial number) all adapters that support licensing:

  `# sfkey --inventory`

- To display an adapter serial number and installed license keys:

  `# sfkey --adapter=eth4 --report`

```
2-inteface adapter: eth4, eth5
Product name:       Solarflare SFN7122F SFP+ Server Adapter
Part number:        SFN7122F
```

```
Serial number:      712200205071133867100441
MAC addresses:      00-0F-53-21-9B-B0, 00-0F-53-22-8B-B1
Installed keys:     Onload, PTP, SolarCapture Pro, SolarSecure Filter
Engine,
Active keys:         Onload, PTP, SolarCapture Pro, SolarSecure Filter
Engine
Blacklisted keys:   0
Invalid keys:       0
Unverifiable keys:  0
Inapplicable keys:  0
```

- To install a license:

  Copy the license key data to a .txt file on the target server. All keys can be in the same key file and the file applied on multiple servers. The following example uses a license key file called key.txt created on the local server.

  ```
  # sfkey --adapter=eth<N> --install key.txt
  ```

  ```
  sfkey firmware update utility: v3.3.3.6330
  Copyright Solarflare Communications 2006-2013, Level 5 Networks 2002-
  2005
  Reading keys...
  Writing keys to eth1...
  Adapter: eth1
  Product name: Solarflare SFN7122F SFP+ Adapter
  Part number: SFN7122F
  Serial number: 712200205071133867100591
  MAC address: 00-0F-53-21-9B-B0
  Installed keys: OpenOnload, PTP, SolarCapture Pro
  Active keys: OpenOnload, PTP, SolarCapture Pro
  Blacklisted keys: 0
  Invalid keys: 0
  ```

License Inventory

Use the combined `--inventory` and `--keys` options to identify the licenses installed on an adapter.

```
sfkey --adapter=eth4 --inventory --keys
eth4, eth5: 712200205071133867100441 (Flareon), $ONL, PTP, SCP, SSFE, !PM,
```

License information is displayed in *[Prefix] [Acronym] [Suffix]* format.

```
  Prefix:           $      Factory-fitted,
   (may be omitted) !      Not present.
  Acronym:          LNA    Line Arbitration,
                    ONL    Onload,
                    PCAP   Packet Capture,
                    PM     Performance Monitor,
                    PTP    Precision Time Protocol,
                    RSE    Resilient Ethernet,
                    SCL    SolarCapture Live,
                    SCP    SolarCapture Pro,
                    SSFE   SolarSecure Filter Engine,
                    CSS    Capture SolarSystem,
                    NAC    Network Access Control,
                    An     Application unknown to this version of sfkey
                           ('n' is a placeholder for the application id).
```

```
Suffix:         <none> Licensed,
                +      Site licensed,
                ~      Evaluation license,
                *      Inactive license,
                @      Inactive site license,
                -      No state available.
```

sfkey Options

Table 22 describes all sfkey options.

Table 22: sfkey options

Option	Description
`--backup`	Output a report of the installed keys in all adapters. The report can be saved to file and later used with the `--install` option.

`--install <filename>`	Install license keys from the given file and report the result. To read from stdin use "-" in place of filename. Keys are installed to an adapter, so if an adapter's ports are eth4 and eth5, both ports will be affected by the keys installed.
	sfc driver reload is required after sfkey installs a PTP license.
	To reload the sfc driver:
	`# modprobe -r sfc; modprode sfc`
	or when Onload is installed:
	`# onload_tool reload`

`--inventory`	List by serial number all adapters that support licensing. By default this will list adapters that support licenses. To list all adapters use the `--all` option. To list keys use the `--keys` option.
`--keys`	Include keys in `--inventory` output - see License Inventory above.
`--noevaluationupdate`	Do not update evaluation keys.
`-a, --all`	Apply sfkey operation to all adapters that support licensing.
`-c, --clear`	Delete all existing license keys from an adapter - except factory installed keys.
`-h, --help`	Display all sfkey options.
`-i, --adapter`	identify specific adapter to apply sfkey operation to.

Table 22: sfkey options

Option	Description
`-r, --report`	Display an adapter serial number and current license status (see example above). Use with `--all` or with `--adapter`. If an installed or active key is reported as 'An' (where n is a number), it indicates a license unknown to this version of sfkey - use an updated sfkey version.
`-s, --silent`	Silent mode, output errors only.
`-v, --verbose`	Verbose mode.
`-V, --version`	Display sfkey version and exit.
`-x, --xml`	Report format as XML.

3.24 Performance Tuning on Linux

Introduction

The Solarflare family of network adapters are designed for high-performance network applications. The adapter driver is pre-configured with default performance settings that have been designed to give good performance across a broad class of applications. Occasionally, application performance can be improved by tuning these settings to best suit the application.

There are three metrics that should be considered when tuning an adapter:

- Throughput
- Latency
- CPU utilization

Different applications may be more or less affected by improvements in these three metrics. For example, transactional (request-response) network applications can be very sensitive to latency whereas bulk data transfer applications are likely to be more dependent on throughput.

The purpose of this section is to highlight adapter driver settings that affect the performance metrics described. This section covers the tuning of all Solarflare adapters.

Latency will be affected by the type of physical medium used: 10GBase-T, twinaxial (direct-attach), fiber or KX4. This is because the physical media interface chip (PHY) used on the adapter can introduce additional latency. Likewise, latency can also be affected by the type of SFP/SFP+/QSFP module fitted.

In addition, you may need to consider other issues influencing performance, such as application settings, server motherboard chipset, CPU speed, cache size, RAM size, additional software installed on the system, such as a firewall, and the specification and configuration of the LAN. Consideration of such issues is not within the scope of this guide.

Tuning settings

Port mode

The selected port mode for SFN7000 series adapters should correspond to the speed and number of connectors in use, after using any splitter cables. If a restricted set of connectors is configured, the driver can then transfer resources from the unused connectors to those configured, potentially improving performance.

Adapter MTU (Maximum Transmission Unit)

The default MTU of 1500 bytes ensures that the adapter is compatible with legacy 10/100Mbps Ethernet endpoints. However if a larger MTU is used, adapter throughput and CPU utilization can be improved. CPU utilization is improved, because it takes fewer packets to send and receive the same amount of data. Solarflare adapters support an MTU of up to 9216 bytes (this does not include the Ethernet preamble or frame-CRC).

Since the MTU should ideally be matched across all endpoints in the same LAN (VLAN), and since the LAN switch infrastructure must be able to forward such packets, the decision to deploy a larger than default MTU requires careful consideration. It is recommended that experimentation with MTU be done in a controlled test environment.

The MTU is changed dynamically using ifconfig, where `ethX` is the interface name and `<size>` is the MTU size in bytes:

```
# /sbin/ifconfig <ethX> mtu <size>
```

Verification of the MTU setting may be performed by running `ifconfig` with no options and checking the MTU value associated with the interface. The change in MTU size can be made to persist across reboots by editing the file `/etc/sysconfig/network-scripts/ifcfg-ethX` and adding `MTU=<mtu>` on a new line.

Interrupt Moderation (Interrupt Coalescing)

Interrupt moderation reduces the number of interrupts generated by the adapter by coalescing multiple received packet events and/or transmit completion events together into a single interrupt.

The *interrupt moderation interval* sets the minimum time (in microseconds) between two consecutive interrupts. Coalescing occurs only during this interval:

- When the driver generates an interrupt, it starts timing the moderation interval.
- Any events that occur before the moderation interval expires are coalesced together into a single interrupt, that is raised only when the interval expires.

  A new moderation interval then starts, during which no interrupt is raised.
- An event that occurs after the moderation interval has expired gets its own dedicated interrupt, that is raised immediately.

  A new moderation interval then starts, during which no interrupt is raised.

Solarflare adapters, by default, use an *adaptive algorithm* where the interrupt moderation delay is automatically adjusted between zero (no interrupt moderation) and 60 microseconds. The adaptive algorithm detects latency sensitive traffic patterns and adjusts the interrupt moderation interval accordingly.

Interrupt moderation settings are **critical for tuning adapter latency**:

- Disabling the adaptive algorithm will:
  - reduce jitter
  - allow setting the moderation interval as required to suit conditions.
- Increasing the interrupt moderation interval will:
  - generate less interrupts
  - reduce CPU utilization (because there are less interrupts to process)
  - increase latency
  - improve peak throughput.
- Decreasing the interrupt moderation interval will:
  - generate more interrupts
  - increase CPU utilization (because there are more interrupts to process)
  - decrease latency
  - reduce peak throughput.
- Turning off interrupt moderation will:
  - generate the most interrupts
  - give the highest CPU utilization
  - give the lowest latency
  - give the biggest reduction in peak throughput.

For many transaction request-response type network applications, the benefit of reduced latency to overall application performance can be considerable. Such benefits typically outweigh the cost of increased CPU utilization. It is recommended that:

- Interrupt moderation is disabled for applications that require best latency and jitter performance, such as market data handling.
- Interrupt moderation is enabled for high throughput single (or few) connection TCP streaming applications, such as iSCSI.

Interrupt moderation can be changed using ethtool, where `ethX` is the interface name. Before adjusting the interrupt moderation interval, it is recommended to disable adaptive moderation:

```
ethtool -C <ethX> adaptive-rx off
```

To set the RX interrupt moderation `interval` in microseconds (µs):

```
ethtool –C <ethX> rx-usecs <interval>
```

To turn off interrupt moderation, set an interval of zero (0):

```
ethtool –C <ethX> rx-usecs 0
```

The above example also sets the transmit interrupt moderation interval, unless the driver module parameter `separate_tx_channels` is enabled. (Normally packet RX and TX completions will share interrupts, so RX and TX interrupt moderation intervals must be equal, and the adapter driver automatically adjusts tx-usecs to match rx-usecs.) Refer to Table 27 on page 113.

To set the TX interrupt moderation interval, if `separate_tx_channels` is enabled:

```
ethtool –C <ethX> tx-usecs <interval>
```

Interrupt moderation settings can be checked using `ethtool –c` .

NOTE: The performance benefits of TCP Large Receive Offload are limited if interrupt moderation is disabled. See TCP Large Receive Offload (LRO) on page 97.

TCP/IP Checksum Offload

Checksum offload moves calculation and verification of IP Header, TCP and UDP packet checksums to the adapter. The driver has all checksum offload features enabled by default. Therefore, there is no opportunity to improve performance from the default.

Checksum offload is controlled using ethtool:

- Receive Checksum:

  ```
  # /sbin/ethtool –K <ethX> rx <on|off>
  ```

- Transmit Checksum:

  ```
  # /sbin/ethtool –K <ethX> tx <on|off>
  ```

Verification of the checksum settings may be performed by running ethtool with the –k option.

NOTE: Solarflare recommend you do not disable checksum offload.

TCP Segmentation Offload (TSO)

TCP Segmentation Offload (TSO) offloads the splitting of outgoing TCP data into packets to the adapter. TSO benefits applications using TCP. Applications using protocols other than TCP will not be affected by TSO.

Enabling TSO will reduce CPU utilization on the transmit side of a TCP connection and improve peak throughput, if the CPU is fully utilized. Since TSO has no effect on latency, it can be enabled at all times. The driver has TSO enabled by default. Therefore, there is no opportunity to improve performance from the default.

TSO is controlled using ethtool:

```
# /sbin/ethtool -K <ethX> tso <on|off>
```

Verification of the TSO settings may be performed by running `ethtool` with the `-k` option.

TCP and IP checksum offloads must be enabled for TSO to work.

NOTE: Solarflare recommend that you do not disable this setting.

TCP Large Receive Offload (LRO)

TCP Large Receive Offload (LRO) is a feature whereby the adapter coalesces multiple packets received on a TCP connection into a single larger packet before passing this onto the network stack for receive processing. This reduces CPU utilization and improves peak throughput when the CPU is fully utilized. The effectiveness of LRO is bounded by the interrupt moderation delay, and is limited if interrupt moderation is disabled (see Interrupt Moderation (Interrupt Coalescing) on page 95). Enabling LRO does not itself negatively impact latency.

NOTE: The Solarflare network adapter driver enables LRO by default. By its design, LRO is of greater benefit when working with smaller packets. For Solarflare adapter, LRO will become disabled if the MTU is set larger than 3979. When the MTU is set larger than 3978, LRO cannot be enabled and will be reported as 'fixed disabled' by ethtool.

NOTE: LRO should **NOT** be enabled when using the host to forward packets from one interface to another. For example, if the host is performing IP routing.

NOTE: It has been observed that as RHEL6 boots the libvirtd daemon changes the default forwarding setting such that LRO is disabled on all network interfaces. This behavior is undesirable as it will potentially lower bandwidth and increase CPU utilization - especially for high bandwidth streaming applications.

To determine if LRO is enabled on an interface:
`ethtool -k ethX`

If IP forwarding is not required on the server, Solarflare recommends either:

– Disabling the libvirtd service (if this is not being used),

– Or, as root before loading the Solarflare driver:
`sysctl -w net.ipv4.conf.default.forwarding=0`
(This command can be loaded into `/etc/rc.local`),

– Or, after loading the Solarflare driver, turn off forwarding for only the Solarflare interfaces and re-enable LRO:
`sysctl -w net.ipv4.conf.ethX.forwarding=0`
`ethtool -K ethX lro on`
(where *X* is the id of the Solarflare interface).

Disabling the libvirtd service is a permanent solution, whereas the other recommendations are temporary and will not persist over reboot.

LRO should not be enabled if IP forwarding is being used on the same interface as this could result in incorrect IP and TCP operation.

LRO can be controlled using the module parameter `lro`. Add the following line to `/etc/modprobe.conf` or add the options line to a file under the `/etc/modprobe.d` directory to disable LRO:

```
options sfc lro=0
```

Then reload the driver so it picks up this option:

```
rmmod sfc
modprobe sfc
```

The current value of this parameter can be found by running:

```
cat /sys/module/sfc/parameters/lro
```

LRO can also be controlled on a per-adapter basis by writing to this file in `sysfs`:

```
/sys/class/net/ethX/device/lro
```

- To disable LRO:

  ```
  echo 0 > /sys/class/net/ethX/device/lro
  ```

- To enable LRO:

  ```
  echo 1 > /sys/class/net/ethX/device/lro
  ```


- To show the current value of the per-adapter LRO state:

  `cat /sys/class/net/ethX/device/lro`

Modifying this file instantly enables or disables LRO, no reboot or driver reload is required. This setting takes precedence over the `lro` module parameter

Current LRO settings can be identified with Linux ethtool e.g.

`ethtool -k ethX`

TCP and IP checksum offloads must be enabled for LRO to work.

TCP Protocol Tuning

TCP Performance can also be improved by tuning kernel TCP settings. Settings include adjusting send and receive buffer sizes, connection backlog, congestion control, etc.

For Linux kernel versions, including 2.6.16 and later, initial buffering settings should provide good performance. However for earlier kernel versions, and for certain applications even on later kernels, tuning buffer settings can significantly benefit throughput. To change buffer settings, adjust the `tcp_rmem` and `tcp_wmem` using the `sysctl` command:

- Receive buffering:

  ```
  sysctl net.ipv4.tcp_rmem="<min> <default> <max>"
  ```

- Transmit buffering:

  ```
  sysctl net.ipv4.tcp_wmem="<min> <default> <max>"
  ```

(`tcp_rmem` and `tcp_wmem` can also be adjusted for IPV6 and globally with the net.ipv6 and net.core variable prefixes respectively).

Typically it is sufficient to tune just the max buffer value. It defines the largest size the buffer can grow to. Suggested alternate values are max=500000 (1/2 Mbyte). Factors such as link latency, packet loss and CPU cache size all influence the affect of the max buffer size values. The minimum and default values can be left at their defaults minimum=4096 and default=87380.

Buffer Allocation Method

The Solarflare driver has a single optimized buffer allocation strategy. This replaces the two different methods controlled with the `rx_alloc_method` driver module parameter which were available using 3.3 and previous drivers.

The net driver continues to expose the `rx_alloc_method` module option, but the value is ignored and it only exists to not break existing customer configurations.

TX PIO

PIO (programmed input/output) describes the process where data is directly transferred by the CPU to or from an I/O device. It is an alternative technique to the I/O device using bus master DMA to transfer data without CPU involvement.

Solarflare 7000 series adapters support TX PIO, where packet s on the transmit path can be “pushed” to the adapter directly by the CPU. This improves the latency of transmitted packets but can cause a very small increase in CPU utilization. TX PIO is therefore especially useful for smaller packets.

The TX PIO feature is enabled by default for packets up to 256 bytes. The maximum packet size that can use PIO can be configured with the driver module option `piobuf_size`.

3.25 Web Server - Driver Optimization

Introduction

The Solarflare net driver from version 4.4.1.1017 on Solarflare SFN7000 series adapters includes optimizations aimed specifically at web service providers and cloud based applications.

Tuning recommendations are documented in Table 23 for users concerned with Content Delivery Networks (CDN), HTTP web hosting application technologies such as HA Proxy, nginx and HTTP web servers.

When tested on the Solarflare SFN7002 and SFN7122 adapters using the recommended driver and firmware with minimal driver/hardware tunings, performance improvements have been observed in the following areas:

- increased the rate at which servers can process new HTTP connections
- increased the rate at which servers can process HTTP requests
- increased sustained throughput when processing large files via HTTP
- improved kernel throughput performance

Customers requiring further details or to access test data should send an email to support@solarflare.com.

Driver Tuning

Whilst most driver enhancements are internal changes, transparent and non-configurable by the user, the following driver module options can be used to tune the driver for particular user applications.

- rss_numa_local

  Using the 4.4.1.1017 driver this option is enabled by default. This will restrict RSS to use CPU cores only on the NUMA node closest to the adapter. This is particularly important for processors supporting DDIO.

  RSS channels not on the local NUMA node can still be accessed using the `ethtool -U` commands to identify a core (action) on which to process the specified ethtool ntuple filter traffic. For example if `rss_cpus=cores`, then an RSS receive channel and associated MSI-X interrupt is created for every core.

- rx_recycle_ring_size

  The default value for the maximum number of receive buffers to recycle pages for has been changed to 512, and in newer drivers will be further increased to 1024.

- rx_copybreak

  A default value of 192 bytes has been selected as the maximum size of packet (bytes) that will be copied directly to the network stack.

Driver module options can be enabled in a user-created file (e.g sfc.conf) in the `/etc/modprobe.d` directory, for example:

```
options sfc rss_numa_local=Y
options sfc rx_recycle_ring_size=512
```

For further descriptions and to list all sfc driver module options:

```
# modinfo sfc
```

nginx Tuning

Table 23: nginx Server Tuning

Tuning	Notes
SO_REUSEPORT	Solarflare testing involving nginx used version v1.7.9 with applied patch to support so_reuseport. See the following link for details: http://forum.nginx.org/read.php?29,241283,241283.
rss_cpus=N	Create N receive queues where N=(number of logical cores)/2. See Receive Side Scaling (RSS) on page 65 for options.

Table 23: nginx Server Tuning

Tuning	Notes
rss_numa_local=1	On SMP systems it is recommended to have all interrupts on the NUMA node local to the Solarflare adapter: rss_numa-local=1, and pin nginx threads to the free CPUs even when these are on the non-local node. When this is not possible, CPU cores can be divided equally between interrupts and nginx threads. rss_numa_local=1 is the default setting.
Pinning threads	Application threads and interrupts should not be pinned to the same CPU cores.
ethtool -C adaptive-rx off	Disable the irq-balance service to prevent re-distribution of interrupts by the kernel. Disable adaptive interrupt moderation before setting the interrupt moderation interval.
ethtool -C rx-usecs 60	Set the interrupt moderation interval. When processing smaller packets it is generally better to set a higher interval i.e. 60µsecs and for larger packets a lower interval or even zero to disable interrupt moderation. See Interrupt Moderation (Interrupt Coalescing) on page 95.

Adapters - Software Support

To benefit from recent driver optimizations, the following (minimum) net driver and firmware versions should be used:

```
# ethtool -i eth<N>
driver: sfc
version: 4.4.1.1017
firmware-version: 4.4.2.1011 rx1 tx1
```

For latency sensitive applications, the adapter firmware variant should be set with the sfboot utility to ultra-low-latency:

```
# sfboot --adapter=eth<N> firmware-variant=ultra-low-latency
```

The ultra-low-latency firmware variant is being used when the output from ethtool (above) shows the *rx1* and *tx1* values.

A reboot of the server is required after changes using sfboot.

3.26 Interrupt Affinity

Interrupt affinity describes the set of host CPUs that may service a particular interrupt.

This affinity therefore dictates the CPU context where received packets will be processed and where transmit packets will be freed once sent. If the application can process the received packets in the same CPU context by being affinitized to the relevant CPU, then latency and CPU utilization can be improved. This improvement is achieved because well tuned affinities reduce inter-CPU communication.

Tuning interrupt affinity is most relevant when MSI-X interrupts and RSS are being used. The irqbalance service, which typically runs by default in most Linux distributions, is a service that automatically changes interrupt affinities based on CPU workload.

In many cases the irqbalance service hinders rather than enhances network performance. It is therefore necessary to disable it and then set interrupt affinities.

- To disable irqbalance permanently, run:

  `/sbin/chkconfig -level 12345 irqbalance off`

- To see whether irqbalance is currently running, run:

  `/sbin/service irqbalance status`

- To disable irqbalance temporarily, run:

  `/sbin/service irqbalance stop`

Once the irqbalance service has been stopped, the Interrupt affinities can be configured manually.

NOTE: The Solarflare driver will evenly distribute interrupts across the available host CPUs (based on the `rss_cpus` module parameter).

To use the Solarflare driver default affinities (recommended), the irqbalance service must be disabled before the Solarflare driver is loaded (otherwise it will immediately overwrite the affinity configuration values set by the Solarflare driver).

Example 1:

How affinities should be manually set will depend on the application. For a single streamed application such as Netperf, one recommendation would be to affinitize all the Rx queues and the application on the same CPU. This can be achieved with the following steps:

1 Determine which interrupt line numbers the network interface uses. Assuming the interface is eth0, this can be done with:

```
# cat /proc/interrupts | grep eth0-
123:      13302          0          0          0       PCI-MSI-X  eth0-0
131:          0         24          0          0       PCI-MSI-X  eth0-1
139:          0          0         32          0       PCI-MSI-X  eth0-2
147:          0          0          0         21       PCI-MSI-X  eth0-3
```


This output shows that there are four channels (rows) set up between four CPUs (columns).

2 Determine the CPUs to which these interrupts are assigned to:

```
# cat /proc/irq/123/smp_affinity
00000000,00000000,00000000,00000000,00000000,00000000,00000000,00000001
#  cat /proc/irq/131/smp_affinity
00000000,00000000,00000000,00000000,00000000,00000000,00000000,00000002
#  cat /proc/irq/139/smp_affinity
00000000,00000000,00000000,00000000,00000000,00000000,00000000,00000004
#  cat /proc/irq/147/smp_affinity
00000000,00000000,00000000,00000000,00000000,00000000,00000000,00000008
```

This shows that `RXQ[0]` is affinitized to `CPU[0]`, `RXQ[1]` is affinitized to `CPU[1]`, and so on. With this configuration, the latency and CPU utilization for a particular TCP flow will be Dependant on that flow's RSS hash, and which CPU that hash resolves onto.

NOTE: Interrupt line numbers and their initial CPU affinity are not guaranteed to be the same across reboots and driver reloads. Typically, it is therefore necessary to write a script to query these values and apply the affinity accordingly.

3 Set all network interface interrupts to a single CPU (in this case `CPU[0]`):

```
# echo 1 >  /proc/irq/123/smp_affinity
# echo 1 >  /proc/irq/131/smp_affinity
# echo 1 >  /proc/irq/139/smp_affinity
# echo 1 >  /proc/irq/147/smp_affinity
```

NOTE: The read-back of `/proc/irq/N/smp_affinity` will return the old value until a new interrupt arrives.

4 Set the application to run on the same CPU (in this case `CPU[0]`) as the network interface's interrupts:

```
# taskset 1 netperf
# taskset 1 netperf  -H <host>
```

NOTE: The use of taskset is typically only suitable for affinity tuning single threaded, single traffic flow applications. For a multi threaded application, whose threads for example process a subset of receive traffic, taskset is not suitable. In such applications, it is desirable to use RSS and Interrupt affinity to spread receive traffic over more than one CPU and then have each receive thread bind to each of the respective CPUs. Thread affinities can be set inside the application with the `shed_setaffinity()` function (see Linux man pages). Use of this call and how a particular application can be tuned is beyond the scope of this guide.

If the settings have been correctly applied, all interrupts from `eth0` are being handled on `CPU[0]`. This can be checked:

```
# cat /proc/interrupts | grep eth0-
123:      13302          0          0          0       PCI-MSI-X  eth0-0
131:          0         24          0          0       PCI-MSI-X  eth0-1
139:          0          0         32          0       PCI-MSI-X  eth0-2
147:          0          0          0         21       PCI-MSI-X  eth0-3
```


Example 2:

An example of affinitizing each interface to a CPU on the same package:

First identify which interrupt lines are servicing which CPU and IO device:

```
# cat /proc/interrupts | grep eth0-
123:      13302          0    1278131          0       PCI-MSI-X  eth0-0
# cat /proc/interrupts | grep eth1-
131:          0         24          0          0       PCI-MSI-X  eth1-0
```

Find CPUs on same package (have same 'package-id'):

```
# more /sys/devices/system/cpu/cpu*/topology/physical_package_id
::::::::::::::
/sys/devices/system/cpu/cpu0/topology/physical_package_id
::::::::::::::
1
::::::::::::::
/sys/devices/system/cpu/cpu10/topology/physical_package_id
::::::::::::::
1
::::::::::::::
/sys/devices/system/cpu/cpu11/topology/physical_package_id
::::::::::::::
0
…
```

Having determined that cpu0 and cpu10 are on package 1, we can assign each ethX interface's MSI-X interrupt to its own CPU on the same package. In this case we choose package 1:

```
# echo 1 >  /proc/irq/123/smp_affinity         # 1hex is bit 0 = CPU0
# echo 400 >  /proc/irq/131/smp_affinity       # 400hex is bit 10 = CPU10
```

Other Considerations

PCI Express Lane Configurations

The PCI Express (PCIe) interface used to connect the adapter to the server can function at different speeds and widths. This is independent of the physical slot size used to connect the adapter. The possible widths are multiples x1, x2, x4, x8 and x16 lanes of (2.5Gbps for PCIe Gen 1, 5.0 Gbps for PCIe Gen 2 and 8.0Gbps for PCIe Gen 3) in each direction. *Solarflare adapters are designed for x8 or x16 lane operation*.

On some server motherboards, choice of PCIe slot is important. This is because some slots (including those that are physically x8 or x16 lanes) may only electrically support x4 lanes. In x4 lane slots, Solarflare PCIe adapters will continue to operate, but not at full speed. The Solarflare driver will warn if it detects that the adapter is plugged into a PCIe slot which electrically has fewer than x8 lanes.

SFN5xxx and SFN6xxx Solarflare adapters require a PCIe Gen 2 x8 slot for optimal operation. Solarflare SFN7xxx series adapters require a PCIe Gen 3 x8 or x16 slot for optimal performance. The Solarflare driver will warn if it detects that the adapter is placed in a sub-optimal slot.

Warning messages can be viewed in `dmesg` from `/var/log/messages`.

The `lspci` command can be used to discover the currently negotiated PCIe lane width and speed:

```
lspci -d 1924: -vv
             02:00.1 Class 0200: Unknown device 1924:0710 (rev 01)
                    ...
                    Link: Supported Speed 2.5Gb/s, Width x8, ASPM L0s, Port 1
                    Link: Speed 2.5Gb/s, Width x8
```

NOTE: The Supported speed may be returned as 'unknown', due to older `lspci` utilities not knowing how to determine that a slot supports PCIe Gen. 2.0/5.0 Gb/s or PCIe Gen 3.0/8,0 Gb/s.

In addition, the latency of communications between the host CPUs, system memory and the Solarflare PCIe adapter may be PCIe slot dependent. Some slots may be “closer” to the CPU, and therefore have lower latency and higher throughput. If possible, install the adapter in a slot which is local to the desired NUMA node

Please consult your server user guide for more information.

CPU Speed Service

Most Linux distributions will have the `cpuspeed` service running by default. This service controls the CPU clock speed dynamically according to current processing demand. For latency sensitive applications, where the application switches between having packets to process and having periods of idle time waiting to receive a packet, dynamic clock speed control may increase packet latency. Solarflare recommend disabling the cpuspeed service if minimum latency is the main consideration.

The service can be disabled temporarily:

```
/sbin/service cpuspeed stop
```

The service can be disabled across reboots:

```
/sbin/chkconfig –level 12345 cpuspeed off
```

CPU Power Service

On RHEL7 systems, `cpuspeed` is replaced with `cpupower`. Solarflare recommend disabling the `cpupower` service if minimum latency is the main consideration. The service is controlled via `systemctl`:

```
systemctl stop cpupower
systemctl disable cpupower
```

Tuned Service

On RHEL7 systems, it may be beneficial to disable the `tuned` service if minimum latency is the main consideration. Users are advised to experiment. The service is controlled via `systemctl`:

```
systemctl stop tuned
systemctl disable tuned
```


Busy poll

If the kernel supports the *busy poll* features (Linux 3.11 or later), and minimum latency is the main consideration, Solarflare recommend that the `busy_poll` socket options should be enabled with a value of 50 microseconds as follows:

```
sysctl net.core.busy_poll=50 && sysctl net.core.busy_read=50
```

Only sockets having a non-zero value for `SO_BUSY_POLL` will be polled, so the user should do one of the following:

- set the poll timeout with the global busy_ read option, as shown above,
- set the per-socket `SO_BUSY_POLL` socket option on selected sockets.

Setting busy_read also sets the default value for the `SO_BUSY_POLL` option.

Memory bandwidth

Many chipsets use multiple channels to access main system memory. Maximum memory performance is only achieved when the chipset can make use of all channels simultaneously. This should be taken into account when selecting the number of memory modules (DIMMs) to populate in the server. For optimal memory bandwidth in the system, it is likely that:

- all DIMM slots should be populated
- all NUMA nodes should have memory installed.

Please consult the motherboard documentation for details.

Intel® QuickData / NetDMA

On systems that support Intel I/OAT (I/O Acceleration Technology) features such as QuickData (a.k.a NetDMA), Solarflare recommend that these are enabled as they are rarely detrimental to performance.

Using Intel® QuickData Technology allows data copies to be performed by the system and not the operating system. This enables data to move more efficiently through the server and provide fast, scalable, and reliable throughput.

Enabling QuickData

- On some systems the hardware associated with QuickData must first be enabled (once only) in the BIOS
- Load the QuickData drivers with `modprobe ioatdma`

Server Motherboard, Server BIOS, Chipset Drivers

Tuning or enabling other system capabilities may further enhance adapter performance. Readers should consult their server user guide. Possible opportunities include tuning PCIe memory controller (PCIe Latency Timer setting available in some BIOS versions).

Tuning Recommendations

The following tables provide recommendations for tuning settings for different applications.

- Throughput - Table 24 on page 108
- Latency - Table 25 on page 109
- Forwarding - Table 26 on page 111

Recommended Throughput Tuning

Table 24 shows recommended tuning settings for throughput:

Table 24: Throughput Tuning Settings

Tuning Parameter	How?
MTU Size	Configure to maximum supported by network: `/sbin/ifconfig <ethX> mtu <size>`
Interrupt moderation	Leave at default (Enabled).
TCP/IP Checksum Offload	Leave at default (Enabled).
TCP Segmentation Offload	Leave at default (Enabled).
TCP Large Receive Offload	Leave at default (Enabled).

TCP Protocol Tuning	Leave at default for 2.6.16 and later kernels.
	For earlier kernels:
	`sysctl net.core.tcp_rmem 4096 87380 524288`
	`sysctl net.core.tcp_wmem 4096 87380 524288`

Receive Side Scaling (RSS)	Application dependent
Interrupt affinity & irqbalance service	Interrupt affinity settings are application dependent Stop irq balance service: `/sbin/service irqbalance stop` Reload the drivers to use the driver default interrupt affinity.
Buffer Allocation Method	Leave at default. Some applications may benefit from specific setting. The Solarflare driver now supports a single optimized buffer allocation strategy and any value set by the rx_alloc_method parameter is ignored.

Table 24: Throughput Tuning Settings

Tuning Parameter	How?
PCI Express Lane Configuration	Ensure the adapter is in an x8 slot (2.0 or later), and that current speed (not the supported speed) reads back as “x8 and 5GT/s”, or “x8 and 8GT/s”, or “x8 and Unknown”.
CPU Speed Service (cpuspeed)	Leave enabled.
Memory bandwidth	Ensure memory utilizes all memory channels on system motherboard.
Intel QuickData (Intel chipsets only)	Enable in BIOS and install driver: `modprobe ioatdma`

Recommended Latency Tuning

Table 25 shows recommended tuning settings for latency:

Table 25: Latency Tuning Settings

Tuning Parameter	How?
MTU Size	Configure to maximum supported by network: `/sbin/ifconfig <ethX> mtu <size>`
Interrupt moderation	Disable with: `ethtool -C <ethX> rx-usecs-irq 0`
TCP/IP Checksum Offload	Leave at default (Enabled).
TCP Segmentation Offload	Leave at default (Enabled).
TCP Large Receive Offload	Disable using sysfs: `echo 0 > /sys/class/net/ethX/device/lro`
TCP Protocol Tuning	Leave at default, but changing does not impact latency.
Receive Side Scaling	Application dependent.
Interrupt affinity & irqbalance service	Interrupt affinity settings are application dependent Stop irq balance service: `/sbin/service irqbalance stop` Reload the drivers to use the driver default interrupt affinity.

Table 25: Latency Tuning Settings

Tuning Parameter	How?
Buffer Allocation Method	Leave at default. Some applications may benefit from specific setting. The Solarflare driver now supports a single optimized buffer allocation strategy and any value set by the rx_alloc_method parameter is ignored.
PCI Express Lane Configuration	Ensure the adapter is in an x8 slot (2.0 or later), and that current speed (not the supported speed) reads back as “x8 and 5GT/s”, or “x8 and 8GT/s”, or “x8 and Unknown”.
CPU Speed Service (cpuspeed)	Disable with: `/sbin/service cpuspeed stop`

CPU Power Service (cpupower)	Disable with:
	`systemctl stop cpupower`
	`systemctl disable cpupower`

Tuned Service	Experiment disabling this with:
	`systemctl stop tuned`
	`systemctl disable tuned`

Busy poll (Linux 3.11 and later)	Enable with a value of 50µs:
	`sysctl net.core.busy_poll=50 \`
	`    && sysctl net.core.busy_read=50`

Memory bandwidth	Ensure memory utilizes all memory channels on system motherboard.
Intel QuickData (Intel chipsets only)	Enable in BIOS and install driver: `modprobe ioatdma`

Recommended Forwarding Tuning

Table 26 shows recommended tuning settings for forwarding

Table 26: Forwarding Tuning Settings

Tuning Parameter	How?
MTU Size	Configure to maximum supported by network: `/sbin/ifconfig <ethX> mtu <size>`

Interrupt moderation	Configure an explicit interrupt moderation interval by setting the following driver options (see Driver Tuning on page 101):
	`irq_adapt_enable=0`
	`tx_irq_mod_usec=150`

TCP/IP Checksum Offload	Leave at default (Enabled).
TCP Segmentation Offload	Leave at default (Enabled).
TCP Large Receive Offload	Disable using sysfs: `echo 0 > /sys/class/net/ethX/device/lro`

TCP Protocol Tuning	Leave at default for 2.6.16 and later kernels.
	For earlier kernels:
	`sysctl net.core.tcp_rmem 4096 87380 524288`
	`sysctl net.core.tcp_wmem 4096 87380 524288`

Receive Side Scaling (RSS)	Leave the `rss_cpus` option at the default, to use all CPUs for RSS. Ensure the `rss_numa_local` driver option is set to its default value of 1 (see Driver Tuning on page 101).
Interrupt affinity & irqbalance service	Interrupt affinity. Affinitize each ethX interface to its own CPU (if possible select CPU's on the same Package). Refer to Interrupt Affinity on page 103. Stop irqbalance service: `/sbin/service irqbalance stop`
Buffer Allocation Method	Leave at default. Some applications may benefit from specific setting. The Solarflare driver now supports a single optimized buffer allocation strategy and any value set by the rx_alloc_method parameter is ignored.

Table 26: Forwarding Tuning Settings

Tuning Parameter	How?
Buffer Recycling	Make receive buffer recycling more aggressive by setting the following driver option (see Driver Tuning on page 101): `rx_recycle_ring_size=256`
PIO	Disable PIO by setting the following driver option (see Driver Tuning on page 101): `piobuf_size=0`
Transmit push	Disable transmit push by setting the following driver option (see Driver Tuning on page 101): `tx_push_max_fill=0`
Direct copying	Disable copying directly from the network stack for transmits by setting the following driver option (see Driver Tuning on page 101): `tx_copybreak=0`

Ring sizes	Change the number of descriptor slots on each ring by setting the following driver options (see Driver Tuning on page 101):
	`tx_ring=512`
	`rx_ring=512`
	Note that as the `tx_irq_mod_usec` interrupt moderation interval increases, the number of required `tx_ring` and `rx_ring` descriptor slots also increases. Insufficient descriptor slots will cause dropped packets.

3.27 Module Parameters

Table 27 lists the available parameters in the Solarflare Linux driver module (`modinfo sfc`):

Table 27: Driver Module Parameters

Parameter	Description	Possible Value	Default Value
sxps_enabled	Enable or disable the Solarflare net driver to perform transmit flow steering. If the kernel does support XPS, this should be enabled in the kernel before using the SARFS feature.	0|1	0
sarfs_table_size	The size of the table used to maintain SARFS filters.	uint	256
sarfs_global_holdoff_ms	The maximum rate at which SARFS will insert or remove filters. This can be increased on heavily loaded servers or decreased to increase responsiveness.	uint	10ms
sarfs_sample_rate	The frequency at which TCP packets are inspected by the SARFS feature. This can be increased on heavily loaded servers to reduce the CPU usage by ARFS. Setting the sample rate to a non-zero value enables the SARFS feature. See also sxps_enabled above. The recommended sample rate is 20.	uint	0 packets
piobuf_size	Identify the largest packet size that can use PIO. Setting this to zero effectively disables PIO	uint	256 bytes
rx_alloc_method	Allocation method used for RX buffers. The Solarflare driver now supports a single optimized buffer allocation strategy and any value set by the rx_alloc_method parameter is ignored. See Buffer Allocation Method on page 99.	uint	AVN(0) new kernels. PAGE(2) old kernels
rx_refill_threshold	RX descriptor ring fast/slow fill threshold (%).	uint	90
lro_table_size[1]	Size of the LRO hash table. Must be a power of 2.	uint	128

Table 27: Driver Module Parameters

Parameter	Description	Possible Value	Default Value
lro_chain_max[1]	Maximum length of chains in the LRO hash table.	uint	20
lro_idle_jiffies[1]	Time (in jiffies) after which an idle connection's LRO state is discarded.	uint	101
lro_slow_start_packets[1]	Number of packets that must pass in-order before starting LRO.	uint	20000
lro_loss_packets[1]	Number of packets that must pass in-order following loss before restarting LRO.	uint	20
rx_desc_cache_size	Set RX descriptor cache size.	int	64
tx_desc_cache_size	Set TX descriptor cache size.	int	16
rx_xoff_thresh_bytes	RX fifo XOFF threshold.	int	-1 (auto)
rx_xon_thresh_bytes	RX fifo XON threshold.	int	-1 (auto)
lro	Large receive offload acceleration	int	1
separate_tx_channels	Use separate channels for TX and RX	uint	0
rss_cpus	Number of CPUs to use for Receive-Side Scaling, or 'packages', 'cores' or 'hyperthreads'	uint or string	<empty>
irq_adapt_enable	Enable adaptive interrupt moderation	uint	1
irq_adapt_low_thresh	Threshold score for reducing IRQ moderation	uint	10000
irq_adapt_high_thresh	Threshold score for increasing IRQ moderation	uint	20000
irq_adapt_irqs	Number of IRQs per IRQ moderation adaptation	uint	1000
napi_weight	NAPI weighting	uint	64
rx_irq_mod_usec	Receive interrupt moderation (microseconds)	uint	60
tx_irq_mod_usec	Transmit interrupt moderation (microseconds)	uint	150
allow_load_on_failure	If set then allow driver load when online self-tests fail	uint	0
onload_offline_selftest	Perform offline self-test on load	uint	1
interrupt_mode	Interrupt mode (0=MSIX, 1=MSI, 2=legacy)	uint	0
falcon_force_internal_sram	Force internal SRAM to be used	int	0

Table 27: Driver Module Parameters

Parameter	Description	Possible Value	Default Value
rss_numa_local	Constrain RSS to use CPU cores on the NUMA node local the Solarflare adapter. Set to 1 to restrict, 0 otherwise.	0|1	1
max_vfs	Enable VFs in the net driver. When specified as a single integer the VF count will be applied to all PFs. When specified as a comma separated list, the first VF count is assigned to the PF with the lowest index i.e. the lowest MAC address, then the PF with the next highest MAC address etc.	uint	0

1. Check OS documentation for availability on SUSE and RHEL versions.

3.28 Linux ethtool Statistics

The Linux command `ethtool` will display an extensive range of statistics originated from the MAC on the Solarflare network adapter. To display statistics use the following command:

```
ethtool -S ethX
```

(where X is the ID of the Solarflare interface)

Using a Solarflare net driver earlier than version 4.4.1.1017, the ethtool statistics counters can be reset by reloading the sfc driver:

```
# modprobe -r sfc
# modprobe sfc
```

Drivers from version 4.4.1.1017 (included in onload-201502) have to manage multi-PF configurations and for this reason statistics are not reset by reloading the driver. The only methods currently available to reset stats is to cold-reboot (power OFF/ ON) the server or reload the firmware image.

Per port statistics (`port_`) are from the physical adapter port. Other statistics are from the specified PCIe function.

Table 28 below lists the complete output from the `ethtool -S` command.

NOTE: `ethtool -S` output depends on the features supported by the adapter type

.

Table 28: Ethtool -S output

Field	Description
port_tx_bytes	Number of bytes transmitted.
port_tx_packets	Number of packets transmitted.
port_tx_pause	Number of pause frames transmitted with valid pause op_code.
port_tx_control	Number of control frames transmitted. Does not include pause frames.
port_tx_unicast	Number of unicast packets transmitted. Includes flow control packets.
port_tx_multicast	Number of multicast packets transmitted.
port_tx_broadcast	Number of broadcast packets transmitted.
port_tx_lt64	Number of frames transmitted where the length is less than 64 bytes.
port_tx_64	Number of frames transmitted where the length is exactly 64 bytes.
port_tx_65_to_127	Number of frames transmitted where the length is between 65 and 127 bytes
port_tx_128_to_255	Number of frames transmitted where the length is between 128 and 255 bytes
port_tx_256_to_511	Number of frames transmitted where the length is between 256 and 511 bytes
port_tx_512_to_1023	Number of frames transmitted where length is between 512 and 1023 bytes
port_tx_1024_to_15xx	Number of frames transmitted where the length is between 1024 and 1518 bytes (1522 with VLAN tag).
port_tx_15xx_to_jumbo	Number of frames transmitted where length is between 1518 bytes (1522 with VLAN tag) and 9000 bytes.
port_rx_bytes	Number of bytes received. Not include collided bytes.
port_rx_good_bytes	Number of bytes received without errors. Excludes bytes from flow control packets.

Table 28: Ethtool -S output

Field	Description
port_rx_bad_bytes	Number of bytes with invalid FCS. Includes bytes from packets that exceed the maximum frame length.
port_rx_packets	Number of packets received.
port_rx_good	Number of packets received with correct CRC value and no error codes.
port_rx_bad	Number of packets received with incorrect CRC value.
port_rx_pause	Number of pause frames received with valid pause op_code.
port_rx_control	Number of control frames received. Does not include pause frames.
port_rx_unicast	Number of unicast packets received.
port_rx_multicast	Number of multicast packets received.
port_rx_broadcast	Number of broadcasted packets received.
port_rx_lt64	Number of packets received where the length is less than 64 bytes.
port_rx_64	Number of packets received where the length is exactly 64 bytes.
port_rx_65_to_127	Number of packets received where the length is between 65 and 127 bytes.
port_rx_128_to_255	Number of packets received where the length is between 128 and 255 bytes.
port_rx_256_to_511	Number of packets received where the length is between 256 and 511 bytes.
port_rx_512_to_1023	Number of packets received where the length is between 512 and 1023 bytes.
port_rx_1024_to_15xx	Number of packets received where the length is between 1024 and 1518 bytes (1522 with VLAN tag).
port_rx_15xx_to_jumbo	Number of packets received where the length is between 1518 bytes (1522 with VLAN tag) and 9000 bytes.

Table 28: Ethtool -S output

Field	Description
port_rx_gtjumbo	Number of packets received where the length is greater than 9000 bytes.
port_rx_bad_gtjumbo	Number of packets received where the length is greater than 9000 bytes, but with incorrect CRC value.
port_rx_overflow	Number of packets dropped by receiver because of FIFO overrun.
port_rx_nodesc_drop_cnt port_rx_nodesc_drops	Number of packets dropped by the network adapter because of a lack of RX descriptors in the RX queue. Packets can be dropped by the NIC when there are insufficient RX descriptors in the RX queue to allocate to the packet. This problem occurs if the receive rate is very high and the network adapter receive cycle process has insufficient time between processing to refill the queue with new descriptors. A number of different steps can be tried to resolve this issue: • Disable the irqbalance daemon in the OS • Distribute the traffic load across the available CPU/cores by setting rss_cpus=cores. Refer to Receive Side Scaling section • Increase receive queue size using ethtool.
port_rx_pm_trunc_bb_overflow	Overflow of the packet memory burst buffer - should not occur.
port_rx_pm_discard_bb_overflow	Number of packets discarded due to packet memory buffer overflow.

Table 28: Ethtool -S output

Field	Description
port_rx_pm_trunc_vfifo_full	Number of packets truncated or discarded because there was not enough packet memory available to receive them. Happens when packets cannot be delivered as quickly as they arrive due to: • packet rate exceeds maximum supported by the adapter. • adapter is inserted into a low speed or low width PCI slot – so the PCIe bus cannot support the required bandwidth. • packets are being replicated by the adapter and the resulting bandwidth cannot be handled by the PCIe bus. • host memory bandwidth is being used by other devices resulting in poor performance for the adapter.
port_rx_pm_discard_vfifo_full	Count of the number of packets dropped because of a lack of main packet memory on the adapter to receive the packet into.
port_rx_pm_trunc_qbb	Not currently supported.
port_rx_pm_discard_qbb	Not currently supported.
port_rx_pm_discard_mapping	Number of packets dropped because they have an 802.1p priority level configured to be dropped
port_rx_dp_q_disabled_packets	Increments when the filter indicates the packet should be delivered to a specific rx queue which is currently disabled due to configuration error or error condition.
port_rx_dp_di_dropped_packets	Number of packets dropped because the filters indicate the packet should be dropped. Can happen because: • the packet does not match any filter. • the matched filter indicates the packet should be dropped.

Table 28: Ethtool -S output

Field	Description
port_rx_dp_streaming_packets	Number of packets directed to RXDP streaming bus which is used if the packet matches a filter which directs it to the MCPU. Not currently used.
port_rx_dp_hlb_fetch	Count the number of times the adapter descriptor cache is empty when a new packet arrives, for which the adapter must do an emergency fetch to replenish the cache with more descriptors.
port_rx_dp_hlb_wait	Increments each time the adapter has done an hlb_fetch which has not yet completed.
rx_unicast	Number of unicast packets received.
rx_unicast_bytes	Number of unicast bytes received.
rx_multicast	Number of multicast packets received.
rx_multicast_bytes	Number of multicast bytes received.
rx_broadcast	Number of broadcast packets received.
rx_broadcast_bytes	Number of broadcast bytes received.
rx_bad	Number of packets received with incorrect CRC value.
rx_bad_bytes	Number of bytes received from packets with incorrect CRC value.
rx_overflow	Number of packets dropped by receiver because of FIFO overrun.
tx_unicast	Number of unicast packets transmitted.
tx_unicast_bytes	Number of unicast bytes transmitted.
tx_multicast	Number of multicast packets transmitted.
tx_multicast_bytes	Number of multicast bytes transmitted.
tx_broadcast	Number of broadcast packets transmitted.
tx_broadcast_bytes	Number of broadcast bytes transmitted.
tx_bad	0.
tx_bad_bytes	0.

Table 28: Ethtool -S output

Field	Description
tx_overflow	Number of packets dropped by transmitter because of FIFO overrun.
tx_merge_events	The number of TX completion events where more than one TX descriptor was completed.
tx_tso_bursts	Number of times when outgoing TCP data is split into packets by the adapter driver. Refer to TCP Segmentation Offload (TSO) on page 97.
tx_tso_long_headers	Number of times TSO is applied to packets with long headers.
tx_tso_packets	Number of physical packets produced by TSO.
tx_pushes	Number of times a packet descriptor is 'pushed' to the adapter from the network adapter driver.
tx_pio_packets	Number of packets sent using PIO.
rx_reset	0
rx_tobe_disc	Number of packets marked by the adapter to be discarded because of one of the following: • Mismatch unicast address and unicast promiscuous mode is not enabled. • Packet is a pause frame. • Packet has length discrepancy. • Due to internal FIFO overflow condition. • Length < 60 bytes.
rx_ip_hdr_chksum_err	Number of packets received with IP header Checksum error.
rx_tcp_udp_chksum_err	Number of packets received with TCP/UDP checksum error.

Table 28: Ethtool -S output

Field	Description
rx_eth_crc_err	Number of packets received whose CRC did not match the internally generated CRC value. This is the total of all receive channels receiving CRC errors.
rx_mcast_mismatch	Number of unsolicited multicast packets received. Unwanted multicast packets can be received because a connected switch simply broadcasts all packets to all endpoints or because the connected switch is not able or not configured for IGMP snooping - a process from which it learns which endpoints are interested in which multicast streams.
rx_frm_trunc	Number of frames truncated because an internal FIFO is full. As a packet is received it is fed by the MAC into a 128K FIFO. If for any reason the PCI interface cannot keep pace and is unable to empty the FIFO at a sufficient rate, the MAC will be unable to feed more of the packet to the FIFO. In this event the MAC will truncate the frame - marking it as such and discard the remainder. The driver on seeing a 'partial' packet which has been truncated will discard it.
rx_merge_events	Number of RX completion events where more than one RX descriptor was completed.
rx_merge_packets	Number of packets delivered to the host through merge events.
tx-0.tx_packets	Per TX queue transmitted packets.
tx_1.tx_packets	Per TX queue transmitted packets.
rx_0.rx_packets	Per RX queue received packets.
rx_1.rx_packets	Per RX queue received packets.

Table 28: Ethtool -S output

Field	Description
rx_no_skb_drops	Number of packets dropped by the adapter when there are insufficient socket buffers available to receive packets into. See also port_rx_nodesc_drop_cnt and port_rx_nodesc_drops above.
rx_nodesc_trunc	Number of frames truncated when there are insufficient descriptors to receive data into. Truncated packets will be discarded by the adapter driver.
ptp_good_syncs	These PTP stats counters relate to the mechanism used by sfptpd to synchronize the system clock and adapter clock(s) in a server. For each synchronization event sfptpd will select a number of system clock times to be compared to the adapter clock time. If the times can be synchronized, the good_syncs counter is incremented, otherwise the bad_syncs counter is incremented. If sfptpd is unable to synchronize the clocks at this event, the sync_timeout counter is incremented. sfptpd will synchronize clocks 16 times per second - so incrementing counters does not necessarily indicate bad synchronization between local server clocks and an external PTP master clock.
ptp_fast_syncs	
ptp_bad_syncs	
ptp_sync_timeouts	
ptp_no_time_syncs	
ptp_invalid_sync_windows	
ptp_undersize_sync_windows	
ptp_oversize_sync_windows	
ptp_rx_no_timestamp	Number of PTP packets received for which a hardware timestamp was not recovered from the adapter.
ptp_tx_timestamp_packets	Number of PTP packets transmitted for which the adapter generated a hardware timestamp.
ptp_rx_timestamp_packets	Number of PTP packets received for which the adapter generated a hardware timestamp.
ptp_timestamp_packets	Total number of PTP packets for which the adapter generated a hardware timestamp.

Table 28: Ethtool -S output

Field	Description
ptp_filter_matches	Number of PTP packets hitting the PTP filter.
ptp_non_filter_matches	Number of PTP packets which did not match the PTP filter.

NOTE: The adapter will double count packets less that 64bytes (`port_rx_lt64`) as also being a CRC error. This can result in `port_rx_bad` => `rx_eth_crc_err` counter. The difference should be equal to the `port_rx_lt64` counter.

3.29 Driver Logging Levels

For the Solarflare net driver, two settings affect the verbosity of log messages appearing in dmesg output and `/var/log/messages`:

- The kernel console log level
- The netif message per network log level

The kernel console log level controls the overall log message verbosity and can be set with the command `dmesg -n` or through the `/proc/sys/kernel/printk` file:

```
echo 6 > /proc/sys/kernel/printk
```

Refer to ‘man 2 syslog’ for log levels and `Documentation/sysctl/kernel.txt` for a description of the values in `/proc/sys/kernel/printk`.

The netif message level provides additional logging control for a specified interface. These message levels are documented in `Documentation/networking/netif-msg.txt`. A message will only appear on the terminal console if both the kernel console log level and netif message level requirements are met.

The current netif message level can be viewed using the following command:

```
ethtool <iface> | grep -A 1 'message level:'
        Current message level: 0x000020f7 (8439)
                               drv probe link ifdown ifup rx_err tx_err hw
```

Changes to the netif message level can be made with ethtool. Either by name:

```
ethtool -s <iface> msglvl rx_status on
```

or by bit mask:

```
ethtool -s <iface> msglvl 0x7fff
```

The initial setting of the netif msg level for all interfaces is configured using the debug module parameter e.g.

```
modprobe sfc debug=0x7fff
```

```
ethtool <iface> | grep -A 1 'message level:'
        Current message level: 0x00007fff (32767)
                               drv probe link timer ifdown ifup rx_err
tx_err tx_queued intr tx_done rx_status pktdata hw wol
```

3.30 Running Adapter Diagnostics

You can use ethtool to run adapter diagnostic tests. Tests can be run offline (default) or online. Offline runs the full set of tests, which can interrupt normal operation during testing. Online performs a limited set of tests without affecting normal adapter operation.

As root user, enter the following command:

```
ethtool --test ethX offline|online
```

The tests run by the command are as follows:

Table 29: Adapter Diagnostic Tests

Diagnostic Test	Purpose
core.nvram	Verifies the flash memory ‘board configuration’ area by parsing and examining checksums.
core.registers	Verifies the adapter registers by attempting to modify the writable bits in a selection of registers.
core.interrupt	Examines the available hardware interrupts by forcing the controller to generate an interrupt and verifying that the interrupt has been processed by the network driver.
tx/rx.loopback	Verifies that the network driver is able to pass packets to and from the network adapter using the MAC and Phy loopback layers.
core.memory	Verifies SRAM memory by writing various data patterns (incrementing bytes, all bit on and off, alternating bits on and off) to each memory location, reading back the data and comparing it to the written value.
core.mdio	Verifies the MII registers by reading from PHY ID registers and checking the data is valid (not all zeros or all ones). Verifies the MMD response bits by checking each of the MMDs in the Phy is present and responding.

Table 29: Adapter Diagnostic Tests

Diagnostic Test	Purpose
chanX eventq.poll	Verifies the adapter's event handling capabilities by posting a software event on each event queue created by the driver and checking it is delivered correctly. The driver utilizes multiple event queues to spread the load over multiple CPU cores (RSS).
phy.bist	Examines the PHY by initializing it and causing any available built-in self tests to run.

3.31 Running Cable Diagnostics

Cable diagnostic data can be gathered from the Solarflare 10GBASE-T adapters physical interface using the `ethtool -t` command which runs a comprehensive set of diagnostic tests on the controller, PHY, and attached cables. To run the cable tests enter the following command:

```
ethtool -t ethX [online | offline]
```

Online tests are non-intrusive and will not disturb live traffic.

The following is an extract from the output of the ethtool diagnostic offline tests:

```
phy    cable.pairA.length        9
phy    cable.pairB.length        9
phy    cable.pairC.length        9
phy    cable.pairD.length        9
phy    cable.pairA.status        1
phy    cable.pairB.status        1
phy    cable.pairC.status        1
phy    cable.pairD.status        1
```

Cable length is the estimated length in metres. A length value of 65535 indicates length not estimated due to pair busy or cable diagnostic routine not completed successfully.

The cable status can be one of the following values:

0 - invalid, or cable diagnostic routine did not complete successfully

1 - pair ok, no fault detected

2 - pair open or Rt > 115 ohms

3 - intra pair short or Rt < 85 ohms

4 - inter pair short or Rt < 85 ohms

9 - pair busy or link partner forces 100Base-Tx or 1000Base-T test mode.

4 Solarflare Adapters on Windows

This chapter covers the following topics on the Microsoft Windows® platform:

4.1 System Requirements

- Refer to Software Driver Support on page 13 for details of supported Windows versions.
- The optional Solarflare Adapter Manager utility requires Microsoft .NET Framework 3.5 on all supported Windows versions.

4.2 Windows Feature Set

Table 30 lists the features supported by Solarflare adapters on Windows.

Users should refer to Microsoft documentation to check feature availability and support on specific Windows OS versions.

Table 30: Solarflare Windows Features

Jumbo frames	Solarflare adapters support MTUs (Maximum Transmission Units) from 1500 bytes to 9216 bytes. • See Ethernet Frame Length on page 156 • See Configuring Network Adapter Properties in Windows on page 177
Task offloads	Solarflare adapters support Large Segmentation Offload (LSO), Receive Segment Coalescing (RSC), and TCP/UDP/IP checksum offload for improved adapter performance and reduced CPU processing requirements. • See Segmentation Offload on page 154 • See Configuring Network Adapter Properties in Windows on page 177
Receive Side Scaling (RSS)	Solarflare adapters support RSS multi-core load distribution technology. • See Using SAM to View Statistics and State Information on page 165 • See Configuring Network Adapter Properties in Windows on page 177
Interrupt Moderation	Solarflare adapters support Interrupt Moderation to reduce the number of interrupts on the host processor from packet events. • See RSS and Interrupts on page 152 • See Configuring Network Adapter Properties in Windows on page 177

Table 30: Solarflare Windows Features

Teaming and/or Link Aggregation	Improve server reliability and bandwidth by bonding physical ports, from one or more Solarflare adapters, into a team, having a single MAC address and which function as a single port providing redundancy against a single point of failure. • See Using SAM to Configure Teams and VLANs on page 157 • See Sfteam: Adapter Teaming and VLAN Tool on page 200 • See Teaming and VLANs on page 213
Virtual LANs (VLANs)	Support for multiple VLANs per adapter: • See Using SAM to Configure Teams and VLANs on page 157 • See Sfteam: Adapter Teaming and VLAN Tool on page 200 • See Teaming and VLANs on page 213
PXE and iSCSI booting	Solarflare adapters support PXE and iSCSI booting, enabling diskless systems to boot from a remote target operating system. • See Using SAM for Boot ROM Configuration on page 170 • See Sfboot: Boot ROM Configuration Tool on page 183 • See Solarflare Boot ROM Agent on page 435
Fault diagnostics	Solarflare adapters provide comprehensive adapter and cable fault diagnostics and system reports. • See Using SAM to Run Adapter and Cable Diagnostics on page 166 • See Sfcable: Cable Diagnostics Tool on page 205
Firmware updates	Solarflare adapters support adapter firmware upgrades. • See Sfupdate: Firmware Update Tool on page 197

Table 30: Solarflare Windows Features

State and statistics analysis	Solarflare adapters provide comprehensive state and statistics information for data transfer, device, MAC, PHY and other adapter features. • See Using SAM to View Statistics and State Information on page 165 • See Sfteam: Adapter Teaming and VLAN Tool on page 200 for teaming statistics. • See Sfnet on page 208 for per interface statistics.
VMQ	Solarflare drivers support static VMQ for Windows Server 2008 R2 and Dynamic VMQ on Windows Server 2012 See Virtual Machine Queue on page 156.

4.3 Installing the Solarflare Driver Package on Windows

- Adapter Drivers Only Installation on page 130
- Full Solarflare Package Installation on page 132
- Repair, Remove and Change Drivers and Utilities on page 134

NOTE: The Solarflare adapter should be physically inserted before installing the drivers. See Installation on page 22.

The user must have administrative rights to install adapter drivers and may be prompted to enter an administrator user name and password.

If Windows attempts to install the drivers automatically, cancel the Windows New Hardware Found wizard and follow the instructions below.

Solarflare does not recommend installing drivers via Remote Desktop Protocol (RDP). For example via Terminal Services.

The drivers install package is named after the Solarflare document part number e.g.

SF-107785-LS-2_Solarflare_Windows_x64_64-bit_Driver_Package.exe

This can be renamed e.g. setup.exe before use.

4.4 Adapter Drivers Only Installation

The steps below describe how to install only the Solarflare adapter drivers in Windows. To install the drivers from the command line, see Install Drivers and Options From a Windows Command Prompt on page 135.

1 Double-click the supplied `Setup.exe`. to start the Solarflare Driver Package Setup wizard. If prompted, confirm your administrator privileges to continue installing the drivers.

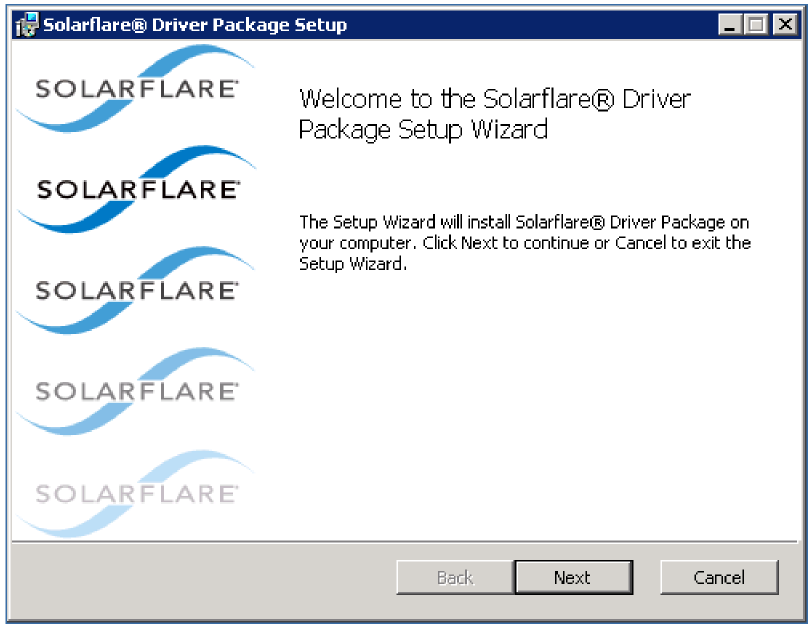


Figure 10: Solarflare Driver Package Setup

2 From the Custom Setup screen, select the **Install Solarflare® device drivers** option only.

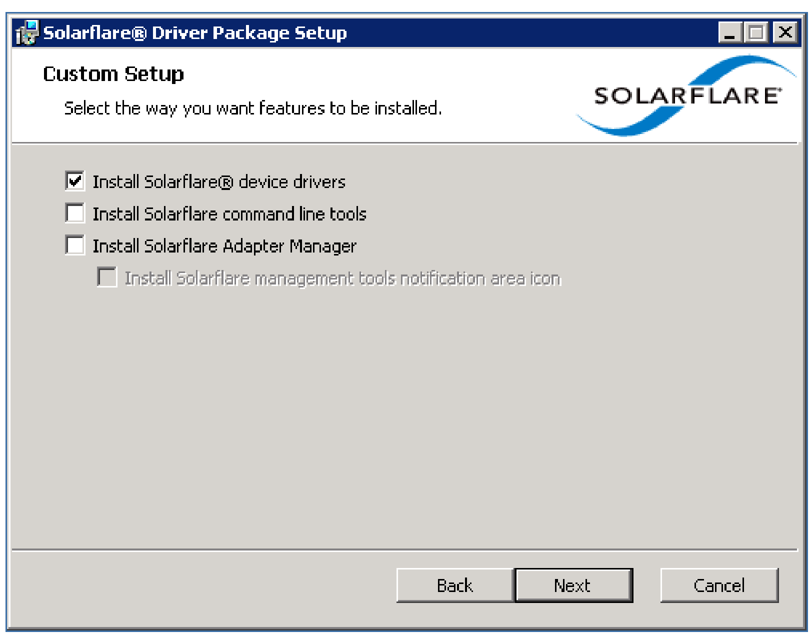


Figure 11: Solarflare Custom Setup

3 Click **Finish** to close the wizard. Restart Windows if prompted to do so.

4.5 Full Solarflare Package Installation

This section cover the following topics:

Prerequisites

- The Solarflare Adapter Manager Utility (SAM) requires Microsoft .NET Framework 3.5 assemblies. These are available by installing .NET version 3.5 and may also be available in version 4.x with backward compatibility for 3.5.

To install the required components from Powershell prompt (Windows Server editions only):

```
Install-WindowsFeature NET-Framework-Core
```

Solarflare Package Installation Procedure

The steps below describe how to install the complete Solarflare installation package. To install this from the command line, see Install Drivers and Options From a Windows Command Prompt on page 135.

1 Double-click the supplied `Setup.exe`. The Solarflare Driver Package Setup wizard starts.

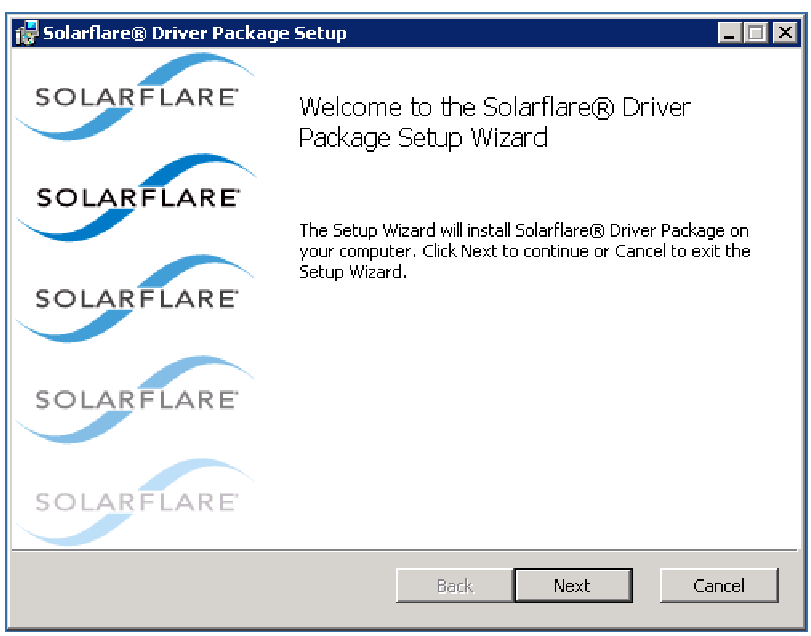


Figure 12: Solarflare Driver Package Setup

If prompted, confirm your administrator privileges to continue installing the drivers.

2 Follow the setup instructions in the wizard to complete the driver installation procedure. See Figure 13:

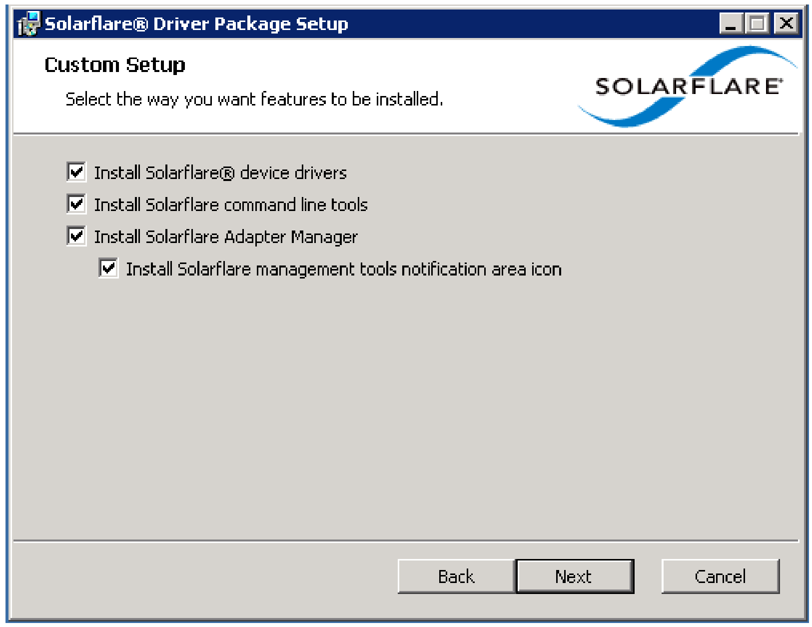


Figure 13: Solarflare Driver Package Custom Setup

Table 31 lists the setup options:

Table 31: Solarflare Custom Setup

Option	Description
Install Solarflare device drivers	Installs Solarflare drivers for Windows. The Solarflare drivers are installed by default.
Install Solarflare command line tools	Installs the following Solarflare Windows command line tools: `sfboot.exe` – Boot ROM configuration tool `sfupdate.exe` – Firmware update tool `sfteam.exe` – Adapter teaming tool `sfcable.exe` – Cable diagnostics tool `sfnet.exe` – Adapter configuration tool See Windows Command Line Tools on page 182. These tools are installed by default.

Table 31: Solarflare Custom Setup

Option	Description
Install Solarflare Adapter Manager	Installs Solarflare Adapter Manager (SAM) for easy access to adapter configuration options, wizards for teaming and VLAN setup, adapter statistics, and diagnostic tools. See Managing Adapters with SAM on page 143 for more details. SAM is installed by default. Note: If this option is grayed out, you need to exit the Solarflare installer and then install Microsoft .NET Framework 3.5 before re-running the Solarflare installer.
Install Solarflare management tools notification area icon	Installs a Solarflare notification area icon for launching Solarflare Adapter Manager (SAM) locally or for a remote computer. The icon is not installed by default.

3 Click **Finish** to close the wizard. Restart Windows if prompted to do so.

To confirm the drivers installed correctly, do either of the following:

- Open the Windows Device Manager and check the Solarflare adapter is present under Network Adapters.
- Start Solarflare Adapter Manager (**Start > All Programs > Solarflare Drivers > Solarflare Adapter Manager**). If the Solarflare adapter is installed and working correctly, it will be shown in the SAM main screen, along with any other adapters, as in Table 15 on page 144.

Repair, Remove and Change Drivers and Utilities

From the **Control Panel > Programs > Programs and Features**, select the **Solarflare Driver Package** then select **Uninstall**, **Change** or **Repair** from the menu bar above the program list.

4.6 Install Drivers and Options From a Windows Command Prompt

This section covers the following subjects:

Command Line Usage on page 135

Using ADDLOCAL on page 137

Command Line Usage

To view command line options available, run the setup-<release>.exe /? command to extract files using the Solarflare Setup Bootstrapper. When this has completed the Solarflare Driver Package Setup Window will be displayed.

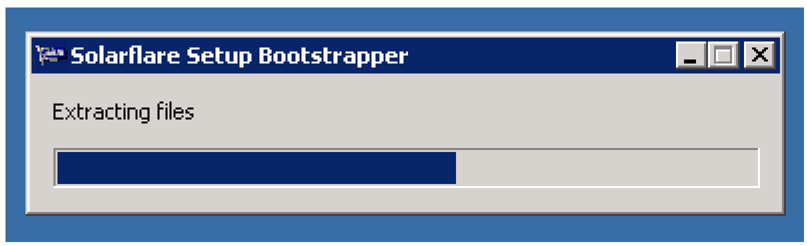


Figure 14: Command Line Install.

Installing from the Windows command line allows scripted, silent and unattended installation of the core Solarflare drivers and package utilities. The drivers install package is named after the Solarflare document part number e.g.

SF-107785-LS-2_Solarflare_Windows_x64_64-bit_Driver_Package.exe

This can be renamed e.g setup.exe before invoking from the command line.

The following example will install default package options silently with no message output:

```
setup.exe /Quiet /Install
```

Table 32 lists other command line examples. Note that command line options are case insensitive, so `/install` and `/INSTALL` are the same.

Table 32: Solarflare Installation Options

Example	Action
`setup.exe /Admininstall <path>`	Allows an administrator to unpack and install the package to a network share and to specify which features of the package can be installed by users.
`setup.exe /Extract <path>`	Extracts the contents of `setup.exe` to the specified path.

Table 32: Solarflare Installation Options

Example	Action
`setup.exe /ExtractDrivers <path>`	Extract the adapter driver to the specified path.
`setup.exe /Filename <filename>`	Log all output to the specified file.
`setup.exe /Force`	Allow passive or quiet mode to replace an existing installation with an earlier version.
`setup.exe /Help`	Shows a help screen and exits.
`setup.exe /Install`	Installs or configures the package.
`setup.exe /Install /Log <filename>`	Install the drivers and logs messages to the specified file.
`setup.exe /Install /Package <packagefilename>`	Installs the drivers and utilities specified in `packagefilename`.
`setup.exe /Install /Passive`	Performs an unattended installation of the drivers and utilities, rebooting the host to complete the installation as required.
`setup.exe /Install /Quiet`	Performs a **silent** installation of the drivers and utilities, rebooting the host – without prompting – to complete the installation as required.
`setup.exe /Reinstall`	Reinstalls the drivers and utilities.
`setup.exe /Uninstall`	Removes the drivers and utilities from the host operating system.
`setup.exe /Install /Verbose`	Performs a verbose installation of the drivers and utilities, outputting details for each stage of the installation procedure.
`setup.exe /Package <PackageFilename>`	Identify the package file to use for the operation.
`setup.exe /Version`	Shows version information for the drivers.
`setup.exe /Quiet /Install ADDLOCAL=NetworkAdapterManager`	Silently installs the drivers and Solarflare Adapter Manager only (other utilities will not be installed). See, Using ADDLOCAL on page 137.

Table 32: Solarflare Installation Options

Example	Action
`setup.exe /Quiet /Install ADDLOCAL=NetworkAdapterManager REBOOT=Suppress`	Silently installs the drivers and Solarflare Adapter Manager only, but suppresses the auto-restart at the end of the installation.
`<PROPERTY>=<Value>`	Specify one or more install properties.

Using ADDLOCAL

ADDLOCAL is a standard Windows Installer property that controls which features are installed via the command line. For Solarflare adapters, the following features can be installed from the command line:

- **CoreDrivers** – Installs the core adapter drivers
- **NetworkAdapterManager** – Installs Solarflare Adapter Manager (SAM)
- **CommandLineTools** – Installs Solarflare command line tools: sfboot.exe, sfupdate.exe, sfcable.exe, sfteam.exe, sfnet.exe.
- **Launcher** – Installs the Solarflare system tray icon, providing easy access to the Solarflare Adapter Manager (SAM).

Multiple features may be installed by separating each feature with a comma (spaces are not allowed).

ADDLOCAL cannot prevent **Launcher** from being installed if either **NetworkAdapterManager** or **CommandLineTools** are not installed or are still being installed.

ADDLOCAL examples

- Install the package interactively with the default installation options selected (equivalent to `Setup.exe` or `Setup.exe /Install`).

  ```
  Setup.exe /Install ADDLOCAL=CoreDrivers,
  NetworkAdapterManager,CommandLineTools,Launcher
  ```

- Install the package without any management tools. Displays a limited user interface with status and progress only.

  ```
  Setup.exe /Quiet /Install ADDLOCAL=CoreDrivers
  ```

- Install Solarflare Adapter Manager (SAM) only. This command shows no user interface during installation and will restart the host system if required.

  ```
  Setup.exe /Quiet /Install ADDLOCAL=NetworkAdapterManager
  ```


Using REBOOT

REBOOT is a standard Windows Installer property that controls when reboots occur:

- **Force** – force a reboot at the end of the installation
- **Suppress** – suppress any reboot at the end of the installation
- **ReallySuppress** – suppress any reboots during the installation, and at the end.

See https://msdn.microsoft.com/en-us/library/aa371101%28v=vs.85%29.aspx.

REBOOT example

- Install Solarflare Adapter Manager (SAM) only, showing no user interface during installation, but suppress the auto-reboot at the end of the installation.

```
Setup.exe /Quiet /Install ADDLOCAL=NetworkAdapterManager REBOOT=Suppress
```

Extract Solarflare Drivers

If it is necessary to extract the Solarflare Windows drivers, e.g. before WDS installs, this can be done from the Windows command line.

1. From the Command prompt, navigate to the directory where the installation package is located.
2. Enter the following command:

```
Setup.exe /Extract <DestinationDirectory>
```

The Destination Directory will list the following sub-directory structure - The actual folders/files displayed will depend on the Solarflare driver package installed:

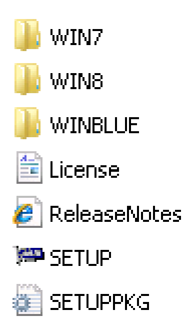


Table 33 lists the drivers supplied with the Solarflare Driver installation package:

Table 33: Solarflare Drivers

Folder	Where Used
WIN7	Driver for Windows Server 2008 R2, for use on a WDS server, or to install directly to an iSCSI target.
WIN8	Driver for Windows Server 2012, for use on a WDS server, or to install directly to an iSCSI target.

Table 33: Solarflare Drivers

Folder	Where Used
WINBLUE	Driver for Windows Server 2012 R2, for use on a WDS server, or to install directly to an iSCSI target.
SETUP	Launch the Solarflare Driver Package Setup window.
SETUPPKG	Package file listings.

4.7 Unattended Installation

This section covers the following subjects:

Windows Driver Locations

The following steps use drivers extracted from the Solarflare installation package. Refer to Table 33 for driver folder locations.

Unattended Installation using WDS

Windows Deployment Services (WDS) enables the deployment of Windows over a network (from a WDS server), avoiding the need to install each operating system directly from a CD or DVD.

- This guide assumes you have installed and are familiar with WDS. For more information on WDS, see Further Reading on page 143.
- You should also be familiar with PXE booting over Solarflare adapters. See Configuring the Boot ROM for PXE or iSCSI Booting on page 170 for more information.

The following steps are an example of how to set up an unattended installation using the WDS interface:

Add a Boot Image

1 From the left hand pane of the WDS MMC snap in, right-click the **Boot Images** node and select **Add Boot** Image.

2 Specify a name for the image group and click **Add Boot Image**.

3 Select the `boot.wim` file from the **Windows installation DVD** (in the `\Sources` folder). The `Boot.wim` file contains the **Windows PE** and the **Windows Deployment Services** client.

4 Click **Open**, then click **Next**.

5 Follow the instructions in the wizard to add the boot image.

Add an Install Image

1 From the left hand pane of the WDS MMC snap in, right-click the **Install Images** node and select **Add Install** Image.

2 Specify a name for the image group and click **Add Install Image**.

3 Select the `install.wim` file from your installation DVD (in the `\Sources` folder), or create your own install image. Consult the WDS documentation for details on creating custom install images.

4 Click **Open**, then click **Next**.

5 Follow the instructions in the wizard to add the image.

Adding Solarflare Drivers to the WDS Boot Image

These steps describe how to add the Solarflare drivers into the Boot Image.

Modifying the Boot Image

You next need to modify the boot image to include the Solarflare Drivers extracted from the setup package. Table 33 identifies drivers required for the target operating system. To modify the boot image Solarflare recommends using the **ImageX** tool supplied with the **Windows Automated Installation Kit (AIK)**.

1 Within WDS, expand the server where the boot image is located and select the boot image you want to modify. From the right-click menu, select **Disable**.

2 Create a Windows PE customization working directory (in this example `c:\windowspe-x86`). Within a command prompt, from:

```
C:\program files\windows aik\tools\petools\
```

and enter the following command:

```
copype.cmd x86 c:\windowspe-x86
```


3 Enter the following ImageX commands from the PE customization working directory:

```
imagex /info <Drive>:\remoteinstall\boot\x86\images\<boot.wim>
```

NOTE: `<Drive>` is the path where the `remoteinstall` folder is located. `<boot.wim>` is the name of your boot image.

4 Mount the boot image with the following command from your PE customization working directory:

```
imagex /mountrw <Drive>:\remoteinstall\boot\x86\images\<boot.wim> 2 mount
```

5 Copy the contents of the appropriate Solarflare driver folder (see Table 33) to a subdirectory within your PE customization working directory (in this example `c:\windowspe-x86\drivers`).

6 Add the Solarflare VBD driver to the image by entering the following command from your PE customization working directory:

```
peimg /inf=c:\windowspe-x86\drivers\netSFB*.inf mount\windows
```

7 Add the Solarflare NDIS driver to the image by entering the following command from your PE customization working directory:

```
peimg /inf=c:\windowspe-x86\drivers\netSFN6*.inf mount\windows
```

8 Unmount the image, using the following command from your PE customization working directory:

```
imagex /unmount /commit mount
```

9 From WDS, expand the server where the boot image is located and select the boot image you have modified. From the right-click menu, select **Enable**.

Create Custom Install Image

These steps describe how to add the Solarflare drivers into the Custom Install Image. These are the same Solarflare drivers added to the boot image.

Preparing the Custom Install Image

1 From WDS, locate the install image from the **Install Images** folder on your server.

2 Right-click the image and select **Export Image** from the menu.

3 Export the image to a location where it can be mounted. Solarflare recommend using the Windows PE customization working directory as this saves creating a second directory. In this example: `c:\windowspe-x86`.

Modifying the Install Image

1 Mount the install image with the following command from your PE customization working directory:

```
imagex /mountrw <Drive>:\<path>\<install.wim> 1 mount
```

NOTE: `<Drive>` is the path where the `remoteinstall` folder is located. `<boot.wim>` is the name of your boot image.

2 Copy the contents of the appropriate Solarflare driver folder in Table 33 to a sub-directory in your PE customization working directory (in this example `c:\windowspe-x86\drivers`). If you are using the same directory as for the boot image, this directory should already be present.

3 Add the Solarflare VBD driver to the image by entering the following command from your PE customization working directory:

```
peimg /inf=c:\windowspe-x86\drivers\netSFB*.inf mount\windows
```

4 Add the Solarflare NDIS driver to the image by entering the following command from your PE customization working directory:

```
peimg /inf=c:\windowspe-x86\drivers\netSFN6*.inf mount\windows
```

5 Unmount the image, using the following command from your PE customization working directory:

```
imagex /unmount /commit mount
```

Import the Custom Image to WDS

1 From WDS, select the **Image** group you want to add the image to. Right-click and select **Import Image**.

2 Browse to the location of the custom image, and click **Next**.

3 Follow the instructions in the wizard to import the image.

Create the WDSClientUnattend.xml File

The WDSClientUnattend.xml file is used by the Windows PE boot environment to configure settings including the language, credentials for connecting to the WDS server, the partitioning of the disk and which image to deploy.

NOTE: You can use the Windows System Image Manager (Part of the Windows Automated Installation Kit) to create the `WDSClientUnattend.xml` file.

To associate your WDSClientUnattend.xml file with your modified boot image:

1 Copy the `WDSClientUnattend.xml` file to the following folder in the `RemoteInstall` folder: `RemoteInstall\WDSClientUnattend`.

2 Open the Windows Deployment Services MMC snap-in, right-click the **server** that contains the Windows Server 2008 R2, 2012, or 2012 R2 boot image with which you want to associate the file, and then select **Properties**.

3 On the **Client** tab, select **Enable unattended installation**, browse to the `WDSClientUnattend.xml` file, then click **Open**.

4 Click **OK** to close the Properties page.

Create the AutoUnattend.xml File

The AutoUnattend.xml file is used during the installation of Windows Server 2008 R2, 2012, and 2012 R2 to automatically populate the various configuration settings.

NOTE: You can use the Windows System Image Manager (Part of the Windows Automated Installation Kit) to create the `AutoUnattend.xml` file.

To associate your `AutoUnattend.xml` file with your custom install image:

1 Copy the `AutoUnattend.xml` file to the following folder in the `RemoteInstall` folder: `RemoteInstall\WDSClientUnattend`.

2 Open the Windows Deployment Services MMC snap-in, select the custom install image with which you want to associate the file, right-click and then select **Properties**.

3 Select the **Allow image to install in unattend mode** option.

4 Click **Select File** and browse to your `AutoUnattend.xml` file.

Further Reading

- Installing and configuring Windows Deployment Services (WDS):
  http://technet.microsoft.com/en-us/library/cc771670%28WS.10%29.aspx
- Windows PE Customization:
  http://technet.microsoft.com/en-us/library/cc721985%28WS.10%29.aspx
- Getting Started with the Windows AIK:
  http://technet.microsoft.com/en-us/library/cc749082%28WS.10%29.aspx
- Performing Unattended Installations:
  http://technet.microsoft.com/en-us/library/cc771830%28WS.10%29.aspx
- How to add network driver to WDS boot image:
  http://support.microsoft.com/kb/923834
- Windows Deployment Services Getting Started Guide for Windows Server 2012
  http://technet.microsoft.com/en-us/library/jj648426.aspx

4.8 Managing Adapters with SAM

- Using SAM on page 145
- Using SAM to Configure Adapter Features on page 150
- Using SAM to Configure Teams and VLANs on page 157
- Using SAM to View Statistics and State Information on page 165
- Using SAM to Run Adapter and Cable Diagnostics on page 166
- Using SAM for Boot ROM Configuration on page 170

NOTE: The Windows dialog boxes displayed by SAM will appear differently on different Microsoft Windows OS versions.

Introduction

The Solarflare Adapter Manager (SAM) is a Microsoft Management Console (MMC) plug-in for managing Solarflare adapters, teams and VLANs. SAM shows information for all adapters installed on the server, alongside the standard MMC plug-in Actions pane.

Using SAM, you can easily configure Ethernet and task offloading settings, set up teams and VLANs, configure the Boot ROM for PXE or iSCSI booting, and upgrade the adapter firmware.

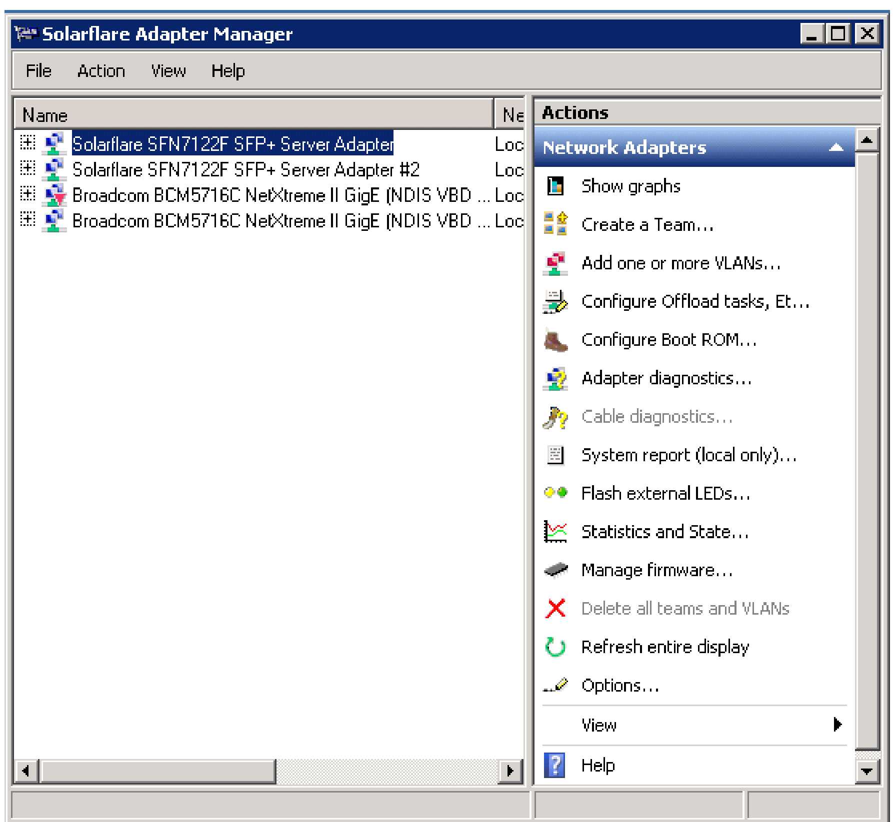


Figure 15: SAM Main Screen - Windows Server 2012

SAM's diagnostics utilities allow you to run tests on the adapter, and on 10GBASE-T adapters, on the cable to discover any potential issues which may be affecting adapter performance. Also, SAM's detailed statistics and state information can be used to view data transfer figures, sent and received packet types, as well as other traffic-related details.

SAM is included with the Solarflare drivers installation package.

4.9 Managing Adapters Remotely with SAM

SAM can be used to administer Solarflare adapters on your server from a remote computer. SAM can be used remotely to administer adapters on any supported Windows platform, including a Windows Server Core Installation. Remote Administration provides access to all SAM features, except for generating a system report.

To allow SAM to remotely administer your server, you need to add a Computer Management snap-in to the computer Microsoft Management Console (MMC).

4.10 Using SAM

Starting SAM

There are various ways of starting SAM.

To manage a local computer, do one of the following:

- If the Solarflare notification area icon is installed, right-click the icon and choose **Manage network adapters on this computer**.
- On Windows Server 2008 R2, choose **Start > All Programs > Solarflare Network Adapters > Manage network adapters on this computer**.

  On Windows Server 2012 or later, click the **Start** button followed by the arrow button, then choose **Solarflare Network Adapters > Manage network adapters on this computer**.

- On Windows Server 2008 R2, choose **Start > Administrative Tools > Computer Management > System Tools > Network Adapters**.

  On Windows Server 2012 or later, click the **Start** button, then choose **Administrative Tools > Computer Management > System Tools > Network Adapters**.

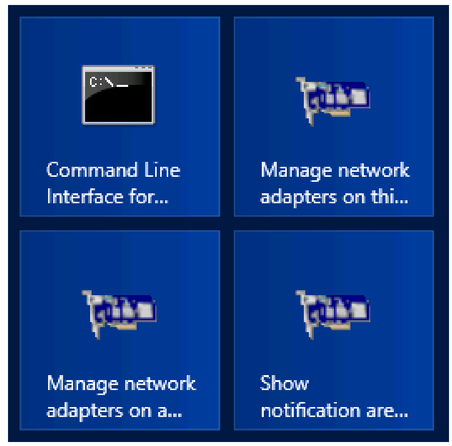


Figure 16: SAM icons on Start screen

NOTE: You may be asked for permission to continue by the User Account Control when starting SAM. You must run SAM as an administrator to make any changes.

To manage a remote computer, do one of the following:

- If the Solarflare notification area icon is installed, right-click the icon and choose **Manage network adapters on a remote computer**.
- On Windows Server 2008 R2, choose **Start > All Programs > Solarflare Network Adapters > Manage network adapters on a remote computer**.

  On Windows Server 2012 or later, click the **Start** button followed by the arrow button, then choose **Solarflare Network Adapters > Manage network adapters on a remote computer**.

Viewing Adapter Details

SAM lists all available network adapters installed in the server, regardless of manufacturer or adapter type

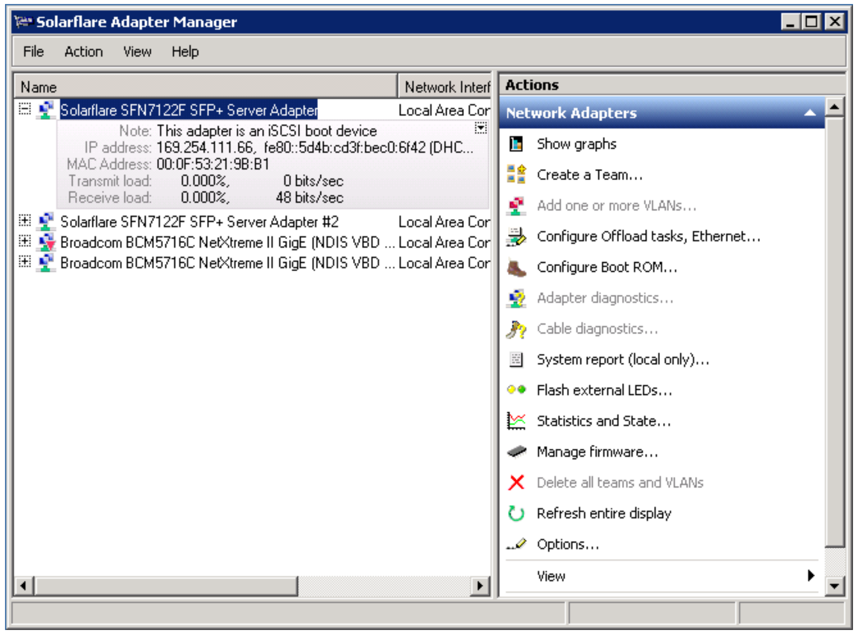


Figure 17: Solarflare Adapter Manager (SAM)

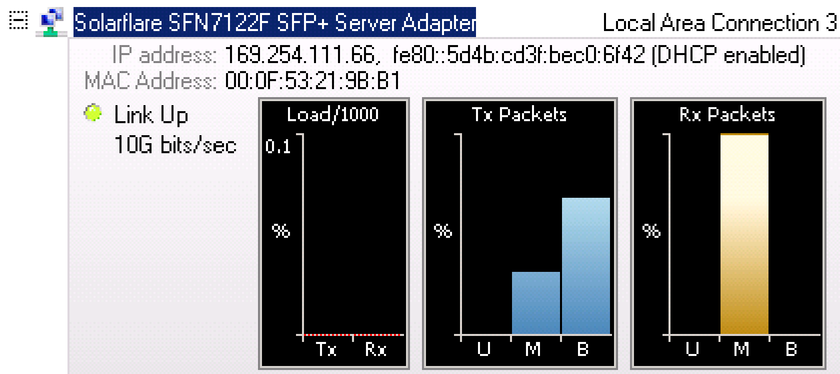
For each adapter, SAM provides the following details:

- Name and network interface
- IP address (IPv4 and IPv6, if available)
- MAC address
- Transmit load
- Receive load

For Solarflare adapters only, SAM also lists any teams or VLANs that have been configured, along with details that allow you to quickly check performance and status.

Viewing Performance Graphs

To view Solarflare performance graphs, Right-click on an adapter and select **Show graphs** from the menu. By default, SAM shows the load, transmitted packets and received packets graphs only. To view other available graphs, Select **Graphs** from the right-click menu, or from the Actions Pane/Action menu. For non-Solarflare adapters only the load graph is displayed.



Configuring Options in SAM

SAM allows you to change the units used to display data, enable separators when displaying large numbers and disable/enable warning messages.

To configure SAM options:

1. Start SAM.
2. From the **Actions** pane, click **Options**, or choose **Action > Options**.

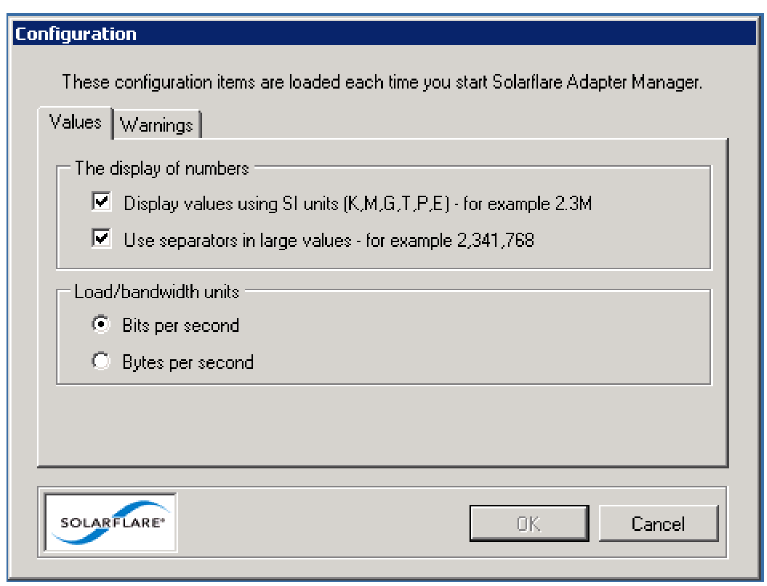


Figure 18: SAM - Actions > Options

3 In the Configuration window, select required options (seeTable 34).

4 Click **OK** to save your options or **Cancel** to retain the existing settings.

Table 34: SAM Configuration Options

Tab	Options	Description
Values	**Display values using SI units**	Displays values using metric prefixes (K, M, G, T, P, E), for example 2.3M for the value 2,300,000. Enabled by default. This can be useful when dealing with the large Tx/Rx numbers that can accumulate with 10Gb networking. Note: The Transmit and Receive bytes columns ignore this setting.
Values	**Use separators in large values**	Use separators with large numbers, for example 2,341,768. Enabled by default.
Values	**Load/bandwidth units**	Use bits per second (default setting), or bytes per second when displaying data transfer figures.
Warnings	**Warnings displayed before a major action takes place**	Warnings for the following actions can be enabled or disabled in SAM: • Deleting a VLAN or removing a network adapter from a team • Deleting a team

Working with Third-Party Adapters

Third-party adapters installed in the server are also listed in the SAM's Network Adapters list, along with the Solarflare adapters and any teams and VLANs which have been set up on the server.

SAM provides some options for working with third-party adapters. The available actions for third party adapters are shown in the **Action** pane.

4.11 Using SAM to Configure Adapter Features

SAM allows you to configure the following features on Solarflare adapters:

- Accessing Adapter Feature Settings on page 150
- Checksum Offload on page 152
- RSS and Interrupts on page 152
- Segmentation Offload
- Ethernet Link Speed on page 155
- Ethernet flow control on page 155
- Ethernet Frame Length on page 156

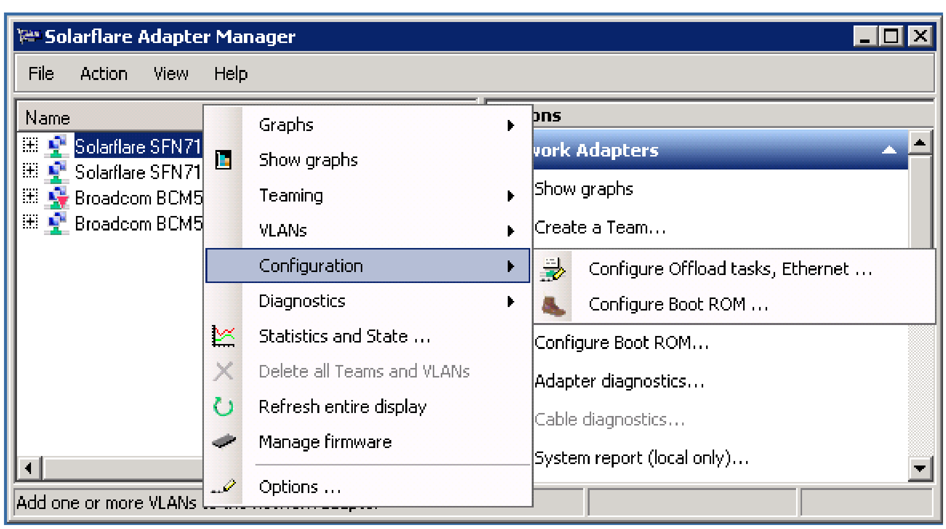
NOTE: Changing the value of an Adapter feature can negatively impact the performance of the adapter. You are strongly advised to leave them at their default values.

NOTE: Before making any changes to your Solarflare adapter features, read the Performance Tuning on Windows section on page 225 first.

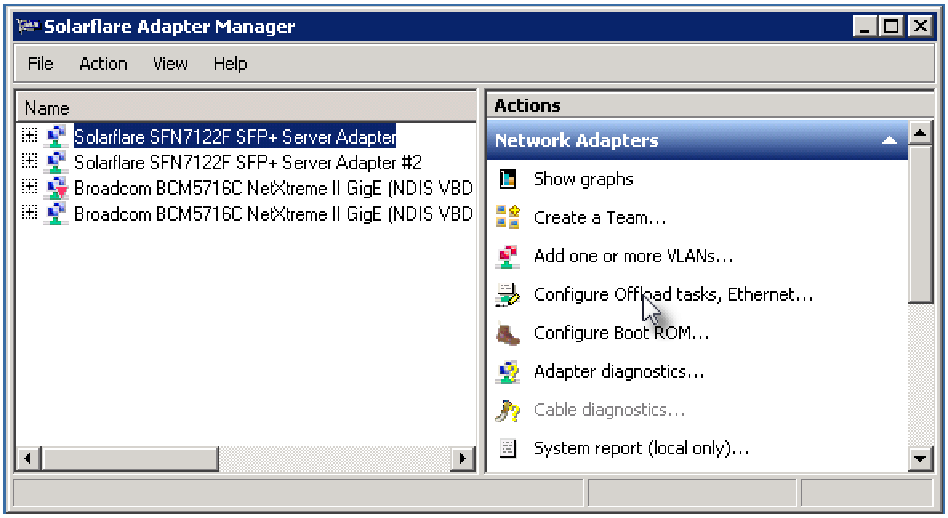
Accessing Adapter Feature Settings

Use *one* of the following methods to access the Adapter Features Dialog:

From SAM, right-click on an adapter and select **Configuration > Configure Offload tasks, Ethernet and other features**.



From SAM, select an adapter and from the **Action** menu, select **Configure Offload tasks, Ethernet and other features**.



The **Adapter Features** dialog box will be displayed:

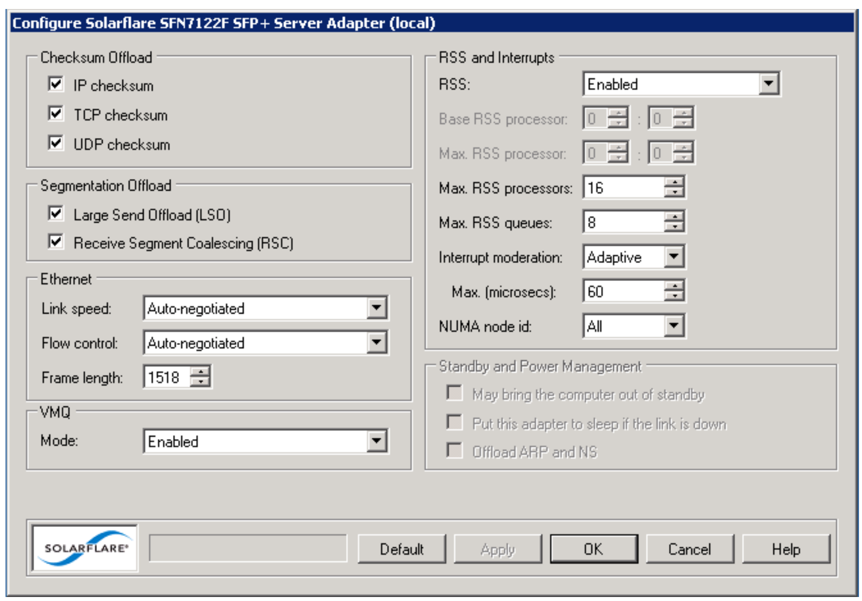


Figure 19: Solarflare Adapter Manager Adapter Features

Click **Apply** or **OK** when changes to Adapter Features are modified.

Note that the **Receive** legend in the **Segmentation Offload** field differs, depending on the version of Windows that is installed:

- for Windows Server 2008 R2, it is **Large Receive Offload (LRO)**
- for Windows Server 2012 and later, it is **Receive Segment Coalescing (RSC)**, as shown.

For more information see Segmentation Offload on page 154.

Checksum Offload

Checksum offloading is supported on IP, TCP and UDP packets. Before transmitting a packet, a checksum is generated and appended to the packet. At the receiving end, the same checksum calculation is performed against the received packet. By offloading the checksum process to the network adapter, the load is decreased on the server CPU.

By default, Solarflare adapters are set up to offload both the calculation and verification of TCP, IP and UDP checksums. The following Checksum Offload options are supported:

Table 35: Checksum Offloads

Check box selected	**Transmit and Receive**	Transmit checksums are generated and received checksums are enabled. This is the default setting.
Check selected but selection grayed out	**Transmit Only or Receive Only**	For either transmit or received checksum only. **NOTE:** The Transmit or Receive Only states can only be set from the Advanced tab of the Driver Properties. See Configuring Network Adapter Properties in Windows on page 177 for more details.
Check box cleared	**Disabled**	Disabled. Data will be checksummed by the host processor for both transmitted and received data.

You can also configure Checksum Offload settings from the network adapter properties. See Configuring Network Adapter Properties in Windows on page 177 for more details.

NOTE: Changing the Checksum Offload settings can impact the performance of the adapter. Solarflare recommend that these remain at the default values. Disabling Checksum Offload disables Large Send Offload.

RSS and Interrupts

Solarflare network adapters support RSS (Receive Side Scaling) and interrupt moderation. Both are enabled by default and can significantly improve the performance of the host CPU when handling large amounts of network data.

RSS attempts to dynamically distribute data processing across the available host CPUs in order to spread the workload. Interrupt moderation is a technique used to reduce the number of interrupts sent to the CPU. With interrupt moderation, the adapter will not generate interrupts closer together than the interrupt moderation interval. An initial packet will generate an interrupt immediately, but if subsequent packets arrive before the interrupt moderation interval, interrupts are delayed.

You can also configure RSS and interrupts settings from the network adapter properties. See Configuring Network Adapter Properties in Windows on page 177 for more details.

NOTE: Changing the RSS and Interrupt Moderation settings can impact the performance of the adapter. You are strongly advised to leave them at their default values.

RSS and Interrupts Options

Table 36 shows the RSS and interrupts options.

Table 36: RSS and Interrupts Options

Displayed (supported) options will differ between Windows OS versions and different Solarflare drivers.	
RSS	Disabled - RSS is disabled. Closest Processor - use cores from a single NUMA node. Closest Processor Static - Network traffic is distributed across available CPUs from a single NUMA node, but there is no dynamic load balancing. NUMA Scaling - CPUs are assigned on a round-robin basis across every NUMA node. NUMA Scaling Static - As for NUMA Scaling but without dynamic load balancing. Conservative Scaling - RSS will use as few processors as possible to sustain the current network load. This helps to reduce the number of interrupts.
Max. RSS processors	Set the number of processors to be used by RSS. If this is greater than or equal to the number of logical processors in the system then all processors are used.
Interrupt moderation	Adaptive - adjusts the interrupt rates dynamically, depending on the traffic type and network usage. Disabled - interrupt moderation is disabled. Enabled - interrupt moderation is enabled.
Max (microseconds)	This setting controls the value for the interrupt moderation time. The default value is 60 microseconds and can be changed for deployments requiring minimal latency.
Base RSS processor	The base processor to be used by RSS. The value is specified as a group (range 0-9) and CPU number (range 0-63).

Table 36: RSS and Interrupts Options

Max. RSS processor	The maximum processor available to RSS. The value is specified as a group (range 0-9) and CPU number (range 0-63).
Max. RSS processors	The maximum number of processors to be used by RSS. The value is in the range 0-256.
Max. RSS queues	The maximum number of receive queues created per interface. The value is in the range 0-64.
NUMA node id	The NUMA node id drop down list box is displayed on Windows platforms that support NUMA architectures. This constrains the set of CPU cores used for RSS to the specified NUMA node. Solarflare recommend you leave this at the default setting of **All. The adapter will attempt to use only processors from the specified NUMA node for RSS. If this is set to ALL or it is greater than or equal to the number of NUMA nodes in the system, all NUMA nodes are used.**

Further Reading

For more information on Windows RSS profiles and options refer to http://msdn.microsoft.com/en-us/library/windows/hardware/ff570864%28v=vs.85%29.aspx

4.12 Segmentation Offload

Solarflare adapters offload the tasks of packet segmentation and reassembly to the adapter hardware, reducing the CPU processing burden and improving performance.

- Large Send Offload (LSO), when enabled, offloads to the adapter the splitting of outgoing TCP data into packets. This reduces CPU use and improves peak throughput. Since LSO has no effect on latency, it can be enabled at all times. The driver has LSO enabled by default.
- Receive Segment Coalescing (RSC) is a Microsoft feature introduced in Windows Server 2012. When enabled the adapter will coalesce multiple received TCP packets on a TCP connection into a single call to the TCP/IP stack. This reduces CPU use and improves peak performance. RSC has a low impact on latency. If a host is forwarding received packets from one interface to another then Windows will automatically disable RSC. RSC is enabled by default.
- Large Receive Offload (LRO) is a Solarflare proprietary mechanism similar to RSC. It is used when RSC is unavailable (i.e. on Windows Server 2008 R2). When enabled the adapter will coalesce multiple received TCP packets on a TCP connection into a single call to the TCP/IP stack. This reduces CPU use and

improves peak performance. However LRO can increase latency and should not be used if a host is forwarding received packets from one interface to another. LRO is disabled by default.

You can also configure LSO and RSC/LRO settings from the NDIS properties. See Configuring Network Adapter Properties in Windows on page 177 for more details.

Ethernet Link Speed

Generally, it is neither necessary or desirable to configure the link speed of the adapter. The adapter by default will negotiate the link speed dynamically, connecting at the maximum, supported speed. However, if the adapter is unable to connect to the link partner, you may wish to try setting a fixed link speed. For further information see 'Link Speed' in Table 46 on page 178.

Ethernet flow control

Ethernet flow control allows two communicating devices to inform each other when they are being overloaded by received data. This prevents one device from overwhelming the other device with network packets. For instance, when a switch is unable to keep up with forwarding packets between ports. Solarflare adapters allow flow control settings to be auto-negotiated with the link partner.

You can also configure ethernet flow control from the network adapter properties. See Table 46 on page 178 for more details.

Table 37: Ethernet Flow Control Options

Option	Description
Auto-negotiate	Flow control is auto-negotiated between the devices. This is the default setting, preferring **Generate and respond** if the link partner is capable.
Generate and respond	Adapter generates and responds to flow control messages.
Respond only	Adapter responds to flow control messages but is unable to generate messages if it becomes overwhelmed.
Generate only	Adapter generates flow control messages but is unable to respond to incoming messages and will keep sending data to the link partner.
None	Ethernet flow control is disabled on the adapter. Data will continue to flow even if the adapter or link partner is overwhelmed.

Ethernet Frame Length

The maximum Ethernet frame length used by the adapter to transmit data is (or should be) closely related to the MTU (maximum transmission unit) of your network. The network MTU determines the maximum frame size that your network is able to transmit across all devices in the network.

NOTE: For optimum performance set the Ethernet frame length to your network MTU.

If the network uses Jumbo frames, SAM supports frames up to a maximum of 9216 bytes.

Virtual Machine Queue

Solarflare adapters support VMQ to offload the classification and delivery of network traffic destined for *Hyper-V* virtual machines to the network adapter thereby reducing the CPU load on Hyper-V hosts.

Windows Server 2008 R2 allows the administrator user to statically configure the number of CPUs available to process interrupts for VMQ. Interrupts are spread across the specified cores, however the static configuration does not provide best performance when the network load varies over time.

Dynamic VMQ, supported in Windows Server 2012 and later, will dynamically distribute received network traffic across available CPUs while adjusting for network load by, if necessary, bringing in more processors or releasing processors under light load conditions.

VMQ supports the following features:

- Classification of received network traffic in hardware by using the destination MAC address (and optionally also the VLAN identifier) to route packets to different receive queues dedicated to each virtual machine.
- Can use the network adapter to directly transfer received network traffic to a virtual machine's shared memory avoiding a potential software-based copy from the Hyper-V host to the virtual machine.
- Scaling to multiple processors by processing network traffic destined for different virtual machines on different processors.

Table 38: VMQ Mode Options

Enabled	VMQ uses the destination MAC address and also the VLAN identifier for filtering traffic to the intended Hyper-V virtual machine. This is the default.

Table 38: VMQ Mode Options

Enabled (without VLAN filtering)	VMQ uses only the destination MAC address for filtering traffic to the intended Hyper-V virtual machine.
Disabled	VMQ is disabled.

4.13 Using SAM to Configure Teams and VLANs

About Teaming

NOTE: To set up teams and VLANS in Windows using the sfteam command line tool, see Sfteam: Adapter Teaming and VLAN Tool on page 200.

Solarflare adapters support the following teaming configurations:

- IEEE 802.3ad Dynamic link aggregation
- Static link aggregation
- Fault tolerant teams

Teaming allows the user to configure teams consisting of all Solarflare adapter ports on all installed Solarflare adapters or might consist of only selected adapter ports e.g. from a dual port Solarflare adapter, the first port could be a member of team A and the second port a member of team B or both ports members of the same team.

NOTE: Adapter teaming and VLANs are not supported in Windows for iSCSI remote boot enabled Solarflare adapters. To configure load balancing and failover support on iSCSI remote boot enabled adapters, use Microsoft MultiPath I/O (MPIO), which is supported on all Solarflare adapters.

This section is only relevant to teams of Solarflare adapters. Solarflare adapters can be used in multi-vendor teams when teamed using the other vendor's teaming driver.

CAUTION: Windows Server 2012 introduced native support for teaming. Windows teaming and Solarflare teaming configuration should not be mixed in the same server.

Setting Up Teams

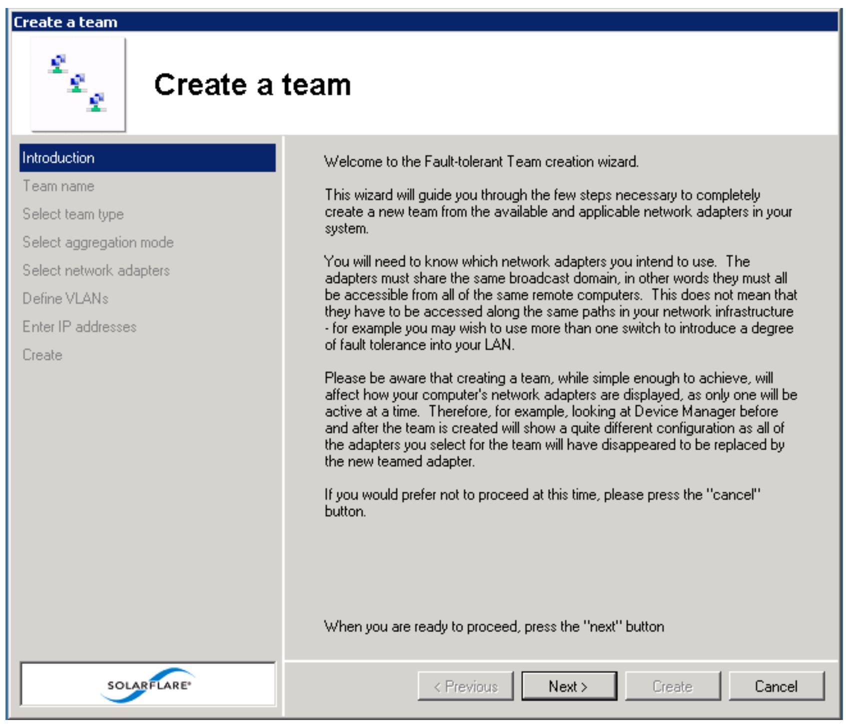
SAM's **Create a Team** setup wizard will guide you through setting up an adapter team, automatically assigning the active adapter, key adapter and standby adapter.

To create a team:

1 Before creating a team, Solarflare strongly recommend taking the server offline to avoid disrupting existing services as the team is being configured.

2 Start SAM and select a Solarflare adapter in the Network Adapter list.

3 From the **Action** menu, select **Create a Team**. The Solarflare Create a team Wizard starts.



4 Team Create Wizard

5 The wizard will guide you through the process of creating a team and optionally adding VLANs to your team (see Table 40 on page 164 for help when selecting VLAN options).

6 Bring the server back online.

7 After creating a team, you can use the **Configure this Team** option from the **Actions** pane to change team settings, such as the Ethernet frame length, key adapter assignment, and adapter priorities within the team.

CAUTION: Before physically removing an adapter from a server, first check it is not the key adapter. You **must** reassign the key adapter if you want to remove it from the team to avoid duplicating the MAC address on your network. See Table 39 on page 160 for details on reassigning the key adapter.

Reconfiguring a Team

When setting up teams, SAM assigns the key, active and standby adapters, and specifies the Ethernet frame length for the team. To change any of these settings, use the **Configure this Team** option, as described below.

To change team settings:

NOTE: Changing team settings can disrupt network traffic flow to and from services running on the server. Solarflare recommend only changing network settings when disruption to the services can be tolerated.

1. Start SAM and, from the Network Adapter list, select the team you want to reconfigure.
2. From the **Action** menu, select **Configure this Team**. The **Configure a Team** dialog box displays.

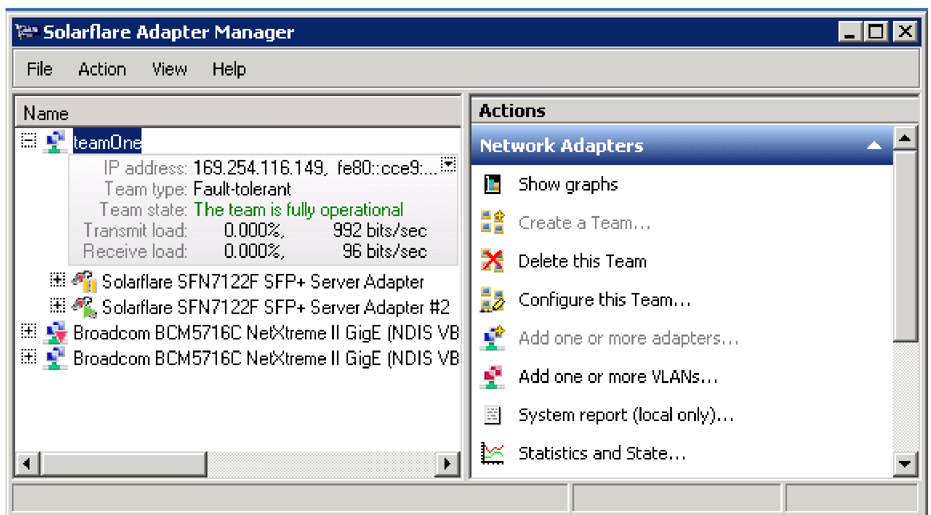


Figure 20: Configure a Team

By default, all teamed adapters are given an equal priority (indicated by the grouped number 1). The current active adapter is indicated by the green active symbol. The key adapter is indicated with the key symbol. Adapters in standby are indicated by the yellow standby symbol. For link aggregated teams there may be more than one active adapter.

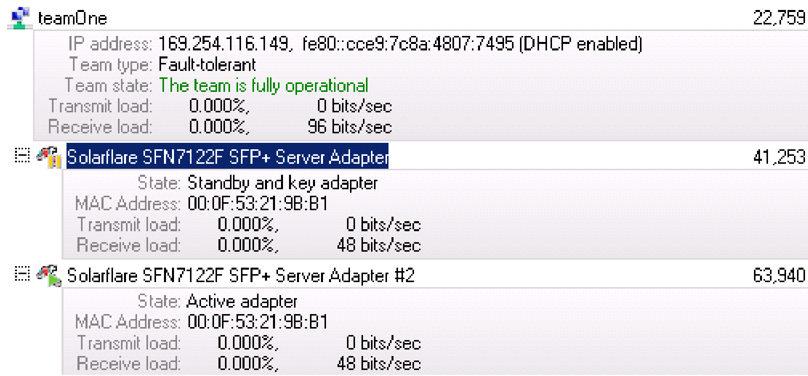


Figure 21: Prioritized Adapters

Figure 21 shows the active adapter with the highest priority, with the second adapter being second priority.

Table 39: Configure a Team Options

To change the key adapter:	Select the new key adapter, then click the **key** button. **Note**: Before physically removing an adapter from a server, first check it is not the key adapter. You **must** reassign the key adapter if you want to remove it from the team to avoid duplicating the MAC address on your network.
To change adapter priority:	By default, all adapters have equal priority. Select an adapter and use the up or down buttons to promote or demote the adapter priority as required. **Note**: For Fault-Tolerant Teams, the highest priority adapter in a team becomes the active adapter, passing all network traffic for the team.

Table 39: Configure a Team Options

To specify a new active adapter:	For Fault -Tolerant Teams only. Set your preferred active adapter to the highest prioritized adapter in the team. The highest prioritized adapter becomes the active adapter in the team after you apply your changes. To change adapter priority, use the up and down buttons.
To specify the Ethernet frame length/MTU:	Specify a value between 1514 and 9216 bytes. Check your network supports the new frame length before setting the new value. **Note**: This setting affects all adapters in the team, and will override any individual adapter settings made from the **Configure Offload tasks, Ethernet and other features** window. See Using SAM to Configure Adapter Features on page 150 for more details.

3 After making your changes, click **Set** and then click **Close**.

Adding Adapters to a Team

If additional Solarflare adapters are installed in your server, you can add them to an existing team to increase the overall resilience or performance (aggregation) of the server connection.

To add adapters to a team:

NOTE: Changing team settings can disrupt current services running on the server. Solarflare recommend only changing network settings when disruption to the services can be tolerated.

1 Start SAM and select a Solarflare adapter team from the Network Adapter list.

2 From the **Actions** list, click **Add one or more adapters**, or choose **Actions > Add one or more adapters**. The Available Network Adapters dialog box is displayed:

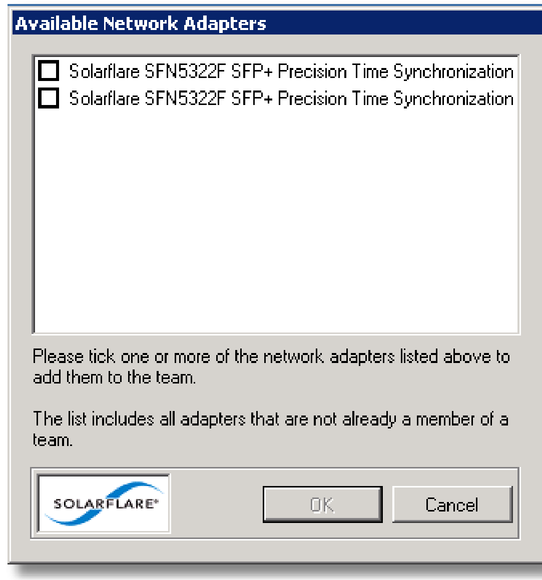


Figure 22: Available Adapters

3 Select the adapter(s) to add to the team. Click **OK** to add the selected adapters and close the dialog box.

Deleting Teams

You can delete a team by selecting **Delete this team** in SAM. Once a team has been deleted, all of its adapters are returned to their original configuration settings and become available on the server once again. Any VLANs set up for the team will be deleted when the team is deleted.

To delete a team:

NOTE: Changing team settings can disrupt current services running on the server. Solarflare recommend only changing network settings when disruption to network services can be tolerated.

1 Start SAM and select a Solarflare adapter team from the Network Adapter list.

2 From the **Action** menu, select **Delete this team**. Alternatively, to delete all teams and VLANs on the server, select **Delete all teams and VLANs**. The Confirm Action Dialog box is displayed.

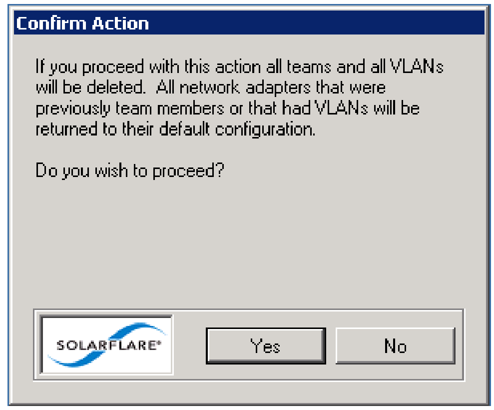


Figure 23: Confirm Action

3 Confirm the deletion when prompted.

> **NOTE: Delete all teams and VLANs** will cause a display refresh which may take some time to complete, depending on the number of teams and VLANs being deleted.

Setting up Virtual LANs (VLANs)

SAM allows you to add up to 64 VLANs per team or adapter. Each VLAN is a virtual network adapter, visible in the Windows Device Manager, through which the operating system is able to receive data tagged with the correct VLAN ID (VID). You may assign one VLAN to accept VLAN 0 or untagged traffic, which allows the interface to communicate with devices that do not support VLAN tagging, or that are sending traffic on VLAN 0.

To create VLANs:

> **NOTE:** Creating VLANs can disrupt current services running on the server. Solarflare recommend only changing network settings when disruption to network services can be tolerated.

1 Start SAM and select the adapter or adapter team from the Network Adapter list.

2 From the **Actions** list, click **Add one or more VLANs**, or choose **Actions > Add one or more VLANs** to display the VLAN Setup Wizard.

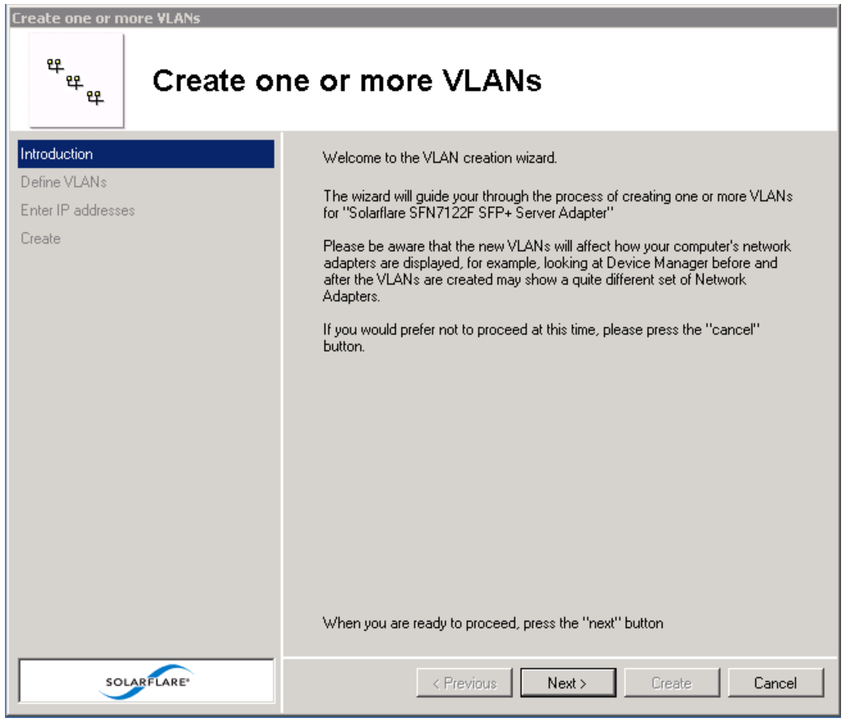


Figure 24: Create VLANs

Table 40: VLAN Options

Option	Description
Name	An optional name for the VLAN network adapter. This option will not be available when remotely administering the server.
Supports the handling of priority traffic	Enables the handling of traffic that is tagged as priority.
Supports untagged and VLAN 0 traffic	Restricts the VLAN to handling packets that are untagged or with VID 0. This option allows the interface to communicate with devices which don't support VLAN tagging.
Supports traffic solely on this VLAN	Restricts the network interface to traffic that is tagged with the specified VLAN.

Deleting VLANs

VLANs can be removed from a team or single adapter when no longer required.

To delete VLANs:

NOTE: Deleting VLANs can disrupt current processes and applications running on the server. Solarflare recommend only changing network settings when disruption to network services can be tolerated.

1. Start SAM.
2. In the Network adapter list, select the VLAN to delete. If necessary, expand the team if the VLAN is attached to a team then select the VLAN.
3. From the Actions list, click **Delete this VLAN**, or choose **Action > Delete this VLAN**.
4. Confirm the deletion in the Confirm Action Dialog box.

4.14 Using SAM to View Statistics and State Information

SAM's Network Adapter list provides an overview of the adapters installed in the host computer. For a more detailed view of the adapter device settings, data transfer statistics, and other features, you can use the adapter Statistics and State.

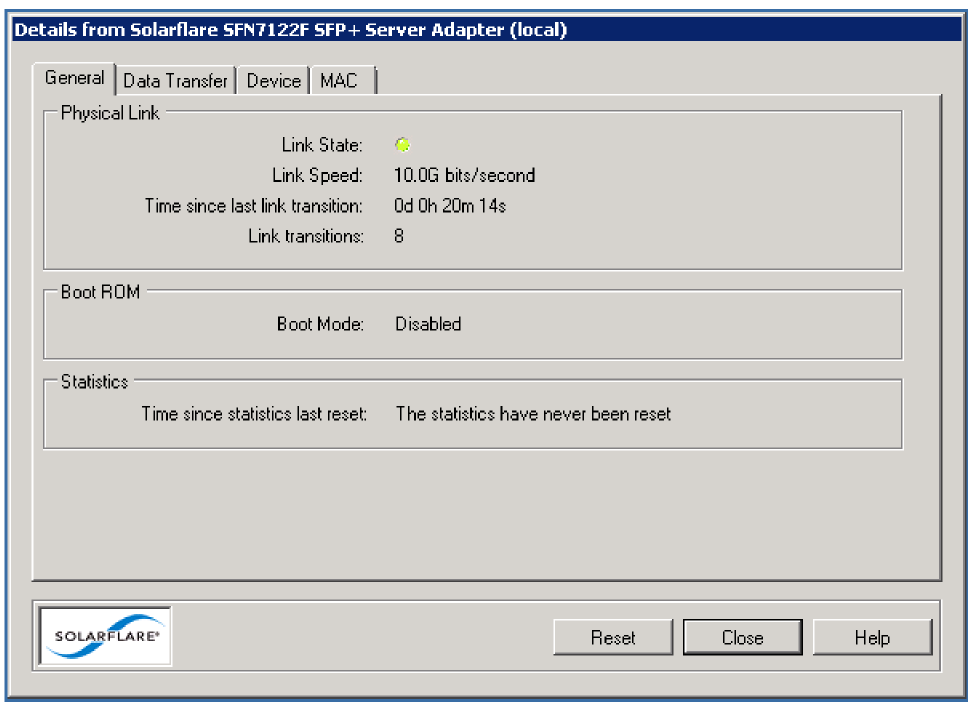


Figure 25: Solarflare Adapter Statistics and State

To view Solarflare statistics and state information:

1. Start SAM and select a Solarflare adapter from the Network Adapter list.
2. From the Actions list, click **Statistics and State**. The **Details from <adapter name>** dialog box is displayed.

> **NOTE:** The tabs displayed will differ, dependent on whether an adapter, VLAN or Team is selected.

3. Click each tab to see the various adapter statistics and state information that is available for the adapter. Note that statistics are collated from the start of the current session. To reset the statistics, see Resetting Adapter Statistics on page 166.
4. When you have finished viewing statistics, click **Close**.

Resetting Adapter Statistics

Statistics for data transfer and the MAC layer are reset, either following a system restart or installing of the adapter drivers. If necessary, you can reset the adapter statistics to restart the accumulated data values at any time.

1. Start SAM and select a Solarflare adapter from the Network Adapter list.
2. From the Actions list, click **Statistics and State**, or choose **Actions > Statistics and State**. The **Details from <adapter name>** dialog box is displayed.
3. In the **General** tab, click the **Reset** button to reset statistics.
4. Click **Close**.

4.15 Using SAM to Run Adapter and Cable Diagnostics

You can verify the Solarflare adapter, driver and cable by running SAM's built-in diagnostic tools (Solarflare 10GBASE-T adapter only).

The tools provide a simple way to verify that the adapter and driver are working correctly, and that the cable has the correct characteristics for high-speed data transfer.

The diagnostics tools also include an option to flash the LEDs (useful for identifying the adapter in a server room), and an option to generate a full system report, both available from the **Actions** menu.

> **NOTE:** Running of these tests will cause traffic to be halted on the selected adapter, and all of its VLANs, unless part of a fault-tolerant team. Diagnostics tests are not available when the adapter is running in iSCSI boot mode.

> **NOTE:** The full system report cannot be generated when remotely administering a server.

Running Driver and Adapter Diagnostics

SAM's driver diagnostics enable you to test the adapter and driver are functioning correctly, returning a simple pass or fail for each test run.

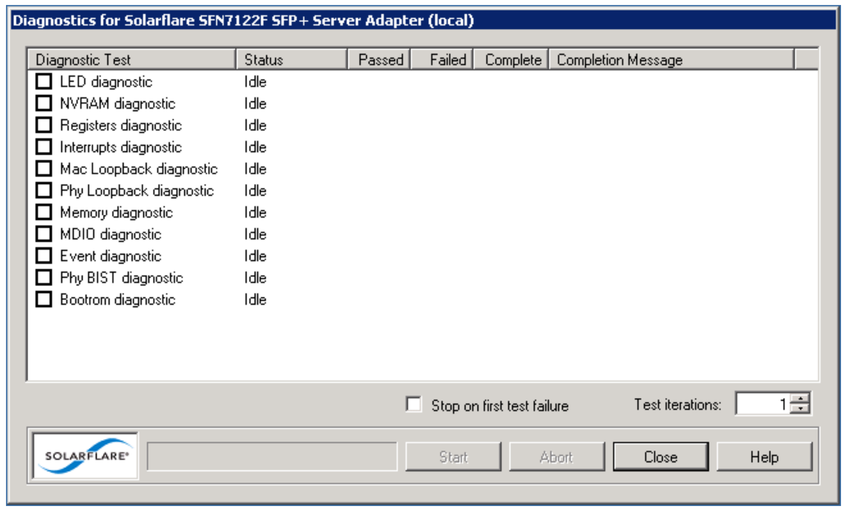


Figure 26: Adapter and Driver Diagnostics Window

1 Start SAM and select a Solarflare adapter from the Network Adapter list.

2 From the Action menu, select **Adapter Diagnostics**. The **Diagnostics for <adapter name>** window is displayed.

3 Select the test you want to run (no tests are selected by default). See Table 41 for a description of the tests that are available.

4 To stop as soon as a failure is detected, select **Stop on first test failure**.

5 To run all the tests more than once, change the value in the **Test iterations** box.

6 Click **Start** to begin testing. The results of each test will be displayed in the Diagnostics window, along with an entry in the Completion Message column describing the reason any particular test has failed.

NOTE: The adapter will stop functioning while the tests are being run. Solarflare recommend only running diagnostics tests when disruption to network services can be tolerated.

NOTE: You can click **Abort** to abandon running tests at any time. This may take a while to complete, dependent on the test being run at the time.

The available tests depend on the installed adapter type.

Table 41: Adapter Diagnostic Tests

Diagnostic Test	Purpose
LED	Flashes the LEDs for 5 seconds.
NVRAM	Verifies the flash memory **board configuration** area by parsing and examining checksums.
Registers	Verifies the adapter registers by attempting to modify the writable bits in a selection of registers.
Interrupts	Examines the available hardware interrupts by requesting the controller to generate an interrupt and verifying that the interrupt has been processed by the network driver.
MAC loopback	Verifies that the network driver is able to pass packets to and from the network adapter using the MAC loopback layer.
PHY loopback	Verifies that the network driver is able to pass packets to and from the network adapter using the PHY loopback layer.
Memory	Verifies SRAM memory by writing various data patterns (incrementing bytes, all bit on and off, alternating bits on and off) to each memory location, reading back the data and comparing it to the written value.
MDIO	Verifies the MII registers by reading from PHY ID registers.
Event	Verifies the adapter's event handling capabilities by posting a software event on each event queue created by the driver and checking it is delivered correctly. The driver creates an event queue for each CPU.
PHY BIST	Examines the PHY by initializing it and starting any available built-in self tests to run.
Bootrom	Verifies the Boot ROM configuration and image checksum. Will warn if no Boot ROM is present.

Running Cable Diagnostics

With high-speed data networking, the suitability of the cable in achieving maximum transfer rates is especially important. SAM's cable diagnostic tool can be used to verify the attached cable, reporting its condition, measured length and electrical characteristics for each cable pairing.

NOTE: Cable diagnostics are only available on Solarflare 10GBASE-T Adapters. For these adapters, Solarflare recommend using good quality Category 6, 6a or 7 cable up to the maximum length as determined by the cable category.

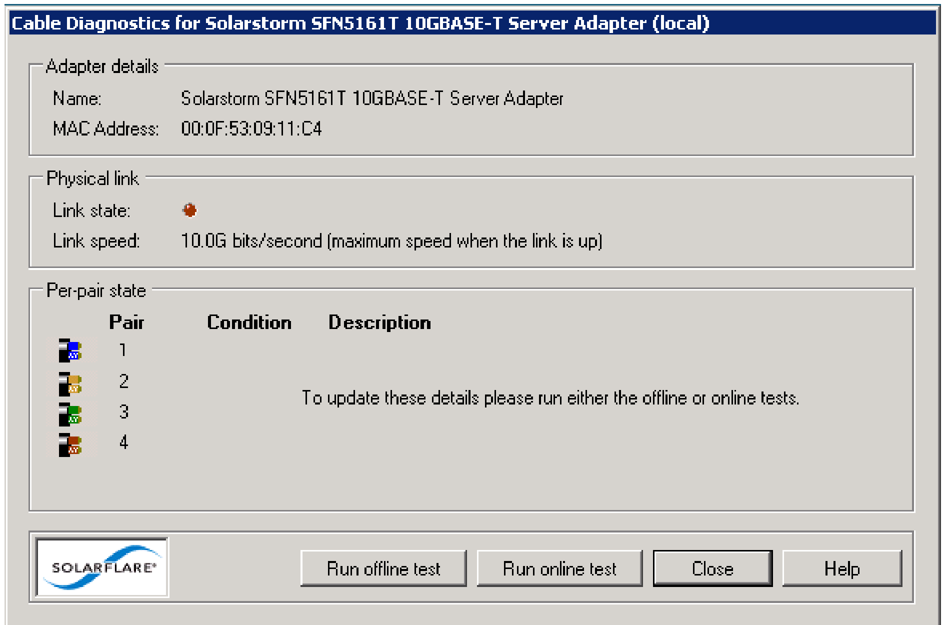


Figure 27: Cable Diagnostics Window

1. Start SAM and select a Solarflare adapter from the Network Adapter list.
2. From the **Action** menu, click **Cable Diagnostics**. The **Cable Diagnostics for <adapter name>** dialog box is displayed.
3. Click **Run offline test** or **Run online test**. Offline testing produce more detailed results, but at the expense of disrupting the connection while tests are running.

CAUTION: The offline tests will cause the network link to momentarily drop and disrupt data flow. Solarflare recommend only running diagnostics tests when disruption to your services can be tolerated.

4. The results of the testing will be displayed in the diagnostics dialog box. For analysis of the cable pair results, see Table 42.

Table 42: Cable Pair Diagnostic Results

Result	Meaning
OK	Cable is operating correctly.
Length measured = ..., SNR margin = ...	The range is ±13dB (approximately). The SNR should be positive.

Table 42: Cable Pair Diagnostic Results

Result	Meaning
Error **Pair short at ...**	A short circuit has been detected at the indicated length. The cable or the connector is faulty and must be replaced.
Error **Pair is open circuit**	An open circuit has been detected. The cable or the connector is faulty and must be replaced.

4.16 Using SAM for Boot ROM Configuration

For booting of diskless systems, Solarflare adapters support Preboot Execution Environment (PXE) and iSCSI booting.

When booting the server directly from an iSCSI target, you will first need to enable iSCSI booting and configure the iSCSI initiator, target and user authentication to match your network and target settings, or rely on DHCP to configure the settings dynamically when the adapter initializes (this is the default setting for all iSCSI options).

Using SAM, you can access the adapter Boot ROM to configure your firmware settings for adapter booting, as described below.

Configuring the Boot ROM for PXE or iSCSI Booting

For more information on configuring the iSCSI target and DHCP settings from the Solarflare Boot Configuration Utility, and how to install an operating system that is enabled for remote iSCSI booting over a Solarflare adapter, See Solarflare Boot ROM Agent on page 435.

To configure PXE or iSCSI booting on the Solarflare Boot ROM:

1 Start SAM and select a Solarflare adapter from the Network Adapter list.

From the **Action** menu, select the **Configure Boot ROM** option. The **Configure Boot ROM** window displays with the **General** tab selected.

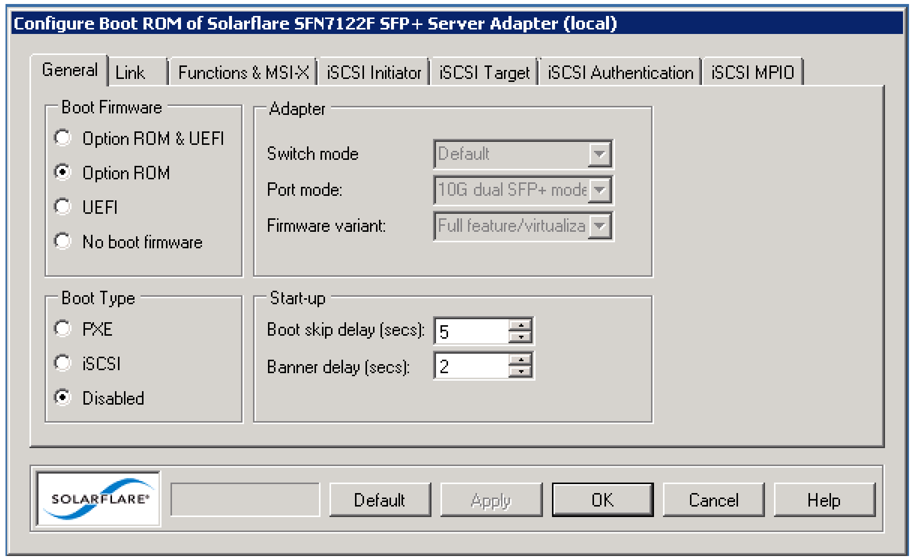


Figure 28: BootROM Configuration

2 From the **Boot Type** panel, select either PXE or iSCSI booting as required. You can also configure the types of Boot Firmware, the maximum number of MSI-X Interrupts supported and start-up configuration used by the Boot ROM utility. For more details on these options see Sfboot: Boot ROM Configuration Tool on page 183.

NOTE: iSCSI booting will not be available if the adapter is a member of a team or has VLANs.

NOTE: Solarflare recommend not changing the MSI-X Interrupts setting.

3 If necessary, from the **Link** tab, change the **Link Speed** option depending on your link requirement. Note that **Auto-negotiated** is correct for most links and should not be changed unless advised. The Link Speed options will vary depending on the installed adapter.

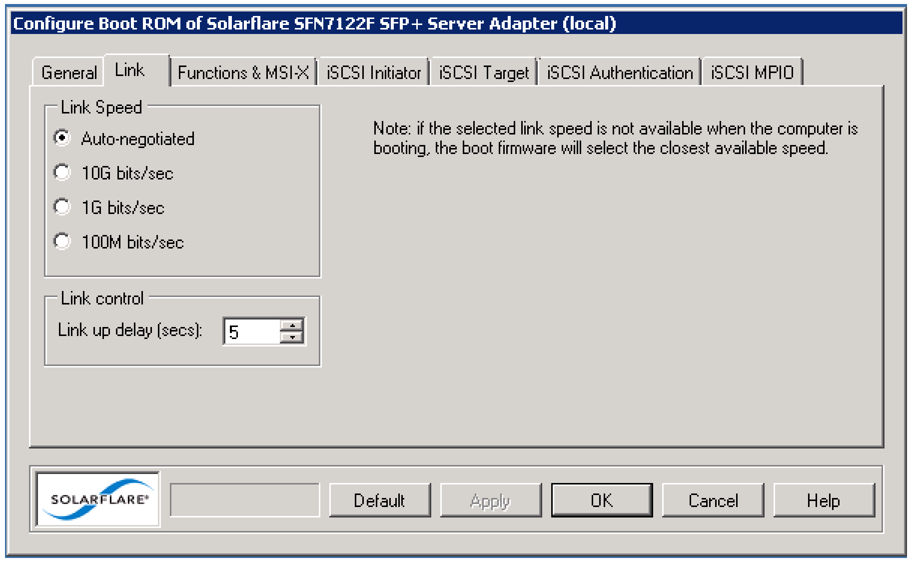


Figure 29: Link tab

4 The Link up delay specifies a wait time before the boot device will attempt to make a connection. This allows time for the network to start following power-up. The default setting is 5 seconds, but can be set from 0–255 seconds. This can be used to wait for spanning tree protocol on a connected switch to unblock the switch port after the physical network link is established.

5 If you selected **PXE** as the boot type, click **OK** to finish the setup procedure.

If you selected **iSCSI booting** as the boot type, click the **iSCSI Initiator** tab and continue with the following steps.

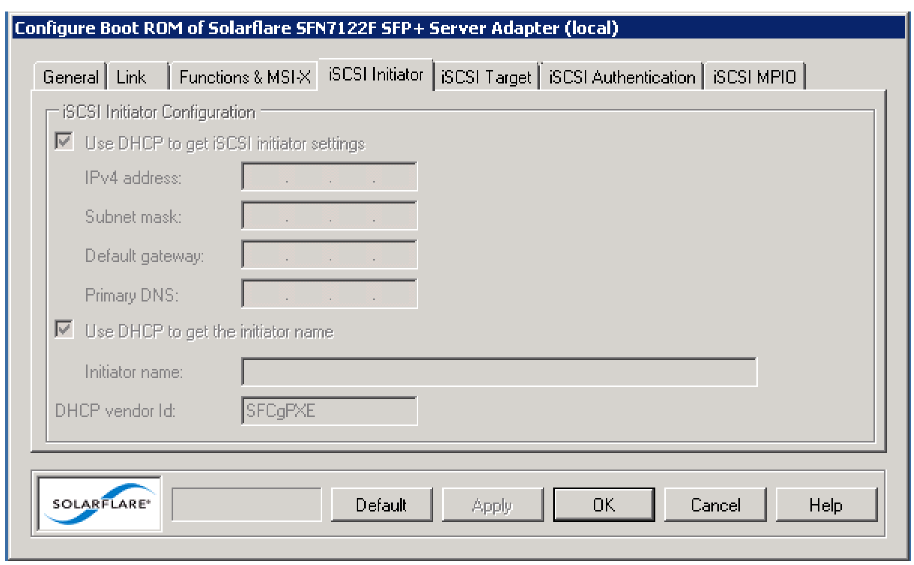


Figure 30: iSCSI Initiator tab

6 If using DHCP to configure the adapter's network settings at boot time, ensure **Use DHCP to get iSCSI Initiator settings** is selected. Otherwise, clear this option and enter network details for the adapter, as described in Table 43.

Table 43: iSCSI Initiator Options

Option	Description
IPv4 Address	An IPv4 address to assign to the adapter. Ensure this address is unique.
Subnet mask	Subnet mask. For example 255.255.255.0
Default Gateway	IPv4 address of your network router.
Primary DNS	IPv4 address of your Primary DNS server.

7 If you are not using DHCP to get the initiator name, clear **Use DHCP to get the initiator name** and enter a iSCSI Qualified Name (IQN) in the Initiator name field.

8 **DHCP vendor Id** specifies the device vendor ID to be advertised to the DHCP server. This setting is always enabled and not affected by any of the other DHCP options. See DHCP Server Setup on page 445 for more details on this and other DHCP options.

9 Click the **iSCSI Target** tab.

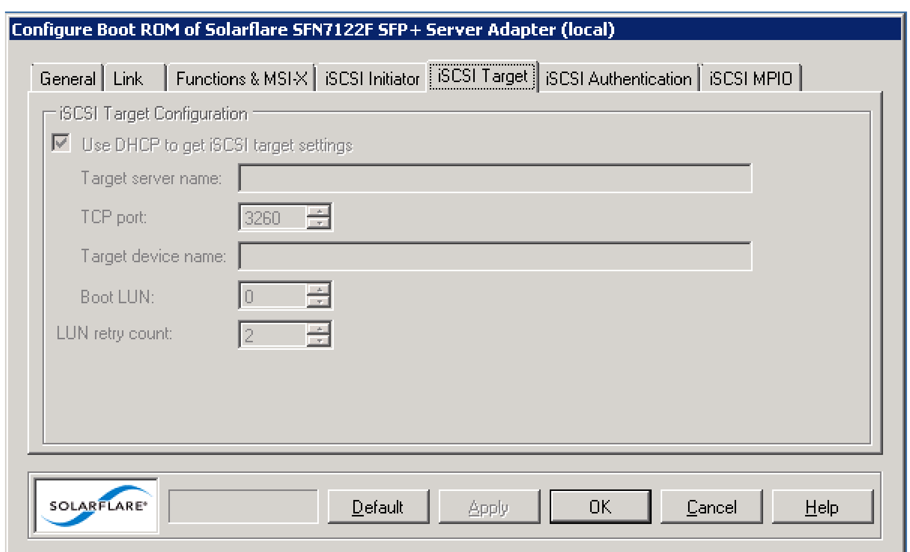


Figure 31: iSCSI Target tab

10 If using DHCP to discover the iSCSI target details, ensure **Use DHCP to get iSCSI target settings** is selected. Otherwise, clear the option and enter details for the iSCSI target, as described in Table 44.

Table 44: iSCSI Target Options

Option	Description
Target server name	Target server network address in the form of a dotted quad (i.e. 10.1.2.3) IPv4 address or fully qualified domain name (FQDN), such as mytarget.myorg.mycompany.com
TCP port	ISCSI port number that has been configured on the target. Default is 3260.
Target device name	The iSCSI Qualified Name (IQN) of the target server, which will look something like: `iqn:2009-01.com.solarflare`.
Boot LUN	Logical unit number which has been set up on the server. The system will attempt to attach to this LUN on boot up and attempt to load the target operating system from it.
LUN retry count	Specifies the number of times the boot device will attempt to connect to the target LUN (logical unit number) before failing. The default setting is 2 retries, but can be set from 0–255. This setting is enabled, even if using DHCP is being used.

11 Click the **iSCSI Authentication** tab.

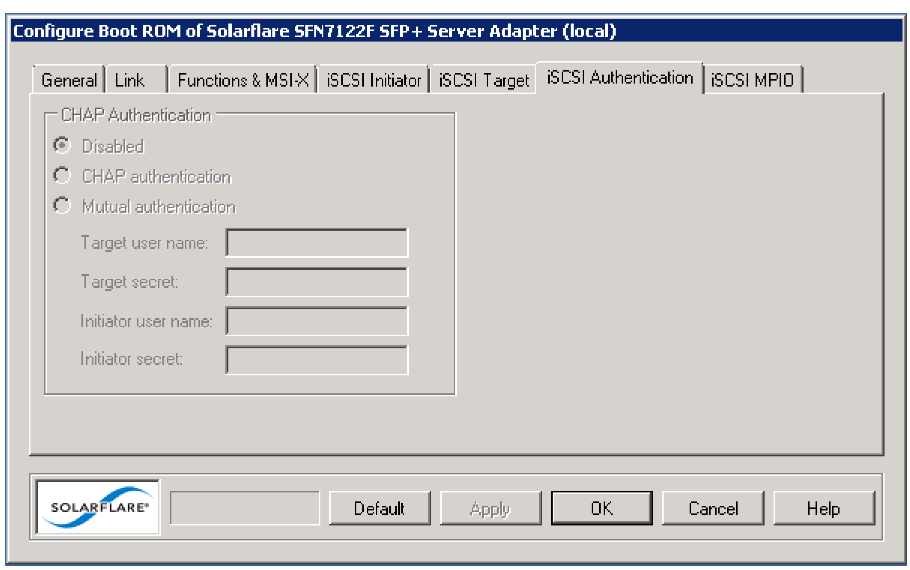


Figure 32: iSCSI Authentication tab

12 By default Challenge Handshake Application Protocol (CHAP) authentication is disabled. You have the following options:

- CHAP authentication - this is target initiated or **one way** authentication
- Mutual authentication - both the target and the initiator will authenticate the connection.

If CHAP authentication is configured on the iSCSI target, enter the correct settings to allow access to the target.

Table 45: CHAP Options

Option	Description
Target user name	Name of the target server, as set on the iSCSI target CHAP settings.
Target secret	Target password.
Initiator user name	Name of this initiator (as set on the target). A minimum of 9 characters. Used for Mutual authentication only
Initiator secret	Password of this initiator (as set on the target). A minimum of 12 characters. Used for Mutual authentication only.

13 Select the **iSCSI MPIO** tab

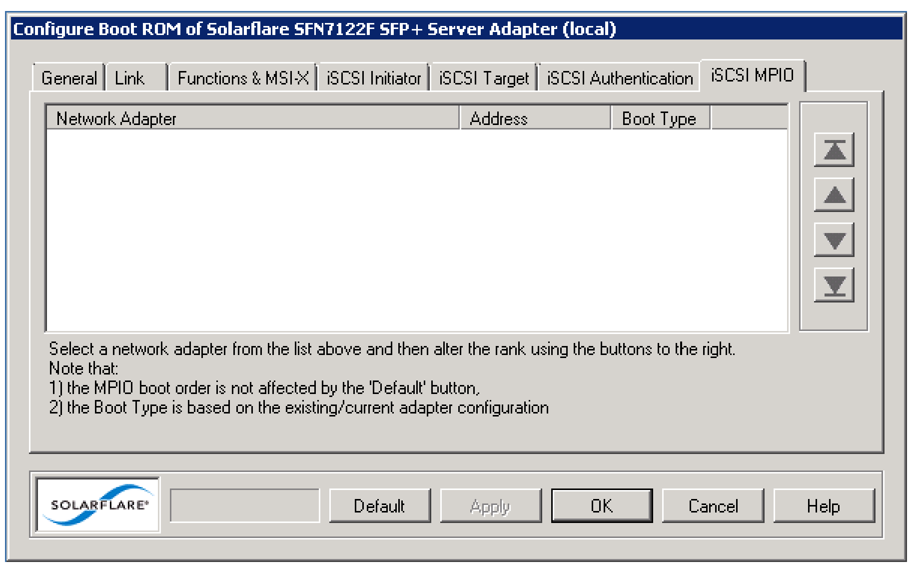


Figure 33: iSCSI MPIO tab

For iSCSI booting in multi-adapter environments, you can set the priority of each adapter. By default, all iSCSI enabled adapters are given an equal priority. The setting is used to determine how traffic is re-routed in case of one adapter entering a failed state.

14 When you have finished configuring the iSCSI settings, click **OK** or **Apply** to save your settings to the Boot ROM.

Disabling Adapter Booting

You can stop the adapter from attempting to initiate either a PXE or iSCSI boot after a restart.

1 Start SAM and select the Solarflare adapter from the Network Adapter list.

2 From the Action menu, click the **Configure Boot ROM** option. The Configure Boot ROM dialog box displays with the **BIOS** tab selected.

3 From the **Boot Type** panel, select **Disabled**.

4 Click **OK** or **Apply** to save your settings to the Boot ROM.

4.17 Managing Firmware with SAM

SAM allows you to monitor the firmware (PHY, Boot ROM and Adapter) for your Solarflare adapters. Either select **Manage firmware** from the **Actions** pane, or from the **Action** menu. The firmware update window is displayed:

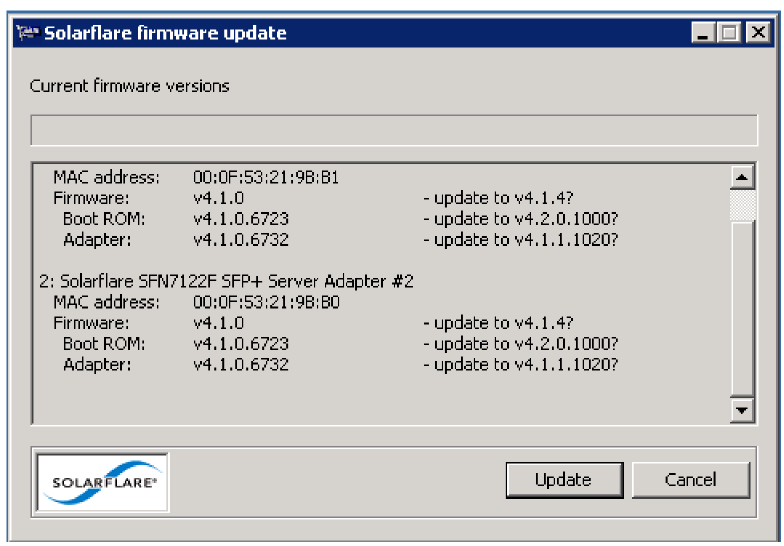


Figure 34: Solarflare firmware update window

If the firmware is up to date, the window will contain the **OK** button. If the firmware is out of date, the OK button is replaced with an **Update** and **Cancel** button. To update the firmware, click **Update**.

You can also use the sfupdate command line tool to manage the firmware on your Solarflare adapters. See Sfupdate: Firmware Update Tool on page 197 for more details.

4.18 Configuring Network Adapter Properties in Windows

Network adapter properties for the Solarflare adapter are available through the Windows Device Manager entry for the relevant network adapter. You can also access the adapter properties using SAM.

NOTE: If SAM is open, any changes made in the adapter properties will not be reflected in SAM until you close the Advanced Properties page.

To configure network adapter properties:

1. From the **Control Panel**, select **System**.
2. Select **Device Manager** from the left hand menu.
3. Expand the Network adapters.
4. Right-click the on the Solarflare adapter, and then click **Properties** to display the properties dialog box.

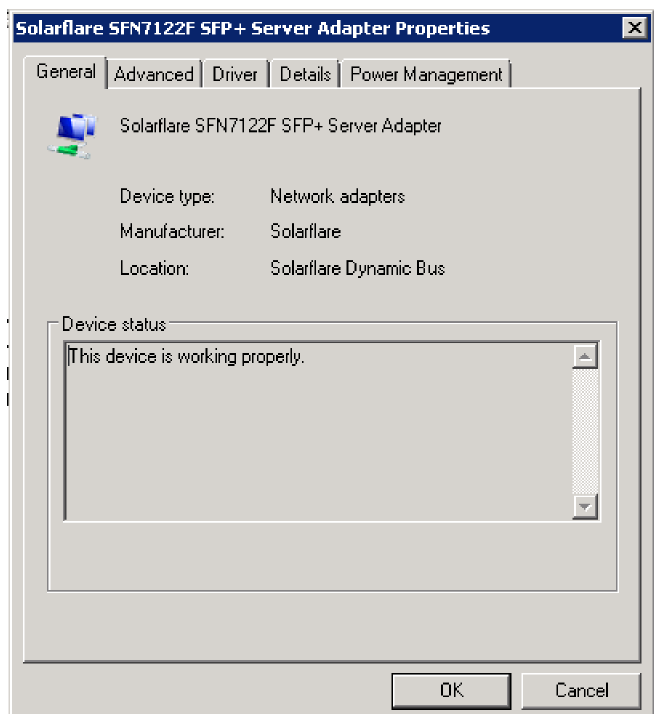


Figure 35: Adapter Properties Dialog

5. Click the **Advanced** tab to view and edit the NDIS properties. See Table 46 for a list of the available properties.

NOTE: Changing these properties may impact the performance of your Solarflare adapter. You are strongly advised to leave them at their default values.

> **NOTE:** Before making any changes to your Solarflare adapter features, read the Performance Tuning on Windows section on 225 first.

Table 46: Solarflare Network Adapter Properties

Property Name	Values	Description
Adaptive Interrupt Moderation	Enabled Disabled	This setting is dependent on the **Interrupt Moderation** setting. If **Interrupt Moderation** is enabled, **Adaptive Interrupt Moderation** allows the adapter to vary its interrupt moderation automatically, according to network traffic demands. If **Adaptive Interrupt Moderation** is disabled, interrupt moderation interval is fixed at the setting specified in Interrupt Moderation Time. Default setting: **Enabled**
Flow Control	Auto Negotiation Disabled Rx & Tx Enabled Rx Enabled Tx Enabled	Ethernet flow control (802.3x) is a way for a network device to signal to a sending device that it is overloaded, such as when a device is receiving data faster than it can process it. The adapter does this by generating a 'pause frame' to request the sending device to temporarily stop transmitting data. Conversely, the adapter can respond to pause frames by suspending data transmission, allowing time for the receiving device to process its data. Default setting: **Auto Negotiation.**
Interrupt Moderation	Enabled Disabled	Interrupt moderation is a technique used to reduce the number of interrupts sent to the CPU. With interrupt moderation, the adapter will not generate interrupts closer together than the interrupt moderation time. An initial packet will generate an interrupt immediately, but if subsequent packets arrive before the interrupt moderation time period, interrupts are delayed. Default setting: **Enabled**
Interrupt Moderation Time	1–1000 us	Specifies the interrupt moderation period when **Interrupt Moderation** is enabled. The default setting (60µs) has been arrived at by lengthy and detailed system analysis, balancing the needs of the operating system against the performance of the network adapter. Default setting: **60µs**

Table 46: Solarflare Network Adapter Properties

Property Name	Values	Description
IPv4 Checksum Offload	Disabled Rx & Tx Enabled Rx Enabled Tx Enabled	IP checksum offload is a hardware offload technology for reducing the load on a CPU by processing IP checksums in the adapter hardware. Offload IP Checksum is enabled by default for transmitted and received data. Default setting: **Rx & Tx Enabled.**
Large Receive Offload (IPv4 and IPv6)	Enabled Disabled	Large Receive Offload (LRO) is an offload technology for reducing the load on a CPU by processing TCP segmentation for received packets in the adapter. This is available only on Windows Server 2008 R2. Default setting: **Disabled**
Large Send Offload Version 2 (IPv4 and IPv6)	Enabled Disabled	Large Send Offload (LSO) is an offload technology for reducing the load on a CPU by processing TCP segmentation for transmitted packets in the adapter. **Caution**: Disabling LSO may reduce the performance of the Solarflare adapter. Default setting: **Enabled**
Locally Administered Address	Value: (MAC address) Not Present	Assigns the specified MAC address to the adapter, overriding the permanent MAC address assigned by the adapter's manufacturer. Addresses are entered as a block of six groups of two hexadecimal digits separated by hyphens (-), for example: `12-34-56-78-9A-BC` **Note**: To be a valid address, the second most significant digit must be a 2, 6, A or E, as in the above example. Check the System Event Log for any configuration issues after setting this value. Default setting: **Not Present**.

Table 46: Solarflare Network Adapter Properties

Property Name	Values	Description
Max Frame Size	1514–9216	Specifies the maximum Ethernet frame size supported by the adapter. **Note**: Devices will drop frames if they are unable to support the specified frame size, so ensure the value you set here is supported by other devices on the network. Default settings: Solarflare adapter: **1514** bytes Teamed adapter: **1518** bytes **Note**: The setting must be a multiple of 2.
Maximum number of RSS Processors	1-256	Maximum number of processors that can be used by RSS. Default value is **16**.
Maximum number of RSS Queues	1-64	Specify the number of RSS receive queues are created by the adapter driver. Default is **8**.
Preferred Numa Node	All 0 to 15	The adapter attempts to use only the CPUs from the specified NUMA node for RSS. If this is set to **All** or is greater than or equal to the number of NUMA nodes in the system all NUMA nodes are used. Default setting: **All**
Receive Segment Coalescing	Enabled Disabled	Receive Segment Coalescing (RSC) is an offload technology for reducing the load on a CPU by processing TCP segmentation for received packets in the adapter. This is available on Windows Server 2012 and later. Default setting: **Enabled**
Receive Side Scaling (RSS)	Enabled Disabled	Receive Side Scaling (RSS) is a technology that enables packet receive processing to scale with the number of available processors (CPUs), distributing the processing workload across the available resources. Default setting: **Enabled**

Table 46: Solarflare Network Adapter Properties

Property Name	Values	Description
Speed & Duplex	100 Mbps Full Duplex 1.0 Gbps Full Duplex 10 Gbps Full Duplex 40 Gbps Full Duplex Auto Negotiation	Configure the adapter speed. Default is **Auto Negotiation**.
TCP Checksum Offload (IPv4 and IPv6)	Disabled Rx & Tx Enabled Rx Enabled Tx Enabled	TCP checksum offload is a hardware offload technology for reducing the load on a CPU by processing TCP checksums in the adapter hardware. Default setting: **Rx & Tx Enabled.**
UDP Checksum Offload (IPv4 and IPv6)	Disabled Rx & Tx Enabled Rx Enabled Tx Enabled	UDP checksum offload is a hardware offload technology for reducing the load on a CPU by processing UDP checksums in the adapter hardware. Default setting: **Rx & Tx Enabled.**
Virtual Machine Queues	Enabled Disabled	VMQ offloads classification and delivery of network traffic destined for Hyper-V virtual machines to the network adapter, reducing CPU utilization on Hyper-V hosts. Default setting: **Enabled.**
VMQ VLAN Filtering	Enabled Disabled	VLAN filtering allows the adapter to use the VLAN identifier for filtering traffic intended for Hyper-V virtual machines. When disabled only the destination MAC address is used for filtering. Default setting: **Enabled.**

4.19 Windows Command Line Tools

The command line tools (see Table 47) provide an alternative method of managing Solarflare network adapters to SAM. They are especially useful on a Windows Server Core installation, where SAM cannot be run locally. As with SAM, you can run the command line tools remotely. The tools can also be scripted.

The command line tools are installed as part of the drivers installation on Windows. See Installing the Solarflare Driver Package on Windows on page 130.

Table 47: List Available Command Line Utilities

Utility	Description
`sfboot.exe`	A tool for configuring Solarflare adapter Boot ROM options, including PXE and iSCSI booting. See Sfboot: Boot ROM Configuration Tool on page 183.
`sfupdate.exe`	A tool for updating adapter Boot ROM and PHY firmware. See Sfupdate: Firmware Update Tool on page 197.
`sfteam.exe`	A tool for managing fault-tolerant adapter teams and VLANs. See Sfteam: Adapter Teaming and VLAN Tool on page 200.
`sfcable.exe`	A tool for that runs cable diagnostics for Solarflare 10GBASE-T server adapters. See Sfcable: Cable Diagnostics Tool on page 205.
`sfnet.exe`	Allows you to display and/or set the offload, Ethernet, RSS, interrupt moderation and VMQ features of any one adapter, VLAN or Team. See Sfnet on page 208.

To start a command line tool, open a Command Line Interface windows and enter the command tool.exe:

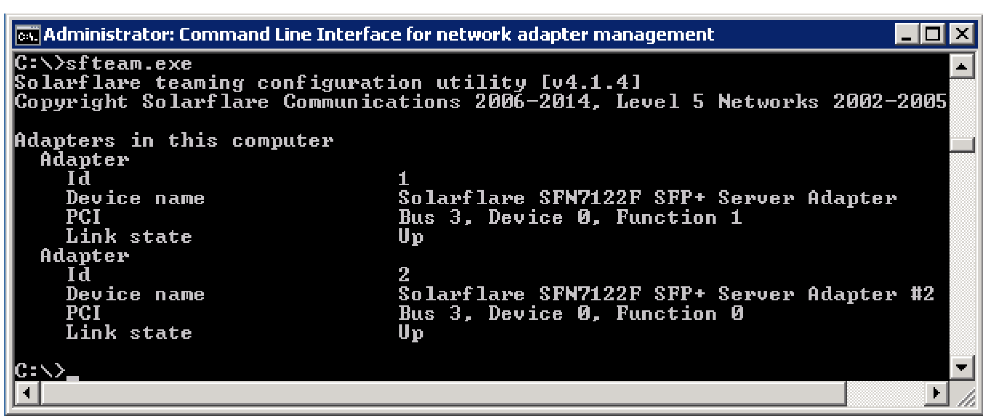


Figure 36: Windows console to run Solarflare command line tools.

NOTE: For all the utilities, the options are documented with the forward slash (/) prefix. You can also use a single dash (-) or a double dash (--) as a prefix.

NOTE: Utilities must be run as an administrator to make any changes. When run as a non administrator, an error message will be displayed.

4.20 Sfboot: Boot ROM Configuration Tool

Sfboot is a Windows command line utility for configuring Solarflare adapter Boot ROM options, including PXE and iSCSI booting. Using sfboot is an alternative to using **Ctrl+B** to access the Boot Rom agent during server startup.

See Configuring the Solarflare Boot ROM Agent on page 436 for more information on the Boot Rom agent.

Sfboot: Command Usage

1. Login with an administrator account.
2. Click **Start > All Programs > Solarflare Network Adapters > Command Line Interface for Network Adapters**.
3. From the Command Prompt, enter the command using the following syntax:

   ```
   sfboot [/Adapter <Identifier>] [options] [parameters]
   ```

   where:

   - `Identifier` is the name or ID of the adapter that you want to manage. Specifying the adapter is optional - if it is not included the command is applied to all Solarflare adapters in the machine.
   - `option` is the option you want to apply. See Sfboot: Command Line Options for a list of available options.

   If using sfboot in a configuration script, you can include the environment variable `%SFTOOLS%` to set the path to the Solarflare tools. For example:

   ```
   SET PATH=%PATH%;%SFTOOLS%
   ```


Sfboot: Command Line Options

Table 48 lists the options for `sfboot.exe`, Table 49 lists the available global parameters, and Table 50 lists the available per-adapter parameters. Note that command line options are case insensitive and may be abbreviated.

NOTE: Abbreviations in scripts should be avoided, since future updates to the application may render abbreviated scripts invalid.

Table 48: Sfboot Options

Option	Description
`/Help`	Displays command line syntax and provides a description of each sfboot option.
`/Version`	Shows detailed version information and exits.
`/Nologo`	Hide the version and copyright message at startup.
`/Verbose`	Shows extended output information for the command entered.
`/Quiet` Aliases: `/Silent`	Suppresses all output, including warnings and errors; no user interaction. You should query the completion code to determine the outcome of commands when operating silently.
`/Log <Filename>`	Logs output to the specified file in the current folder or an existing folder. Specify `/Silent` to suppress simultaneous output to screen, if required.
`/Computer <ComputerName>`	Performs the operation on a specified remote computer. Administrator rights on the remote computer is required.
`/List`	Lists all available Solarflare adapters. This option shows the adapter’s ID number, ifname and MAC address. Note: this option may not be used in conjunction with any other option. If this option is used with configuration parameters, those parameters will be silently ignored.
`/Adapter <Identifier>`	Performs the action on the identified Solarflare network adapter. The adapter identifier can be the adapter ID number, ifname or MAC address, as output by the `/List` option. If `/Adapter` is not included, the action will apply to all installed Solarflare adapters.
`/Clear`	Resets all options to their default values. If an adapter is specified, options for the given adapter are reset, but global options (shown in Table 49) are not reset. Note that `/Clear` can also be used with parameters, allowing you to reset to default values, and then apply the parameters specified.

The following global parameters in Table 49 are used to control the configurable parameters for the Boot ROM driver when running prior to the operating system booting.

Table 49: Sfboot Global Parameters

Parameter	Description
`boot-image=` `all|optionrom|uefi|disabled`	Specifies which boot firmware images are served-up to the BIOS during start-up. This parameter can not be used if the `--adapter` option has been specified.

`port-mode=`
`default|1x10G|2x10G|4x10G|`
`2x40G`
	Configure the port mode to use. This is for SFN7000 series adapters only. The values specify the connectors available after using any splitter cables. The usable values are adapter-dependent:
	• SFN7*xx*2F: 1x10G, 2x10G (default)
	• SFN7*xx*4F: 2x10G, 4x10G (default)
	• SFN7*xx*2Q: 2x10G, 4x10G, 2x40G (default)
	Changes to this setting with `sfboot` require a cold reboot to become effective. MAC address assignments may change after altering this setting.

`firmware-variant=`
`full-feature|ultra-low-latency|`
`capture-packed-stream|auto`
	Configure the firmware variant to use. This is for SFN7000 series adapters only:
	• the SFN7002F adapter is factory set to full-feature
	• all other adapters are factory set to auto.
	Default value = auto - means the driver will select a variant that meets its needs:
	• the VMware driver always uses full-feature
	• otherwise, ultra-low-latency is used.
	The ultra-low-latency variant produces best latency without support for TX VLAN insertion or RX VLAN stripping (not currently used features).

`insecure-filters=` `enabled|disabled`	If enabled bypass filter security on non-privileged functions. This is for SFN7000 series adapters only. This reduces security in virtualized environments. The default is disabled. When enabled a function (PF or VF) can insert filters not qualified by their own permanent MAC address. This is a requirement when using bonded interfaces.

Table 49: Sfboot Global Parameters

Parameter	Description
`mac-spoofing=enabled|disabled`	If enabled, non-privileged functions can create unicast filters for MAC addresses that are not associated with them. This is for SFN7000 series adapters only. The default is disabled. Changes to this setting with `sfboot` require a cold reboot to become effective.
`rx-dc-size=8|16|32|64`	Specifies the size of the descriptor cache for each receive queue. This is for SFN7000 series adapters only. The default is: • 16 if the `port-mode` supports the maximum number of connectors for the adaptor • 32 if the `port-mode` supports a reduced number of connectors.
`tx-dc-size=8|16|32|64`	Specifies the size of the descriptor cache for each transmit queue. This is for SFN7000 series adapters only. The default is: • 32 if the `port-mode` supports the maximum number of connectors for the adaptor • 64 if the `port-mode` supports a reduced number of connectors.
`vi-count=<vi count>`	Sets the total number of virtual interfaces that will be available on the NIC.

The following per-adapter parameters in Table 50 are used to control the configurable parameters for the Boot ROM driver when running prior to the operating system booting.

Table 50: Sfboot Per-adapter Parameters

Parameter	Description
`link-speed=auto|10g|1g|100m`	Specifies the network link speed of the adapter used by the Boot ROM. The default is `auto`. On the 10GBASE-T adapters, `auto` instructs the adapter to negotiate the highest speed supported in common with its link partner. On SFP+ adapters, `auto` instructs the adapter to use the highest link speed supported by the inserted SFP+ module. On 10GBASE-T and SFP+ adapters, any other value specified will fix the link at that speed, regardless of the capabilities of the link partner, which may result in an inability to establish the link. `auto` Auto-negotiate link speed (default) `10G` 10G bit/sec `1G` 1G bit/sec `100M` 100M bit/sec
`linkup-delay=` `<delay time in seconds>`	Specifies the delay (in seconds) the adapter defers its first connection attempt after booting, allowing time for the network to come up following a power failure or other restart. This can be used to wait for spanning tree protocol on a connected switch to unblock the switch port after the physical network link is established. The default is 5 seconds.
`banner-delay=` `<delay time in seconds>`	Specifies the wait period for `Ctrl-B` to be pressed to enter adapter configuration tool. `<delay time in seconds>` = 0-256
`bootskip-delay=` `<delay time in seconds>`	Specifies the time allowed for `Esc` to be pressed to skip adapter booting. `<delay time in seconds>` = 0-256
`boot-type=pxe|iscsi|disabled`	Sets the adapter boot type – effective on next boot. `pxe` – PXE (Preboot eXecution Environment) booting `iscsi` – iSCSI (Internet Small Computer System Interface) booting `disabled` – Disable adapter booting

Table 50: Sfboot Per-adapter Parameters

Parameter	Description
initiator-dhcp=enabled|disabled	Enables or disables DHCP address discovery for the adapter by the Boot ROM except for the Initiator IQN (see initiator-iqn-dhcp). This option is only valid if iSCSI booting is enabled (boot-type=iscsi). If initiator-DHCP is set to disabled, the following options will need to be specified: initiator-ip=<IPv4 address> netmask=<IPv4 subnet mask> The following options may also be needed: gateway=<ip_address> primary-dns=<ip_address>
initiator-ip=<IPv4 address>	Specifies the IPv4 address (in standard "." notation form) to be used by the adapter when initiator-dhcp is disabled. Note that this option is only valid if iSCSI booting is enabled (boot-type=iscsi). Example: sfboot boot-type=iscsi initiator-dhcp=disabled initiator-ip=<192.168.1.3>
netmask=<IPv4 subnet mask>	Specifies the IPv4 subnet mask (in standard "." notation form) to be used by the adapter when initiator-dhcp is disabled. Note that this option is only valid if iSCSI booting is enabled (boot-type=iscsi). Example: sfboot boot-type=iscsi initiator-dhcp=disabled netmask=255.255.255.0
gateway=<IPv4 address>	Specifies the IPv4 subnet mask (in standard "." notation form) to be used by the adapter when initiator-dhcp is disabled. Note that this option is only valid if iSCSI booting is enabled (boot-type=iscsi). Example: sfboot boot-type=iscsi initiator-dhcp=disabled gateway=192.168.0.10

Table 50: Sfboot Per-adapter Parameters

Parameter	Description
`primary-dns=<IPv4 address>`	Specifies the IPv4 address (in standard "." notation form) of the Primary DNS to be used by the adapter when `initiator-dhcp` is disabled. This option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot boot-type=iscsi initiator-dhcp=disabled primary-dns=192.168.0.3`
`initiator-iqn-dhcp= enabled|disabled`	Enables or disables use of DHCP for the initiator IQN only.
`initiator-iqn=<IQN>`	Specifies the IQN (iSCSI Qualified Name) to be used by the adapter when `initiator-iqn-dhcp` is disabled. The IQN is a symbolic name in the "." notation form; for example: iqn.2009.01.com.solarflare, and is a maximum of 223 characters long. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot initiator-iqn-dhcp=disabled initiator-iqn=iqn.2009.01.com.solarflare adapter=2`
`lun-retry-count=<retry count>`	Specifies the number of times the adapter attempts to access and login to the Logical Unit Number (LUN) on the iSCSI Target before failing. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot lun-retry-count=3`

`target-dhcp=enabled|disabled`	Enables or disables the use of DHCP to discover iSCSI target parameters on the adapter.
	If `target-dhcp` is disabled, you must specify the following options:
	`target-server=<DNS name or IPv4 address>`
	`target-iqn=<IQN>`
	`target-port=<port number>`
	`target-lun=<LUN>`
	Example - Enable the use of DHCP to configure iSCSI Target settings:
	`sfboot boot-type=iscsi target-dhcp=enabled`

Table 50: Sfboot Per-adapter Parameters

Parameter	Description
`target-server=` `<DNS name or IPv4 address>`	Specifies the iSCSI target's DNS name or IPv4 address to be used by the adapter when `target-dhcp` is disabled. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot boot-type=iscsi target-dhcp=disabled target-server=192.168.2.2`
`target-port=<port number>`	Specifies the Port number to be used by the iSCSI target when `target-dhcp` is disabled. The default Port number is Port 3260. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot boot-type=iscsi target-dhcp=disabled target-port=3262` This option should only be used if your target is using a non-standard TCP Port.
`target-lun=<LUN>`	Specifies the Logical Unit Number (LUN) to be used by the iSCSI target when `target-dhcp` is disabled. The default LUN is 0. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`).
`target-iqn=<IQN>`	Specifies the IQN of the iSCSI target when `target-dhcp` is disabled. Maximum of 223 characters. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Note that if there are spaces contained in `<IQN>`, then the IQN must be wrapped in double quotes (“”). Example: `sfboot target-dhcp=disabled target-iqn=iqn.2009.01.com.solarflare adapter=2`
`vendor-id=<vendor identifier>`	Specifies the device vendor ID to be advertised to the DHCP server. This must match the vendor id configured at the DHCP server when using DHCP option 43 to obtain the iSCSI target.

Table 50: Sfboot Per-adapter Parameters

Parameter	Description

`chap=enabled|disabled`	Enables or disables the use of Challenge Handshake Protocol (CHAP) to authenticate the iSCSI connection.
	
	Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`).
	
	To be valid, this option also requires the following sub-options to be specified:
	
	`username=<initiator username>`
	`secret=<initiator password>`
	
	Example:
	
	`sfboot boot-type=iscsi chap=enabled username=initiatorusername secret=initiatorsecret`

`username=<username>`	Specifies the CHAP initiator username (maximum 64 characters). Note that this option is required if either CHAP or Mutual CHAP is enabled (`chap=enabled, mutual-chap=enabled`). Note that if there are spaces contained in `<username>`, then it must be wrapped in double quotes (“”). Example: `sfboot boot-type=iscsi chap=enabled username=username`
`secret=<secret>`	Specifies the CHAP initiator secret (minimum 12 characters, maximum 20 characters). Note that this option is valid if either CHAP or Mutual CHAP is enabled (`chap=enabled, mutual-chap=enabled`). Note that if there are spaces contained in `<secret>`, then it must be wrapped in double quotes (“”). Example: `sfboot boot-type=iscsi chap=enabled username=username secret=veryverysecret`

Table 50: Sfboot Per-adapter Parameters

Parameter	Description

`mutual-chap=enabled|disabled`	Enables/disables Mutual CHAP authentication when iSCSI booting is enabled.
	This option also requires the following sub-options to be specified:
	`target-username=<username>`
	`target-secret=<password>`
	`username=<username>`
	`secret=<password>`
	Example:
	`sfboot boot-type=iscsi mutual-chap=enabled username=username secret=veryverysecret target-username=targetusername target-secret=anothersecret`

`target-username=<username>`	Specifies the username that has been configured on the iSCSI target (maximum 64 characters). Note that this option is necessary if Mutual CHAP is enabled on the adapter (`mutual-chap=enabled`). Note that if there are spaces contained in `<username>`, then it must be wrapped in double quotes (“”).
`target-secret=<secret>`	Specifies the secret that has been configured on the iSCSi target (minimum 12 characters; maximum 20 characters). Note: This option is necessary if Mutual CHAP is enabled on the adapter (`mutual-chap=enabled`). Note that if there are spaces contained in `<secret>`, then it must be wrapped in double quotes (“”).
`mpio-priority=<MPIO priority>`	Specifies the Multipath I/O (MPIO) priority for the adapter. This option is only valid for iSCSI booting over multi-port adapters, where it can be used to establish adapter port priority. The range is 1- 255, with 1 being the highest priority.
`mpio-attempts=<attempt count>`	Specifies the number of times MPIO will try and use each port in turn to login to the iSCSI target before failing.
`pf-count=<pf count>`	This is the number of available PCIe PFs per physical network port. This setting is applied to all ports on the adapter. Changes to this setting with `sfboot` require a cold reboot to become effective. MAC address assignments may change after altering this setting.

Table 50: Sfboot Per-adapter Parameters

Parameter	Description
`msix-limit=` `8|16|32|64|128|256|512|1024`	Specifies the maximum number of MSI-X interrupts that each PF will use. The default is 32. Note: Using the incorrect setting can impact the performance of the adapter. Contact Solarflare technical support before changing this setting.
`sriov=enabled|disabled`	Enable SR-IOV support for operating systems that support this. Not required on SFN7000 series adapters.
`vf-count=<vf count>`	The number of virtual functions (VF) advertised to the operating system for each Physical Function on this physical network port. SFN7000 series adapters have a total limit of 2048 interrupts. Earlier adapters support a total limit of 127 virtual functions per port and a total of 1024 interrupts. Depending on the values of msix-limit and vf-msix-limit, some of these virtual functions may not be configured. Enabling all 127 VFs per port with more than one MSI-X interrupt per VF may not be supported by the host BIOS - in which case you may get 127 VFs on one port and none on others. Contact your BIOS vendor or reduce the VF count. The sriov parameter is implied if vf-count is greater than zero. Changes to this setting with `sfboot` require a cold reboot to become effective.
`vf-msix-limit=` `1|2|4|8|16|32|64|128|256`	The maximum number of interrupts a virtual function may use.

Table 50: Sfboot Per-adapter Parameters

Parameter	Description
`pf-vlans=<tag>[,<tag>[,...]]|none`	Comma separated list of VLAN tags for each PF in the range 0-4094 - see sfboot --help for details.
`switch-mode= default|sriov|partitioning| partitioning-with-sriov|pfiov`	Specifies the mode of operation that the port will be used in: `default` - single PF created, zero VFs created. `sriov` - SR-IOV enabled, single PF created, VFs configured with `vf-count`. `partitioning` - PFs configured with `pf-count`, VFs configured with `vf-count`. See NIC Partitioning on page 58 for details. `partitioning-with-sriov` - SR-IOV enabled, PFs configured with `pf-count`, VFs configured with `vf-count`. See NIC Partitioning on page 58 for details. `pfiov` - PFIOV enabled, PFs configured with `pf-count`, VFs not supported. Changes to this setting with `sfboot` require a cold reboot to become effective.

Sfboot: Examples

- Show the current boot configuration for all adapters:

  `sfboot`

  Sample console output:

```
Solarflare boot ROM configuration utility [v4.1.4]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-2005

Solarflare SFN7122F SFP+ Server Adapter - MAC: 00:0F:53:21:9B:B1
  Boot image                            Option ROM only
    Link speed                          Negotiated automatically
    Link-up delay time                  5 seconds
    Banner delay time                   2 seconds
    Boot skip delay time                5 seconds
    Boot type                           Disabled
  PFIOV                                 Disabled
  Number of Physical Functions          2
  MSI-X interrupt limit                 32
  Number of Virtual Functions           0
  VF MSI-X interrupt limit              8
  Firmware variant                      full feature / virtualization
  Insecure filters                      Disabled

Solarflare SFN7122F SFP+ Server Adapter #2 - MAC: 00:0F:53:21:9B:B0
  Boot image                            Option ROM only
```



```
    Link speed                          Negotiated automatically
    Link-up delay time                  5 seconds
    Banner delay time                   2 seconds
    Boot skip delay time                5 seconds
    Boot type                           Disabled
  PFIOV                                 Disabled
  Number of Physical Functions          2
  MSI-X interrupt limit                 32
  Number of Virtual Functions           0
  VF MSI-X interrupt limit              8
  Firmware variant                      full feature / virtualization
  Insecure filters                      Disabled
```

- List all Solarflare adapters installed on the localhost:

  `sfboot /List`

  Sample console output:

```
Solarflare boot ROM configuration utility [v4.1.4]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-2005

Network adapters in this computer:
  1 : Solarflare SFN7122F SFP+ Server Adapter
      MAC address: 00:0F:53:21:9B:B1
  2 : Solarflare SFN7122F SFP+ Server Adapter #2
      MAC address: 00:0F:53:21:9B:B0
```

- List adapters installed on the remote host named “Mercutio”:

  `sfboot /Computer Mercutio /List`

  Sample console output (remote host has two adapters present):

```
Solarflare boot ROM configuration utility [v4.1.4]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-2005

Network adapters in Mercutio:
  1 : Solarflare SFN7122F SFP+ Server Adapter
      MAC address: 00:0F:53:21:9B:B1
  2 : Solarflare SFN7122F SFP+ Server Adapter #2
      MAC address: 00:0F:53:21:9B:B0
```

- Enable iSCSI booting on adapter 2. Implement default iSCSI settings:

  `sfboot /Adapter 2 boot-type=iscsi`

  Sample console output:

```
Solarflare boot ROM configuration utility [v4.1.4]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-2005

Solarflare SFN7122F SFP+ Server Adapter - MAC: 00:0F:53:21:9B:B1
  Boot image                            Option ROM only
    Link speed                          Negotiated automatically
    Link-up delay time                  5 seconds
    Banner delay time                   2 seconds
    Boot skip delay time                5 seconds
    Boot type                           iSCSI
      Use DHCP for Initiator            Enabled
      Use DHCP for Initiator IQN        Enabled
      LUN busy retries                  2
```

```
      Use DHCP for Target                Enabled
      DHCP Vendor Class ID               SFCgPXE
      CHAP authentication                Disabled
      MPIO priority                      0
      MPIO boot attempts                 3
  PFIOV                                  Disabled
  Number of Physical Functions           2
  MSI-X interrupt limit                  32
  Number of Virtual Functions            0
  VF MSI-X interrupt limit               8
  Firmware variant                       full feature / virtualization
  Insecure filters                       Disabled
```

- Enable iSCSI booting on adapter 1 with the following options:
  - Disable DHCP for the Initiator.
  - Specify adapter (iSCSI initiator) IP address 192.168.0.1 and netmask 255.255.255.0.

  ```
  sfboot /Adapter 1 boot-type=iscsi initiator-dhcp=disabled initiator-
  ip=192.168.0.1 netmask=255.255.255.0
  ```

  Sample console output:

```
Solarflare boot ROM configuration utility [v4.1.4]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-2005

Solarflare SFN7122F SFP+ Server Adapter - MAC: 00:0F:53:21:9B:B1
  Boot image                             Option ROM only
    Link speed                           Negotiated automatically
    Link-up delay time                   5 seconds
    Banner delay time                    2 seconds
    Boot skip delay time                 5 seconds
    Boot type                            iSCSI
      Use DHCP for Initiator             Disabled
        Initiator IP address             192.168.0.1
        Initiator netmask                255.255.255.0
        Initiator default gateway        0.0.0.0
        Initiator primary DNS            0.0.0.0
      Use DHCP for Initiator IQN         Enabled
      LUN busy retries                   2
      Use DHCP for Target                Enabled
      DHCP Vendor Class ID               SFCgPXE
      CHAP authentication                Enabled
        User name                        user1
        Secret                           *************
        Mutual CHAP authentication       Disabled
      MPIO priority                      0
      MPIO boot attempts                 3
  PFIOV                                  Disabled
  Number of Physical Functions           2
  MSI-X interrupt limit                  32
  Number of Virtual Functions            0
  VF MSI-X interrupt limit               8
  Firmware variant                       full feature / virtualization
  Insecure filters                       Disabled
```


- On adapter 1, set the following CHAP options:
  - User name “user1”
  - Secret “password12345”

  ```
  sfboot /Adapter 1 boot-type=iscsi chap=enabled username=user1
  secret=password12345
  ```

  Sample console output:

```
Solarflare boot ROM configuration utility [v4.1.4]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-2005

Solarflare SFN7122F SFP+ Server Adapter - MAC: 00:0F:53:21:9B:B1
  Boot image                            Option ROM only
    Link speed                          Negotiated automatically
    Link-up delay time                  5 seconds
    Banner delay time                   2 seconds
    Boot skip delay time                5 seconds
    Boot type                           iSCSI
      Use DHCP for Initiator            Enabled
      Use DHCP for Initiator IQN        Enabled
      LUN busy retries                  2
      Use DHCP for Target               Enabled
      DHCP Vendor Class ID              SFCgPXE
      CHAP authentication               Enabled
        User name                       user1
        Secret                          *************
        Mutual CHAP authentication      Disabled
      MPIO priority                     0
      MPIO boot attempts                3
  PFIOV                                 Disabled
  Number of Physical Functions          2
  MSI-X interrupt limit                 32
  Number of Virtual Functions           0
  VF MSI-X interrupt limit              8
  Firmware variant                      full feature / virtualization
  Insecure filters                      Disabled
```

4.21 Sfupdate: Firmware Update Tool

- Sfupdate: Command Usage on page 198
- Sfupdate: Command Line Options on page 198
- Sfupdate: Examples on page 199

Sfupdate is a Windows command line utility used to manage and upgrade the Solarflare adapter Boot ROM, PHY and adapter firmware. Embedded within the sfupdate executable are firmware images for various Solarflare adapters - the exact updates available via sfupdate are therefore depend on your adapter.

Sfupdate: Command Usage

1. Login with an administrator account.
2. Click **Start > All Programs > Solarflare Network Adapters > Command Line Interface for network adapters**. If you installed the Solarflare system tray icon, you can right-click the icon and choose **Command-line tools** instead.
3. In the Command Prompt window, enter your command using the following syntax:

   `sfupdate [/Adapter <Identifier>] [options]`

   where:

   - `Identifier` is the name or ID of the adapter that you want to manage. Specifying the adapter is optional - if it is not included the command is applied to all Solarflare adapters in the machine.
   - `options` is the option to apply. See Sfupdate: Command Line Options for a list of available options.

Running the command `sfupdate` with no additional parameters will show the current firmware version for all Solarflare adapters and whether the firmware within sfupdate is more up to date. To update the firmware for all Solarflare adapters run the command `sfupdate /Write`

Solarflare recommend that you use sfupdate in the following way:

1. Run `sfupdate` to check that the firmware on all your adapters are up to date.
2. Run `sfupdate /write` to update the firmware on all adapters.

Sfupdate: Command Line Options

Table 51 lists the command options for sfupdate. Note that command line options are case insensitive and may be abbreviated.

NOTE: Abbreviations in scripts should be avoided, since future updates to the application may render your abbreviated scripts invalid.

See Sfupdate: Examples on page 199 for example output.

Table 51: Sfupdate Options

Option	Description
`/Help` or `/H` or `/?`	Displays command line syntax and provides a description of each option.
`/Version`	Shows detailed version information and exits.
`/Nologo`	Hides the version and copyright message at startup.
`/Verbose`	Shows extended output information for the command entered.

Table 51: Sfupdate Options

Option	Description
`/Quiet` Aliases: `/Silent`	Suppresses all output, including warnings and errors; no user interaction. You should query the completion code to determine the outcome of commands when operating silently.
`/Log <Filename>`	Logs output to the specified file in the current folder or an existing folder. Specify `/Silent` to suppress simultaneous output to screen, if required.
`/Computer <ComputerName>`	Performs the operation on the identified remote computer. Administrator rights on the remote host computer is required.
`/Adapter <Identifier>`	Performs the action on the identified Solarflare network adapter. The identifier can be the adapter ID number, name or MAC address.
`/Force`	Forces a firmware update. Can be used to force an update to an older revision of firmware when used with `/Write`.
`/Write`	Writes the updated firmware to the adapter. If the `/Image` option is not specified, `/Write` will write the embedded image from sfupdate to the hardware. The update will fail if the image on the adapter is current or newer; to force an update, specify `/Force` in the command line.
`/Yes`	Update without prompting for a final confirmation. This option may be used with the `/Write` and `/Force` options, but is not required with the `/Quiet` option.
`/Image <ImageFileName>`	Sources firmware image from an external file.
`/NoWarning`	Suppress update warnings.

Sfupdate: Examples

- Display firmware versions for all adapters:

  `sfupdate`

  Sample output from a host with a single SFN7122F adapter installed:

```
Solarflare firmware update utility [v4.1.4]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-2005

1: Solarflare SFN7122F SFP+ Server Adapter
   MAC address: 00:0F:53:21:9B:B1
   Firmware:    v4.1.0                  - update to v4.1.4?
     Boot ROM:  v4.1.0.6723             - update to v4.2.0.1000?
     Adapter:   v4.1.0.6732             - update to v4.1.1.1020?

2: Solarflare SFN7122F SFP+ Server Adapter #2
   MAC address: 00:0F:53:21:9B:B0
```

```
   Firmware:     v4.1.0                      - update to v4.1.4?
     Boot ROM:   v4.1.0.6723                 - update to v4.2.0.1000?
     Adapter:    v4.1.0.6732                 - update to v4.1.1.1020?
```

- Update all adapters to latest version of PHY and Boot ROM firmware:

  `sfupdate /Write`

  Sample output:

```
Solarflare firmware update utility [v4.1.4]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-2005

1: Solarflare SFN7122F SFP+ Server Adapter
   MAC address: 00:0F:53:21:9B:B1
   Firmware:     v4.1.0                      - update to v4.1.4
     Boot ROM:   v4.1.0.6723                 - update to v4.2.0.1000
     Adapter:    v4.1.0.6732                 - update to v4.1.1.1020

2: Solarflare SFN7122F SFP+ Server Adapter #2
   MAC address: 00:0F:53:21:9B:B0
   Firmware:     v4.1.0                      - update to v4.1.4
     Boot ROM:   v4.1.0.6723                 - update to v4.2.0.1000
     Adapter:    v4.1.0.6732                 - update to v4.1.1.1020
```

4.22 Sfteam: Adapter Teaming and VLAN Tool

- Sfteam: Command Usage on page 200
- Sfteam: Command Line Options on page 201
- Sfteam: Examples on page 205

Sfteam is a Windows command line utility used to configure and manage the teaming and VLAN features of the Solarflare adapters. You may find it easier to create and manage teams and VLANs with SAM, Solarflare's graphical adapter manager. As an alternative, or where SAM is not available, sfteam provides a method of creating teams and VLANs from the command line or configuration script.

For general information on teaming and VLANs, see Teaming and VLANs on page 213.

Sfteam: Command Usage

1. Login with an administrator account.
2. Click **Start > All Programs > Solarflare Network Adapters > Command Line Interface for network adapters**. If you installed the Solarflare system tray icon, you can right-click the icon and choose **Command-line tools** instead.

3 In the Command Prompt window, enter your command using the following syntax:

```
sfteam [option]
```

where:

- `option` is the command to apply. See Table 52 for a list of available options.

If using sfteam in a configuration script, you can include the environment variable `%SFTOOLS%` to set the path to the Solarflare tools. For example:

```
SET PATH=%PATH%;%SFTOOLS%
```

or refer to sfteam as:

```
%SFTOOLS%\sfteam
```

Sfteam: Command Line Options

Table 52 lists the command line options sfteam. Note that command line options are case insensitive and may be abbreviated.

NOTE: Abbreviations in scripts should be avoided, since future updates to the application may render your abbreviated scripts invalid.

Table 52: Sfteam Options

Option	Description
`/Help` or `/?` or `/H`	Displays command line syntax and provides a description of each sfteam option.
`/Version`	Shows detailed version information and exits.
`/Nologo`	Hides the version and copyright message at startup.
`/Verbose`	Shows extended output information for the command entered.
`/Quiet` Aliases: `/Silent`	Suppresses all output, including warnings and errors; no user interaction. You should query the completion code to determine the outcome of commands when operating silently.
`/Log <Filename>`	Logs output to the specified file in the current folder or an existing folder. Specify `/Silent` to suppress simultaneous output to screen, if required.
`/Computer <ComputerName>`	Performs the operation on the identified remote computer. Administrator rights on the remote host computer is required.
`/List`	Lists all available Solarflare adapters and any teams and VLANs. This option shows the adapter's ID number, name and MAC address.

Table 52: Sfteam Options

Option	Description
`/Create`	Creates a team or VLAN. To be valid, this option must be used with the `/Adapter` option for each adapter that you want to add to the team. To specify a name for the team, include the `/Name` option. To add VLANs to a team, include the `/Vlan` option. Note that once a team has been created, sfteam does not allow you to change its adapters, VLANs or team name. Either delete the team and set it up again, or use SAM instead to configure the team.
`/Delete <TeamIdentifier>`	Deletes the identified team or group. The team identity can be specified as the team name or group ID. This option cannot be used to delete VLANs.
`/Clear`	Deletes all teams and VLANS.
`/Adapter <AdapterId>`	Specifies the adapter to add to the team. Repeat this option for each adapter that you want to include in the team. This option must be used when a team is first created. It cannot be applied to a team once it has been setup.

Table 52: Sfteam Options

Option	Description
`/Vlan <VLAN tag[,priority[,name [,DHCP|addr,mask[,gateway]]]>`	Creates a VLAN with the specified ID and sets priority traffic handling option. P – Handles priority traffic N - Does not handle priority traffic This option must be used when a team is first created. It cannot be applied to a team once it has been setup. If you specify an IP address, you must specify a netmask as well. If the IP address is not specified, then DHCP is assumed. You can also use `tag,priority,name,DHCP` to be explicit. Formats: • <tag> e.g. 2 (assumes no priority) • "<tag>,<priority>" e.g. "2,p" • "<tag>,<priority>,<name>" e.g. "2,p,my name" • "<tag>,<priority>,<name>,DHCP" e.g. "2,p,my name,DHCP" • "<tag>,<priority>,<name>,<addr>,<mask>" e.g. "2,p,my name,10.1.2.3,255.255.255.0" • "<tag>,<priority>,<name>,<addr>,<mask>,<gateway>" e.g. "2,p,my name,10.1.2.3,255.255.255.0,10.1.2.1" where: • Tag: 0 to 4094 • Priority: either P (priority supported) or N (no priority) • DHCP: may be omitted, and will be assumed, if it's the last field • IP Addresses: IPv4, dotted-quad format Note that <mask> must be present if <addr> is present
`/Name <TeamName>`	Specifies a name for the adapter team. This option must be used when a team is first created. It cannot be applied to a team once it has been setup.

Table 52: Sfteam Options

Option	Description
`/DebugId <AdapterId>`	Debug-only. Identify an adapter id to treat as being as iSCSI boot device.
`/DebugIscsi`	Debug-only. Pretend the adapter is configured for iSCSI booting.
`/Type <TeamType>`	Defines what kind of team is being created. The options are: • tolerant (default) • dynamic • static See Teaming and VLANs on page 213 for an explanation on the different teaming types.
`/Mode <Mode>`	Specifies how the driver will select adapters to be part of the link aggregation. The option is only relevant when the / Type option is either dynamic or static. The options are: • auto (default) • faulttolerant • bandwidth • key adapter See Teaming and VLANs on page 213 for an explanation of the different teaming modes.
`/Distribution <DistributionMode>`	Specify how the driver distributes conversations across dynamic or static link aggregation team members. The available modes are: • auto (default) • activeadapter • layer2hash • layer3hash • layer4hash
`/Statistics`	Display adapter and link-aggregation statistics
`/Detailed`	Display detailed configuration statistics

Sfteam: Examples

- Create **TeamA** with adapter ID 1 and adapter ID 2:

  `sfteam /Create /Adapter 1 /Adapter 2 /Name Team_A`

  Sample output:

```
Solarflare teaming configuration utility [v4.1.4]
Copyright Solarflare Communications 2006-2014 Level 5 Networks 2002-2005

Creating team done (new id=2F)
Setting team name "Team_A" ...  done
Adding adapter 1 ...  done
Adding adapter 2 ...  done
Creating network interface
 - Using DHCP
 - Waiting for the new VLAN device ..
 - Waiting for the new LAN interface
 - Waiting for access to the IP stack
 - Using DHCP  done
```

- Create a VLAN to adapter #2 with VLAN tag 4 and priority traffic handling enabled:

  `sfteam /Create /Adapter 2 /Vlan 4,P`

  Sample output:

```
Solarflare teaming configuration utility [v4.1.4]
Copyright Solarflare Communications 2006-2014 Level 5 Networks 2002-2005

Creating VLAN group done (new id=4V)
Setting VLAN group name (using default name "Group 4V") ...  done
Adding adapter 2 ...  done
Creating VLAN
 - id=4, priority, unnamed
 - Using DHCP
 - Waiting for the new VLAN device ..
 - Waiting for the new LAN interface
 - Waiting for access to the IP stack
 - Using DHCP  done
```

4.23 Sfcable: Cable Diagnostics Tool

- Sfcable: Command Usage on page 206
- Sfcable: Command Line Options on page 206
- Sfcable: Sample Commands on page 207

Sfcable is a Windows command line utility to run cable diagnostics on the Solarflare 10GBASE-T server adapters. A warning will be given if the adapter is not a 10GBASE-T adapter.

Sfcable: Command Usage

1. Login with an administrator account.
2. Click **Start > All Programs > Solarflare Drivers > Command Line Tools**. If you installed the Solarflare system tray icon, you can right-click the icon and choose **Command-line tools** instead.
3. In the Command Prompt window, enter the following command:

```
sfcable [/Adapter <Identifier>] [options]
```

where:

- `Identifier` is the name or ID of the adapter that you want to manage. Specifying the adapter is optional - if it is not included the command is applied to all Solarflare adapters in the machine.
- `option` is the option you to apply. See Table 53 for a list of available options.

Sfcable: Command Line Options

Table 53 lists the command options for sfcable. Note that command line options are case insensitive and may be abbreviated.

NOTE: Abbreviations in scripts should be avoided, since future updates to the application may render your abbreviated scripts invalid.

Table 53: Sfcable Options

Options	Description
`/Help` or `/?` or `/H`	Displays command line syntax and provides a description of each sfcable option.
`/Version`	Shows detailed version information and exits.
`/Nologo`	Hides the version and copyright message at startup.
`/Verbose`	Shows extended output information for the command entered.
`/Quiet` Aliases: `/Silent`	Suppresses all output, including warnings and errors. User should query the completion code to determine the outcome of commands when operating silently (see, Performance Tuning on Windows on page 225).
`/Log <Filename>`	Logs output to the specified file in the current folder or an existing folder. Specify `/Silent` to suppress simultaneous output to screen, if required.
`/Computer <ComputerName>`	Performs the operation on the identified remote computer. Administrator rights on the remote host computer is required.

Table 53: Sfcable Options

Options	Description
`/Adapter <Identifier>`	Performs the action on the identified Solarflare network adapter. The identifier can be the adapter ID number, name or MAC address, as given by the `/List` option.
`/List`	Lists all available Solarflare adapters. This options shows the adapter's ID number, name and MAC address.
`/Offline`	Stops network traffic while the diagnostic tests are running. Running tests offline will produce more detailed results. **Caution**: The offline tests will disrupt data flow. It is not recommended that the tests are run on a live system.
`/DebugId (AdapterId>`	Debug-only. Identify an adapter to treat as being an iSCSI boot device.

Sfcable: Sample Commands

- Run tests offline

  `sfcable /Offline`

  Sample output from a computer with two Solarflare adapters installed:

```
C: sfcable /Offline
Solarflare cable diagnostics utility [v4.1.4]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-2005

1 : Solarflare SFN5121T 10GBASE-T Server Adapter
    MAC address:         00:0F:53:01:40:8C
    Link state:          Up
    Link speed:          10 Gbps
      Pair 1:            OK, length=9m
      Pair 2:            OK, length=9m
      Pair 3:            OK, length=9m
      Pair 4:            OK, length=9m

2 : Solarflare SFN5121T 10GBASE-T Server Adapter #2
    MAC address:         00:0F:53:01:40:8D
    Link state:          Up
    Link speed:          10 Gbps
      Pair 1:            OK, length=9m
      Pair 2:            OK, length=9m
      Pair 3:            OK, length=9m
      Pair 4:            OK, length=9m
```


4.24 Sfnet

Sfnet is a Windows command line utility to configure the physical or virtual adapter settings, such as checksum offloading, RSS, VMQ and Power Management.

NOTE: Changing these settings may significantly alter the performance of the adapter. You should contact Solarflare technical support before changing any of these settings.

Sfnet: Command Usage

1 Login with an administrator account.

2 Click **Start > All Programs > Solarflare Network Adapter > Command Line Interface for network adapters**. If you installed the Solarflare system tray icon, you can right-click the icon and choose **Command-line tools** instead.

3 In the Command Prompt window, enter your command using the following syntax:

```
sfnet [/Adapter Identifier] [options]
```

where:

- `Identifier` is the name or ID of the adapter that you want to manage. Specifying the adapter is optional - if it is not included the command is applied to all Solarflare adapters in the machine.
- `option` is the option to apply. See Sfnet: Command Line Options for a list of available options.

To see all adapters installed on the computer and their current options and parameter settings use the `sfnet /List` option.

If using sfnet in a configuration script, you can include the environment variable %SFTOOLS% to set the path to the Solarflare tools. For example:

```
SET PATH=%PATH%;%SFNET%
```

or refer to sfnet as:

```
%SFTOOLS%\sfnet
```


Sfnet: Command Line Options

Table 54 lists the command options for sfnet. Note that command line options are case insensitive and may be abbreviated.

NOTE: Abbreviations in scripts should be avoided, since future updates to the application may render your abbreviated scripts invalid.

Table 54: Sfnet Options

Options	Description
`/Help` or `/?` or `/H`	Displays command line syntax and provides a description of each sfnet option.
`/Version`	Shows detailed version information and exits.
`/Nologo`	Hides the version and copyright message at startup.
`/Verbose`	Shows extended output information for the command entered.
`/Quiet` Aliases: `/Silent`	Suppresses all output, including warnings and errors; no user interaction. You should query the completion code to determine the outcome of commands when operating silently.
`/Log <Filename>`	Logs output to the specified file in the current folder or an existing folder. Specify silent to suppress simultaneous output to screen, if required.
`/Computer <ComputerName>`	Performs the operation on the identified remote host. Administrator rights on the remote host computer is required.
`/Adapter <Identifier>`	Perform the action on the identified Solarflare physical or virtual network adapter.
`/List`	Lists all available Solarflare adapters, options and current parameter settings.
`/Id`	List output is limited to one line, containing the Id and name, per adapter.
`/StopOnWarning`	Exit the utility if a warning is output.
`/Statistics`	Display adapter statistics and configuration settings for Solarflare interfaces.

Table 55: Supported Key Value Parameter

Parameter	Description
`ipoffload=enabled|disabled`	Specify whether IPv4 checksum offload is enabled.
`tcpoffload=enabled|disabled`	Specify whether TCP checksum offload is enabled. Configures TCPv4 and TCPv6 where applicable.

Table 55: Supported Key Value Parameter

Parameter	Description
`udpoffload=enabled|disabled`	Specify whether UDP checksum offload is enabled. Configures UDPv4 and UDPv6 where applicable.
`lso=enabled|disabled`	Specify whether large send offload (LSO) is enabled. Configures LSOv4 and LSOv6 where applicable.
`lro=enabled|disabled`	Specify whether large receive offload (LRO) is enabled. Configures RSCv4 and RSCv6, or LROv4 and LROv6, where applicable. Support for this option is dependent on the version of Windows operating system and networking stack. Implements Windows Receive Segment Coalescing (RSC) if applicable.
`flowcontrol=auto|enabled|generate|respond|disabled`	Specify Ethernet flow control. This option covers the “Flow Control” and “Flow Control Autonegotiation” device driver advanced properties.
`speed=auto|40g|10g|1g|100m`	Specify the Ethernet link speed.
`mtu=<MTU length>`	Specify the maximum Ethernet frame length. From 1518 to 9216 bytes (even values only).
`rss=disabled|optimized|system|closest|closeststatic|numa|numastatic|conservative`	Specify the receive side scaling (RSS) mode.
`rssbaseprocessor=<group>:<number>`	The base processor available for RSS. If a value is given it must formated as <group>:<number> where group is in the range 0-9 and number in the range 0 to 63.
`rssmaxprocessor=<group>:<number>`	The maximum number of processors available for RSS. If a value is given it must formated as <group>:<number> where group is in the range 0-9 and number in the range 0 to 63.
`maxrssprocessors=<count>`	The maximum number of processors available for RSS. If count is specified it must be in the range 1-256. Support for this option is independent of the version of the operating system and networking stack.
`rssqueuecount=balanced|<value>`	Specify the maximum number of receive queues to use for RSS. If set to balanced the network adapter will choose the number of queues based on the system processor topology. If specified, count must be one of 1|2|4|8|12|16|24|32|48|64. Support for this option is independent of the version of the operating system and networking stack.

Table 55: Supported Key Value Parameter

Parameter	Description
`numanode=all|<value>`	The preferred NUMA node used by RSS. If a value is given, it must be in the range 0-15. Support for this option is independent of the version of the operating system and networking stack.
`moderation=disabled|<value>`	Specify interrupt moderation time (in micro-seconds). If a value is given it must be in the range 1 to 1000. NOTE: this option covers the device driver advanced properties “interrupt moderation time” and “interrupt moderation”.
`adaptive=enabled|disabled`	Allows the adapter to vary interrupt moderation automatically if `interruptmoderation` is enabled.
`wake=enabled|disabled`	Specify whether Wake-on-LAN is enabled.
`sleep=enabled|disabled`	Specify whether the operating system can put the device to sleep when the physical link goes down.
`vmq=enabled|nosplit|novlan|basic|disabled`	enabled = VMQ enabled. nosplit = VMQ enabled without lookahead split. novlan = VMQ enabled without VLAN filtering. basic = VMQ enabled MAC address filtering only. disabled = VMQ disabled.

4.25 Completion codes (%errorlevel%)

Table 56 lists the completion codes returned by the command line utilities. The code may be determined by inspecting `%errorlevel%`

Table 56: Completion Codes

Error code	Description
`0`	Success.
`1`	The application was invoked with /? or /help.
`3`	The application was invoked with /version.
`16`	Application canceled (user probably pressed CTRL-C).
`17`	Application has requested a reboot.
`18`	Reboot is necessary to complete the action.

Table 56: Completion Codes

Error code	Description
19	Incomplete team creation. Team has been created and whatever adapters that could be added have been, and the VLANs (if any) have been created. Some adapters were not able to be added.
32	Application failed initialization.
33	Access denied. Either the remote host refused a connection on the basis of account privileges, or a file could not be opened.
34	Cannot connect. The remote host could not be found or refused the connection because the WMI service was inaccessible (either because the service is not running or because there is a firewall or security policy preventing it being accessed remotely).
35	WMI classes exposed by the Solarflare drivers missing. Usually this means that either the driver have not been installed, no Solarflare adapters are present, or adapters have been disabled.
36	Failed to obtain driver lock. The application has tried to take the Solarflare driver lock because it wants to do something that must not be interrupted by another utility (or SAM) and failed to do so.
37	Adapter not found. Cannot find the adapter specified by /adapter.
38	Adapter not specified. Command line is missing the /adapter option.
39	Later version already installed.
128	User entered an invalid command line.
129	Could not open log file.
130	A general WMI error occurred. Can occur when the connection is lost.
131	Missing prerequisite. The application needs something that is not present in the system.
132	Not supported.

Table 56: Completion Codes

Error code	Description
133	Platform/System not supported.
255	General exit failure.

4.26 Teaming and VLANs

About Teaming

Solarflare adapters support the following teaming configurations:

- IEEE 802.1AX (802.3ad) Dynamic link aggregation
- Static link aggregation
- Fault tolerant teams

Teaming allows the user to configure teams consisting of all Solarflare adapter ports on all installed Solarflare adapters or might consist only of selected ports e.g. from a dual port Solarflare adapter, the first port could be a member of team A and the second port a member of team B or both ports members of the same team.

This section is only relevant to teams of Solarflare adapters. Solarflare adapters can be used in multi-vendor teams when teamed using another vendor's teaming driver.

NOTE: Adapter teaming and VLANs are not supported in Windows for iSCSI remote boot enabled Solarflare adapters. To configure load balancing and failover support on iSCSI remote boot enabled adapters, you can use Microsoft MultiPath I/O (MPIO), which is supported on all Solarflare adapters.

NOTE: Windows Server 2012 has native Windows teaming support. The user can elect to use native Windows driver of the Solarflare teaming, but the two methods should not be mixed.

Creating Teams and VLANs

To set up teams and VLANs in Windows using SAM, see Using SAM to Configure Teams and VLANs on page 157.

To set up teams and VLANs in Windows using the sfteam command line tool, see Sfteam: Adapter Teaming and VLAN Tool on page 200.

Link Aggregation

Link aggregation is a mechanism for supporting load balancing and fault tolerance across a team of network adapters and their associated switch. Link aggregation is a partner teaming mode that requires configuration at both ends of the link. Once configured, all links in the team are bonded into a single virtual link with a single MAC address.

Two or more physical links are used to increase the potential throughput available between the link partners, and also improve resilience against link failures. To be aggregated, all links in the team must be between the same two link partner and each link must be full-duplex. Traffic is distributed evenly to all links connected to the same switch. In case of link failover, traffic on the failed link will be re-distributed to the remaining links.

Link aggregation offers the following functionality:

- Teams can be built from mixed media (i.e. UTP and Fiber).
- All protocols can be load balanced without transmit or receive modifications to frames.
- Multicast and broadcast traffic can be load balanced.
- Short recovery time in case of failover.
- Solarflare supports up to 64 link aggregation port groups per system.
- Solarflare supports up to 64 ports and VLANs in a link aggregation port group.

There are two methods of link aggregation, dynamic and static.

Dynamic Link Aggregation

Dynamic link aggregation uses the Link Aggregation Control Protocol (LACP) as defined in the IEEE 802.1AX standard (previously called 802.3ad) to negotiate the ports that will make up the team. LACP must be enabled at both ends of the link for a team to be operational.

LACP will automatically determine which physical links can be aggregated, and will then perform the aggregation.

An optional LACP marker protocol provides functionality when adding and removing physical links ensuring that no frames are lost, reordered or duplicated.

Dynamic link aggregation offers both fault tolerance and load balancing.

Standby links are supported, but are not considered part of a link aggregation until a link within the aggregation fails.

VLANs are supported within 802.1AX teams.

In the event of failover, the load on the failed link is redistributed over the remaining links.

NOTE: Your switch must support 802.1AX (802.3ad) dynamic link aggregation to use this method of teaming.

Figure 37 shows a 802.1AX Team configuration.

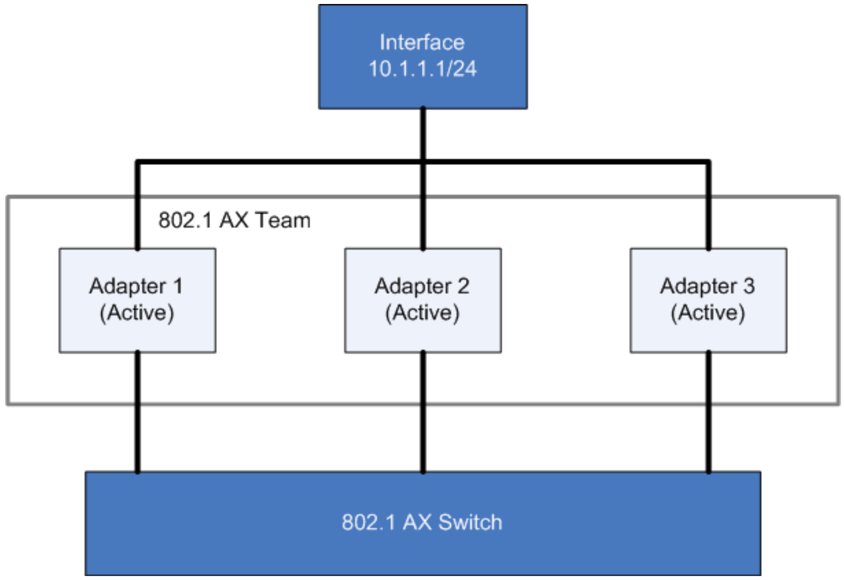


Figure 37: 802.1AX Team

Figure 38 shows a 802.1AX team with a failed link. All traffic is re-routed and shared between the other team links.

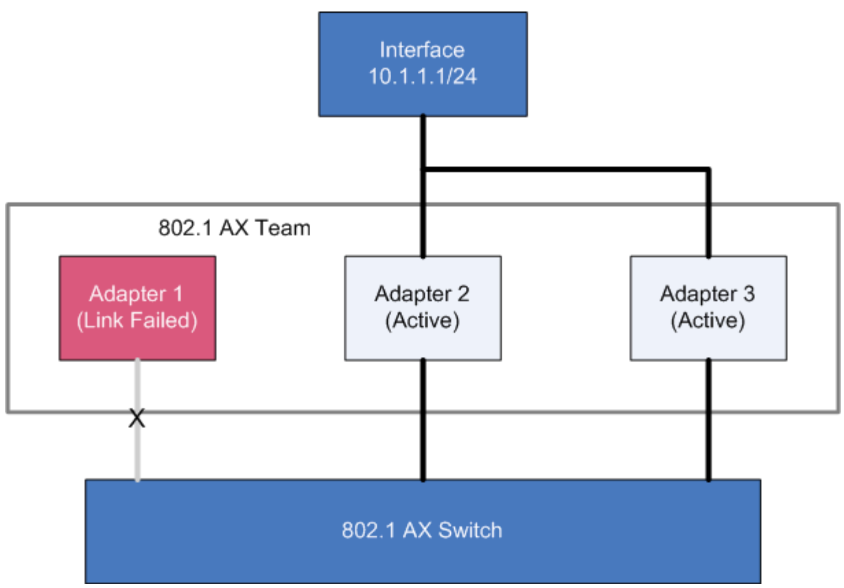


Figure 38: 802.1AX with Failed Link

Static Link Aggregation

Static link aggregation is a switch assisted teaming mode that requires manual configuring of the ports at both ends of the link. Static link aggregation is protocol independent and typically inter-operates with common link aggregation schemes such as Intel Link Aggregation, Cisco Fast EtherChannel and Cisco Gigabit EtherChannel.

With static link aggregation, all links share the traffic load and standby links are not supported. Static link aggregation offers both fault tolerance and load balancing. In the event of failover, the load on the failed link is redistributed over the remaining links.

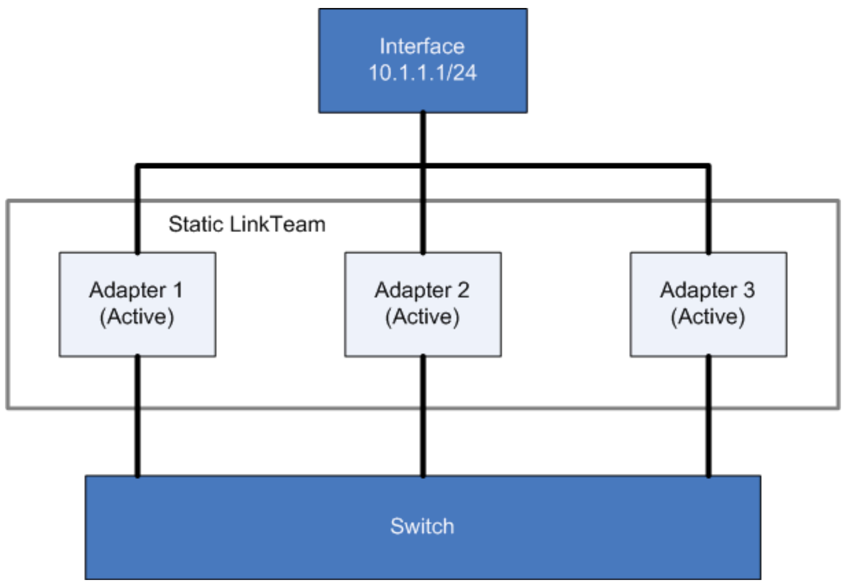


Figure 39: Static Link Aggregation Team

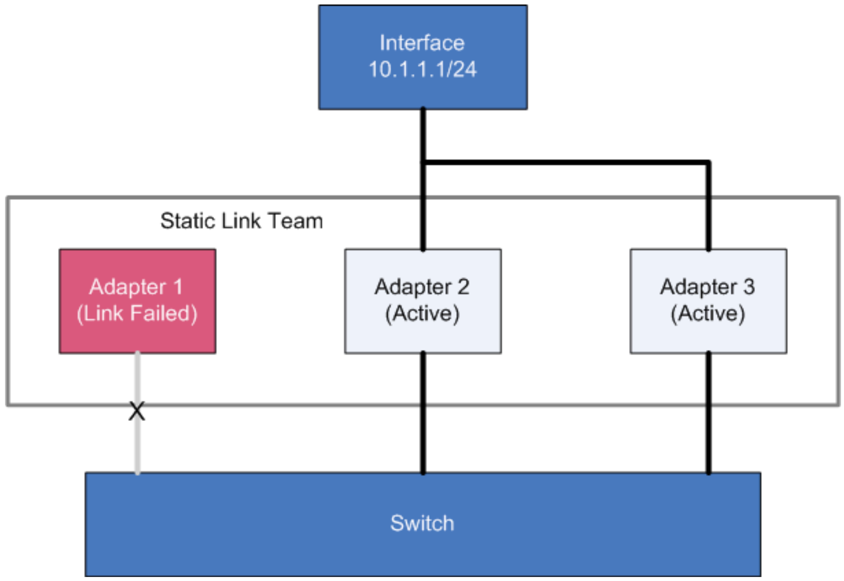


Figure 40: Static Link Team with Failed Link

Fault-Tolerant Teams

Fault tolerant teaming can be implemented on any switch. It can also be used with each network link connected to separate switches.

A fault-tolerant team is a set of one or more network adapters bound together by the adapter driver. A fault-tolerant team improves network availability by providing standby adapters. At any one moment no more than one of the adapters will be active with the remainder either in standby or in a fault state. In Figure 41, Adapter 1 is active and all data to and from the switch passes through it.

NOTE: All adapters in a fault-tolerant team must be part of the same broadcast domain.

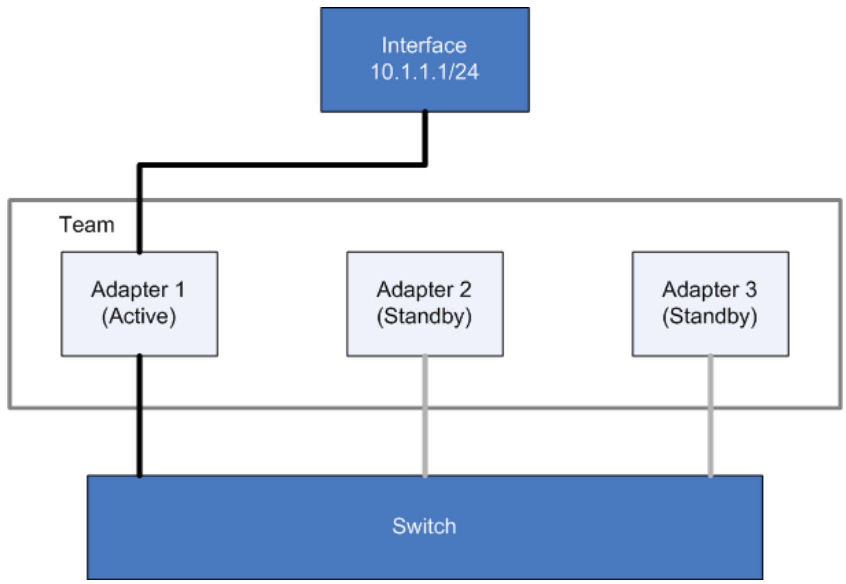


Figure 41: Fault Tolerant Team

Failover

The teaming driver monitors the state of the active adapter and, in the event that its physical link is lost (down) or that it fails in service, swaps to one of the standby adapters. In Figure 42 the previously active adapter has entered a failed state and will not be available in the standby list while the failed state persists.

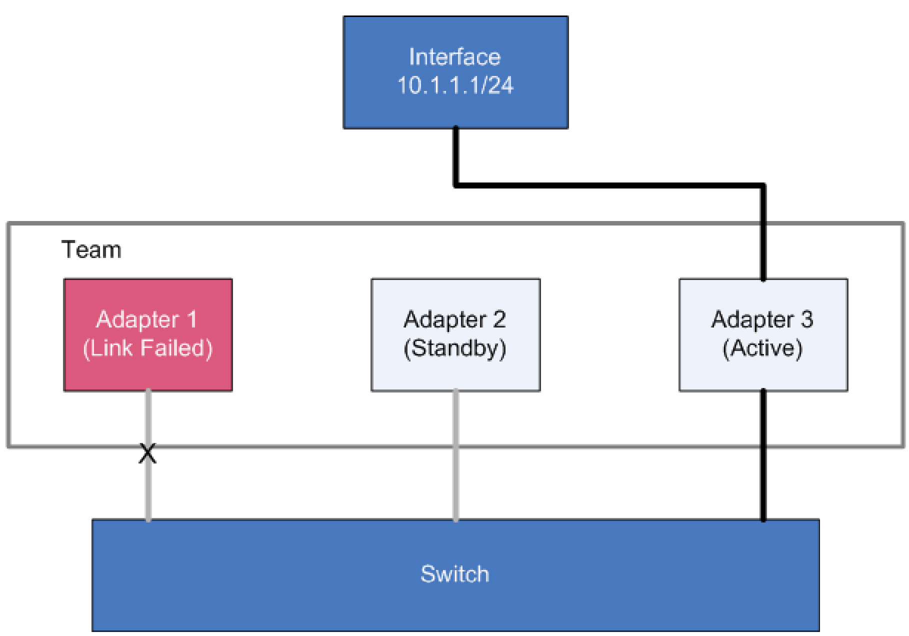


Figure 42: Adapter 1 Failure

Note that, in this example, Adapter 3 is now active. The order in which the adapters are used is determined by a number of factors, including user-definable rank.

VLANs

VLANs offer a method of dividing one physical network into multiple broadcast domains.

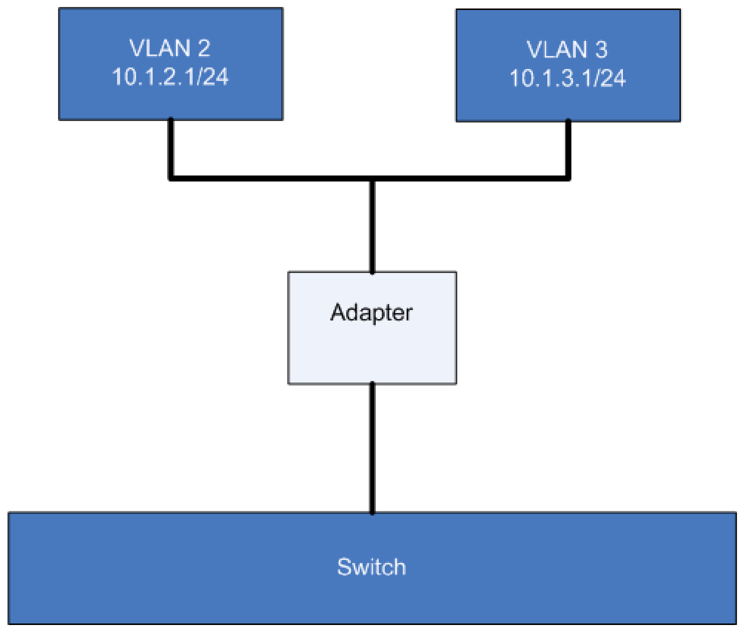


Figure 43: VLANs routing through Solarflare adapter

VLANs and Teaming

VLANs are supported on all Solarflare adapter teaming configurations.

VLANs with Fault Tolerant Teams

Figure 44 shows a fault tolerant team with two VLANs.

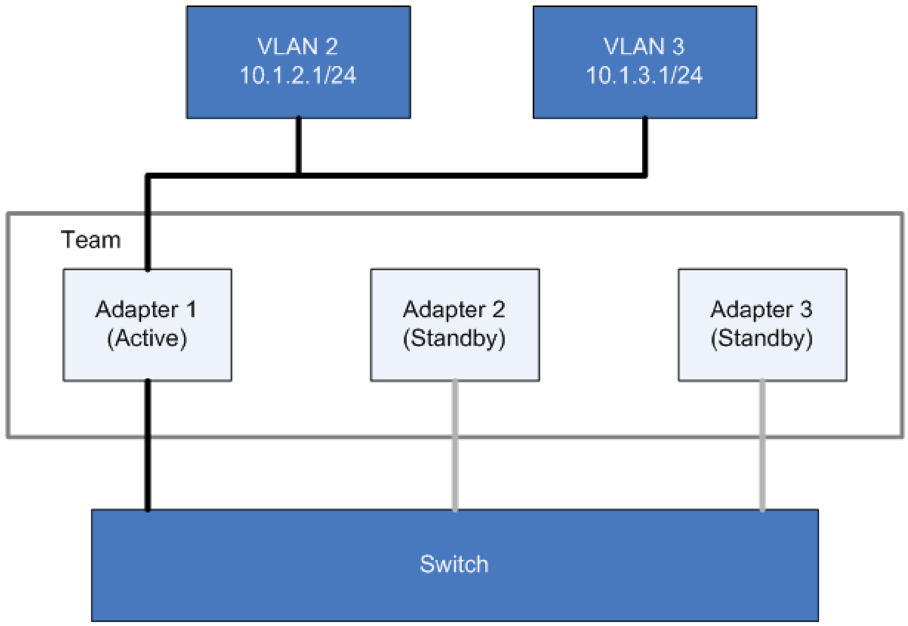


Figure 44: Fault Tolerant VLANs

Failover works in the same way regardless of the number of VLANs, as show in Figure 45.

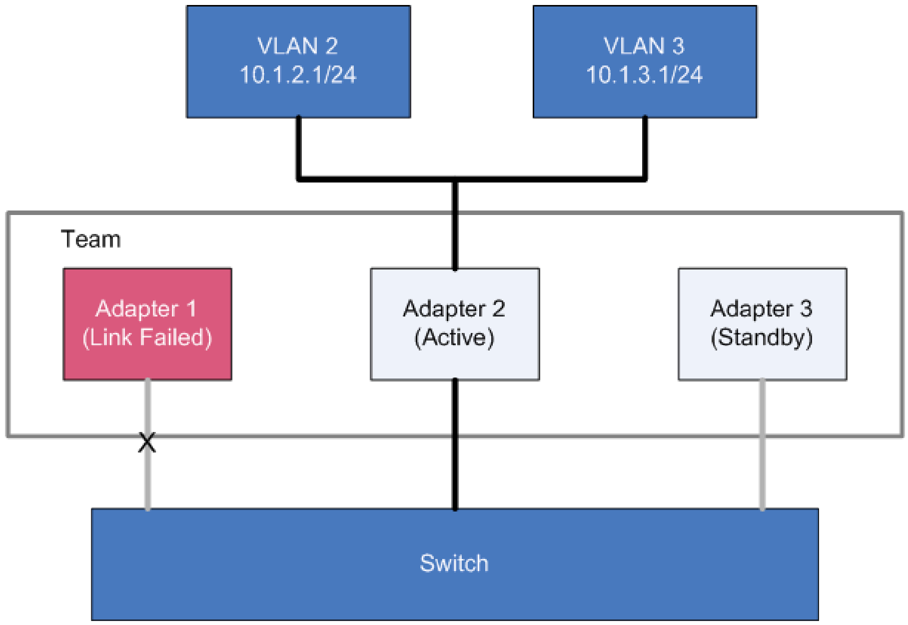


Figure 45: Failover in Fault Tolerant Team VLAN

VLANs with Dynamic or Static Link Aggregation Teams

VLANs work in the same way with either Dynamic or Static Link Aggregation teaming configurations.Figure 46 shows how VLANs work with these teams.

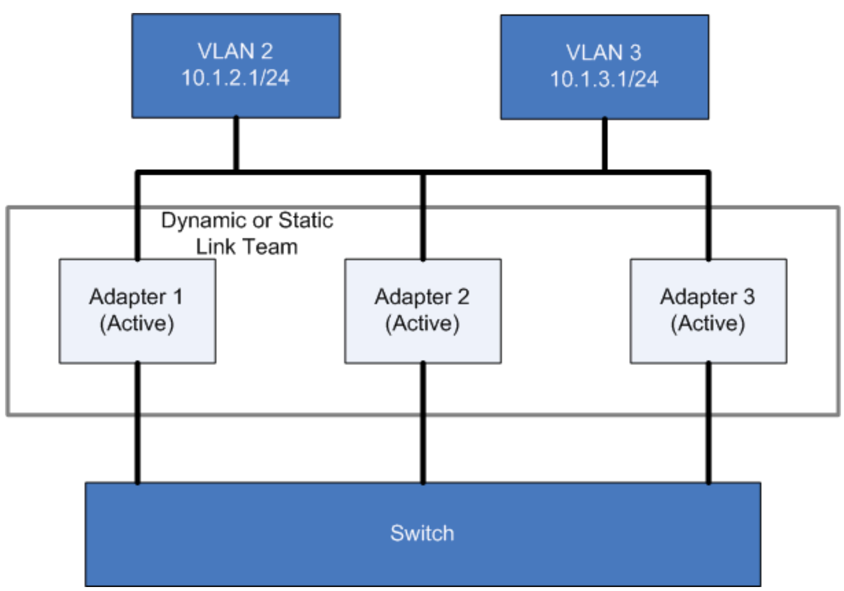


Figure 46: VLAN with Dynamic or Static Link Team

In case of link failure, all traffic is distributed over the remaining links, as in Figure 47.

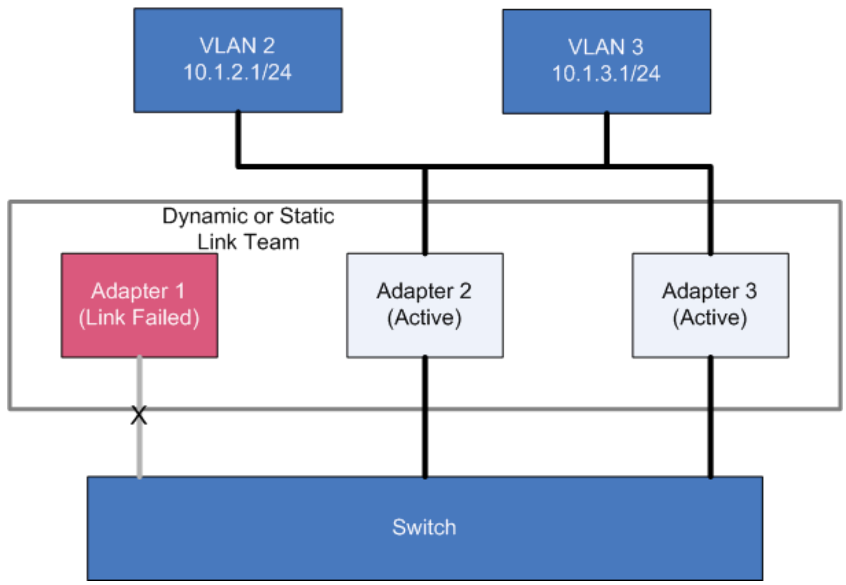


Figure 47: VLAN with Failed Dynamic or Static Team Link

Key Adapter

Every team must have a key adapter. Figure 48 shows Adapter 1 as both the Key and the active adapter. in a Fault-Tolerant Team.

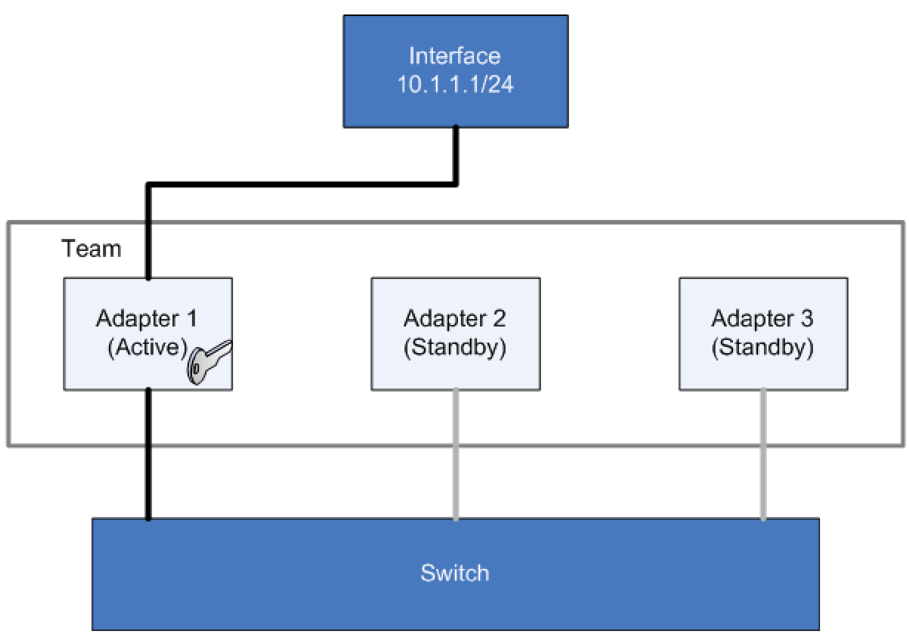


Figure 48: Key Adapter in Fault Tolerant Team

The key adapter must be a member of a team. However, it does not need to be the active adapter. It doesn't even need to be in the list of standby adapters but it must physically be within its host. The Key Adapter defines the team's RSS support (see Receive Side Scaling (RSS) on page 230) and provides the MAC Address that will be used for all traffic sent and received by the team.

When a link failure occurs in the active adapter (for example the physical link is lost) the driver will select another adapter to become active but it will not re-assign the Key Adapter. In Figure 49, Adapter 1 has failed and the team is now using Adapter 2 for all traffic.

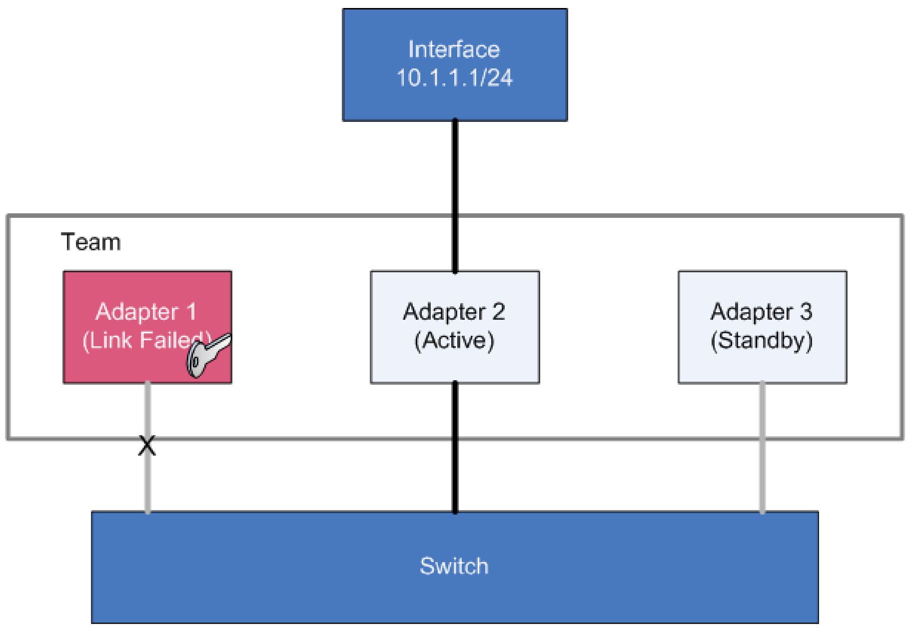
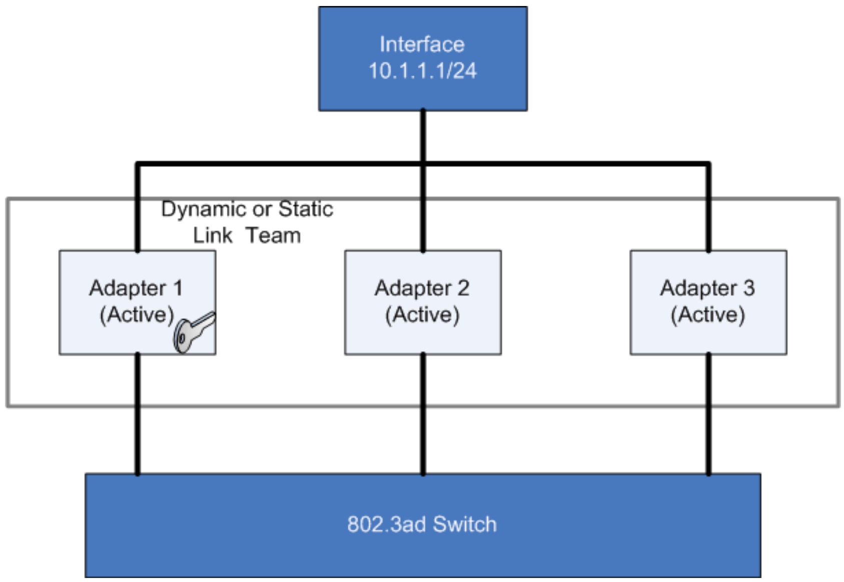


Figure 49: Failover Key Adapter

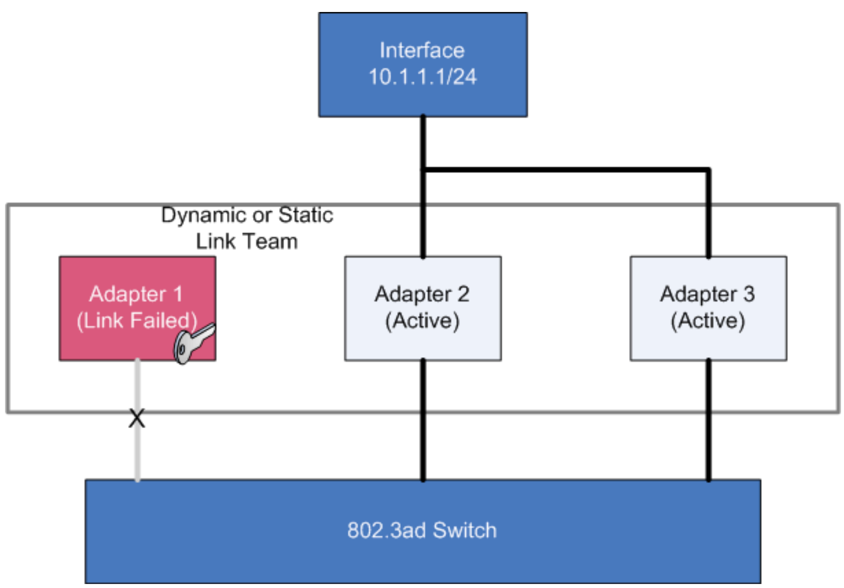
Note that although the Key Adapter (Adapter 1) has a link failure, the integrity of the team is not affected by this failure.

Dynamic and Static Link Aggregation Teams

The assignment of key adapters is supported in both dynamic and static link aggregated teams, and works in the same way for both.



Any link failure on the key adapter does not affect the redistribution of traffic to the other links in the team.



4.27 Performance Tuning on Windows

Introduction

The Solarflare family of network adapters are designed for high-performance network applications. The adapter driver is pre-configured with default performance settings that have been designed to give good performance across a broad class of applications. Occasionally, application performance can be improved by tuning these settings to best suit the application.

There are three metrics that should be considered when tuning an adapter:

- Throughput
- Latency
- CPU utilization

Different applications may be more or less affected by improvements in these three metrics. For example, transactional (request-response) network applications can be very sensitive to latency whereas bulk data transfer applications are likely to be more dependent on throughput.

The purpose of this section is to highlight adapter driver settings that affect the performance metrics described. This section covers the tuning of all Solarflare adapters.

Latency will be affected by the type of physical medium used: 10GBase-T, twinaxial (direct-attach), fiber or KX4. This is because the physical media interface chip (PHY) used on the adapter can introduce additional latency. Likewise, latency can also be affected by the type of SFP/SFP+/QSFP module fitted.

This section is designed for performance tuning Solarflare adapters on Microsoft Windows. This should be read in conjunction with the reference design board errata documents and the following Microsoft performance tuning guides:

- Performance Tuning Guidelines for the current version of Windows Server:
  http://msdn.microsoft.com/en-us/library/windows/hardware/dn529133
- Performance Tuning Guidelines for previous versions of Windows Server:
  http://msdn.microsoft.com/en-us/library/windows/hardware/dn529134.

In addition, you may need to consider other issues influencing performance, such as application settings, server motherboard chipset, CPU speed, cache size, RAM size, additional software installed on the system, such as a firewall, and the specification and configuration of the LAN. Consideration of such issues is not within the scope of this guide.

Tuning Settings

Tuning settings for the Solarflare adapter are available through the Solarflare Adapter Manager (SAM) utility, or via the **Advanced** tab in the **Windows Device Manager** (right-click the adapter and select **Properties**). See Using SAM to Configure Adapter Features on page 150 and Configuring Network Adapter Properties in Windows on page 177 for more details.

Table 57 lists the available tuning settings for Solarflare adapters on Windows.

Table 57: Tuning Settings

Setting	Supported on Windows Server 2008 R2	Supported on Windows Server 2012 / Windows Server 2012 R2
Adaptive Interrupt Moderation	Yes	Yes
Interrupt Moderation	Yes	Yes
Interrupt Moderation Time	Yes	Yes
Large Receive Offload (IPv4)	Yes	No[1]
Large Receive Offload (IPv6)	Yes	No[1]
Large Send Offload V2 (IPv4)	Yes	Yes
Large Send Offload V2 (IPv6)	Yes	Yes
Max Frame Size	Yes	Yes
Offload IPv4 Checksum	Yes	Yes
Port mode	Yes	Yes
Preferred Numa Node	Yes	Yes
Receive Segment Coalescing (IPv4)	No[1]	Yes
Receive Segment Coalescing (IPv6)	No[1]	Yes
Receive Side Scaling	Yes	Yes
RSS Interrupt Balancing	Yes	Yes
TCP Checksum Offload (IPv4)	Yes	Yes
TCP Checksum Offload (IPv6)	Yes	Yes
UDP Checksum Offload (IPv4)	Yes	Yes
UDP Checksum Offload (IPv6)	Yes	Yes

1. Large Receive Offload has been superseded by Receive Segment Coalescing (RSC). Microsoft introduced RSC in Windows Server 2012.

Port mode

The selected port mode for SFN7000 series adapters should correspond to the speed and number of connectors in use, after using any splitter cables. If a restricted set of connectors is configured, the driver can then transfer resources from the unused connectors to those configured, potentially improving performance.

Max Frame Size

The default *maximum frame size* ensures that the adapter is compatible with legacy 10/100Mbps Ethernet endpoints. However if a larger maximum frame size is used, adapter throughput and CPU utilization can be improved. CPU utilization is improved, because it takes fewer packets to send and receive the same amount of data. Solarflare adapters support maximum frame sizes up to 9216 bytes (this does not include the Ethernet preamble or frame check sequence).

NOTE: The maximum frame size setting should include the Ethernet frame header. The Solarflare drivers support 802.1p. This allows Solarflare adapters on Windows to optionally transmit packets with 802.1Q tags for QoS applications. It requires an Ethernet frame header size of 18 bytes (6 bytes source MAC address, 6 bytes destination MAC address, 2 bytes 802.1Q tag protocol identifier, 2 bytes 802.1Q tag control information, and 2 bytes EtherType). The default maximum frame size is therefore 1518 bytes.

Since the maximum frame size should ideally be matched across all endpoints in the same LAN (VLAN), and since the LAN switch infrastructure must be able to forward such packets, the decision to deploy a larger than default maximum frame size requires careful consideration. It is recommended that experimentation with maximum frame size be done in a controlled test environment.

The maximum frame size is changed by changing the Max Frame Size setting in the Network Adapter’s Advanced Properties Page.

Interrupt Moderation (Interrupt Coalescing)

Interrupt moderation reduces the number of interrupts generated by the adapter by coalescing multiple received packet events and/or transmit completion events together into a single interrupt.

The *interrupt moderation interval* sets the minimum time (in microseconds) between two consecutive interrupts. Coalescing occurs only during this interval:

- When the driver generates an interrupt, it starts timing the moderation interval.
- Any events that occur before the moderation interval expires are coalesced together into a single interrupt, that is raised only when the interval expires.

  A new moderation interval then starts, during which no interrupt is raised.
- An event that occurs after the moderation interval has expired gets its own dedicated interrupt, that is raised immediately.

  A new moderation interval then starts, during which no interrupt is raised.

Solarflare adapters, by default, use an *adaptive algorithm* where the interrupt moderation delay is automatically adjusted between zero (no interrupt moderation) and 60 microseconds. The adaptive algorithm detects latency sensitive traffic patterns and adjusts the interrupt moderation interval accordingly.

Interrupt moderation settings are **critical for tuning adapter latency**:

- Disabling the adaptive algorithm will:
  - reduce jitter
  - allow setting the moderation interval as required to suit conditions.
- Increasing the interrupt moderation interval will:
  - generate less interrupts
  - reduce CPU utilization (because there are less interrupts to process)
  - increase latency
  - improve peak throughput.
- Decreasing the interrupt moderation interval will:
  - generate more interrupts
  - increase CPU utilization (because there are more interrupts to process)
  - decrease latency
  - reduce peak throughput.
- Turning off interrupt moderation will:
  - generate the most interrupts
  - give the highest CPU utilization
  - give the lowest latency
  - give the biggest reduction in peak throughput.

For many transaction request-response type network applications, the benefit of reduced latency to overall application performance can be considerable. Such benefits typically outweigh the cost of increased CPU utilization. It is recommended that:

- Interrupt moderation is disabled for applications that require best latency and jitter performance, such as market data handling.
- Interrupt moderation is enabled for high throughput single (or few) connection TCP streaming applications, such as iSCSI.

Interrupt moderation can be disabled or enabled using the Interrupt Moderation setting in the Network Adapter's Advanced Properties Page. The interrupt moderation time value can also be configured from the Network Adapter's Advanced Properties Page.

NOTE: The performance benefits of Receive Segment Coalescing and Large Receive Offload are limited if interrupt moderation is disabled. See Receive Segment Coalescing (RSC) on page 229 and Large Receive Offload (LRO) on page 230.

TCP Checksum Offload

Checksum offload moves calculation and verification of IP Header, TCP and UDP packet checksums to the adapter. The driver has all checksum offload features enabled by default. Therefore, there is no opportunity to improve performance from the default.

Checksum offload configuration is changed by changing the Offload IP Checksum, Offload UDP Checksum and Offload TCP Checksum settings in the Network Adapter's Advanced Properties Page.

- To check that task offloads are enabled (e.g. checksum offload in particular), run the following commands:

  ```
  netsh interface tcp show global
  netsh interface ipv4 show offload
  netsh interface ipv6 show offload
  ```

- See https://msdn.microsoft.com/en-us/library/windows/hardware/ff571012%28v=vs.85%29.aspx for further relevant settings.

NOTE: Solarflare recommend you do not disable checksum offload.

Large Send Offload V2 (LSO)

Large Send Offload (LSO; also known as TCP Segmentation Offload/TSO) offloads the splitting of outgoing TCP data into packets to the adapter. LSO benefits applications using TCP. Applications using protocols other than TCP will not be affected by LSO.

Enabling LSO will reduce CPU utilization on the transmit side of a TCP connection and improve peak throughput, if the CPU is fully utilized. Since LSO has no effect on latency, it can be enabled at all times. The driver has LSO enabled by default. Therefore, there is no opportunity to improve performance from the default.

LSO is changed by changing the Large Send Offload setting in the Network Adapter's Advanced Properties Page.

TCP and IP checksum offloads must be enabled for LSO to work.

NOTE: Solarflare recommend that you do not disable this setting.

Receive Segment Coalescing (RSC)

TCP Receive Segment Coalescing (RSC) is a feature whereby the adapter coalesces multiple packets received on a TCP connection into a single larger packet before passing this onto the network stack for receive processing. This reduces CPU utilization and improves peak throughput when the CPU is fully utilized. The effectiveness of RSC is bounded by the interrupt moderation delay, and is limited if interrupt moderation is disabled (see Interrupt Moderation (Interrupt Coalescing) on page 227). Enabling RSC does not itself negatively impact latency.

RSC is a Microsoft feature introduced in Windows Server 2012. RSC is enabled by default. If a host is forwarding received packets from one interface to another then Windows will automatically disable RSC. For more information about RSC on Windows, see https://technet.microsoft.com/en-gb/library/hh997024.aspx.

The Solarflare network adapter driver enables RSC by default.

RSC is set by changing the Receive Segment Coalescing settings in the Network Adapter's Advanced Properties Page.

TCP and IP checksum offloads must be enabled for RSC to work.

Large Receive Offload (LRO)

Large Receive Offload (LRO) is a feature whereby the adapter coalesces multiple packets received on a TCP connection into a single larger packet before passing this onto the network stack for receive processing. This reduces CPU utilization and improves peak throughput when the CPU is fully utilized. The effectiveness of LRO is bounded by the interrupt moderation delay, and is limited if interrupt moderation is disabled (see Interrupt Moderation (Interrupt Coalescing) on page 227). Enabling LRO does not itself negatively impact latency.

LRO is a Solarflare proprietary mechanism similar to the Windows Receive Segment Coalescing feature. Windows Server 2012 and newer use RSC instead of LRO, and do not support LRO. Older Windows versions that do not support RSC may use LRO instead.

The Solarflare network adapter driver disables LRO by default.

NOTE: LRO should **NOT** be enabled when using the host to forward packets from one interface to another. For example, if the host is performing IP routing.

LRO is set by changing the Large Receive Offload settings in the Network Adapter's Advanced Properties Page.

TCP and IP checksum offloads must be enabled for LRO to work.

Receive Side Scaling (RSS)

Receive Side Scaling (RSS) was first supported as part of the scalable networking pack for Windows Server 2003 and has been improved with each subsequent operating system release. RSS is enabled by default and will be used on network adapters that support it. Solarflare recommend that RSS is enabled for best networking performance.

For further information about using RSS on Windows platforms see the Microsoft white paper *"Scalable Networking: Eliminating the Receive Processing Bottleneck—Introducing RSS"*

This is available from:

http://download.microsoft.com/download/5/D/6/5D6EAF2B-7DDF-476B-93DC-7CF0072878E6/NDIS_RSS.doc

Specific RSS parameters can be tuned on a per-adapter basis. For details see the Microsoft white paper *"Networking Deployment Guide: Deploying High-Speed Networking Features"* available from:

http://download.microsoft.com/download/8/E/D/8EDE21BC-0E3B-4E14-AAEA-9E2B03917A09/HSN_Deployment_Guide.doc

Solarflare network adapters optimize RSS settings by default on Windows operating systems and offer a number of RSS interrupt balancing modes via the network adapter's advanced property page in Device Manager and Solarflare's adapter management tools.

The number of RSS queues can be adjusted to suit the workload:

- The number of RSS CPUs is limited by the number of RSS queues. The driver does not target multiple RSS queues to the same CPU. Therefore:
  - It is best to set the maximum number of RSS queues to be equal to the maximum number of RSS CPUs (or the next higher setting if the equal option is unavailable).
  - The number of queues can be reduced in order to isolate CPU cores for application processing.
  - The number of queues can be increased to spread the load over more cores. This will also increase the amount of receive buffering due to a larger number of RX queues.

NOTE: If hyper-threading is enabled, RSS will only select one thread from each CPU core.

- The current number of queues can be get and set using either SAM, or the Network Adapter’s Advanced Properties Page.
- On Windows Server 2012 and later, the current number of RSS queues can also be queried and set from the PowerShell with the `Get-NetAdapterRss`and `Set-NetAdapterRss` cmdlets.

The set of RSS CPUs can also be adjusted:

- For low latency low jitter applications select the NUMA scaling static RSS profile. Set both the maximum number of RSS processors and the number of RSS queues to be equal to the number of CPU cores

  In multi-port scenarios give each port its own set of RSS processors, using the base and max processor settings to restrict RSS to a subset of the CPUs if required.
- For other applications use as few RSS processors as required to cope with the traffic load, leaving other CPUs free for other tasks

Preferred NUMA Node

The adapter driver chooses a subset of the available CPU cores to handle transmit and receive processing. The Preferred NUMA Node setting can be used to constrain the set of CPU cores considered for processing to those on the given NUMA Node.

To force processing onto a particular NUMA Node, change the Preferred NUMA Node setting on the Network Adapter's Advanced Properties Page.

The NUMA distance of the cores used for the RSS queue and the network application influences performance. To check the NUMA distance of each core from the interface:

- Get Coreinfo from Windows Sysinternals (https://live.sysinternals.com). The output includes processor to NUMA node mappings.

  To get the NUMA node local to the interface, run:

  ```
  Get-WmiObject -Namespace root\wmi -Filter "DummyInstance=False"
  EFX_Port | Format-Table -AutoSize Id,Name,PreferredNumaNode
  ```

- On Windows Server 2012 and later, you can instead run the PowerShell `Get-NetAdapterRss` cmdlet and look at lines starting:

  ```
  RssProcessorArray: [Group:Number/NUMA Distance]
  ```

  This gives the NUMA distance of each core to the interface.

For low latency low jitter applications, RSS queues should be mapped to NUMA nodes that are local to the interface:

- On Windows Server 2008 R2, this should happen automatically. RSS profiles are not supported, and the default behavior is equivalent to ClosestProcessor
- On Windows Server 2012 and later, this should happen automatically if you are using one of the following RSS profiles:
  - ClosestProcessor
  - ClosestProcessorStatic

  The PowerShell `Get-NetAdapterRss` cmdlet will give this information.

  The RSS Profile can be set in the Network Adapter's Advanced Properties Page, or with the PowerShell cmdlet `Set-NetAdapterRss`
- It is also possible to restrict the set of cores available to RSS by setting BaseProcessorGroup/BaseProcessorNumber and MaxProcessorGroup/MaxProcessorNumber.
- See also https://msdn.microsoft.com/en-us/library/windows/hardware/ff570864%28v=vs.85%29.aspx.

Likewise, for low latency low jitter applications, the network application should be run on NUMA nodes that are local to the interface:

- The application can set affinity itself with `SetProcessAffinityMask` and `SetThreadAffinityMask`, and then may present these as options to the user.
- Affinity can be set using Sysinternals Process Explorer (`procexp`). Right click on the process and choose **Set Affinity**.
- Affinity can be controlled when a process is started using:

  `START /AFFINITY <hexmaskl> <command>`

  or

  `START /NODE <num> <command>`

  When `/NODE` and `/AFFINITY` are used together, affinity is interpreted as cores within the node.

For other applications, such as web servers:

- On Windows Server 2012 and later, use one of the following RSS profiles:
  - NumaScaling
  - NumaScalingStatic

  The PowerShell `Get-NetAdapterRss` cmdlet will give this information.

  The RSS Profile can be set in the Network Adapter's Advanced Properties Page, or with the PowerShell cmdlet `Set-NetAdapterRss`

Other Considerations

PCI Express Lane Configurations

The PCI Express (PCIe) interface used to connect the adapter to the server can function at different speeds and widths. This is independent of the physical slot size used to connect the adapter. The possible widths are multiples x1, x2, x4, x8 and x16 lanes of (2.5Gbps for PCIe Gen 1, 5.0 Gbps for PCIe Gen 2 and 8.0Gbps for PCIe Gen 3) in each direction. *Solarflare adapters are designed for x8 or x16 lane operation*.

On some server motherboards, choice of PCIe slot is important. This is because some slots (including those that are physically x8 or x16 lanes) may only electrically support x4 lanes. In x4 lane slots, Solarflare PCIe adapters will continue to operate, but not at full speed. The Solarflare driver will insert a warning in the Windows Event Log if it detects that the adapter is plugged into a PCIe slot which electrically has fewer than x8 lanes.

SFN5xxx and SFN6xxx Solarflare adapters require a PCIe Gen 2 x8 slot for optimal operation. Solarflare SFN7xxx series adapters require a PCIe Gen 3 x8 or x16 slot for optimal performance. The Solarflare driver will insert a warning in the Windows Event Log if it detects that the adapter is placed in a sub-optimal slot.

In addition, the latency of communications between the host CPUs, system memory and the Solarflare PCIe adapter may be PCIe slot dependent. Some slots may be “closer” to the CPU, and therefore have lower latency and higher throughput:

- If possible, install the adapter in a slot which is local to the desired NUMA node
- For Windows Server 2008 R2 best performance will only be obtained if the adapter is closest to NUMA node 0 (i.e. physical CPU package 0).
- Some Windows SKUs only support a single CPU package. If the adapter is plugged into a PCIe slot attached to the second package it will not appear to the OS.

Please consult your server user guide for more information.

Memory bandwidth

Many chipsets use multiple channels to access main system memory. Maximum memory performance is only achieved when the chipset can make use of all channels simultaneously. This should be taken into account when selecting the number of memory modules (DIMMs) to populate in the server. For optimal memory bandwidth in the system, it is likely that:

- all DIMM slots should be populated
- all NUMA nodes should have memory installed.

Please consult the motherboard documentation for details.

BIOS Settings

DELL Systems

Refer to the BIOS configuration guidelines recommended by Dell's white paper *“Configuring Low-Latency Environments on Dell PowerEdge Servers”* available from:

http://i.dell.com/sites/content/shared-content/data-sheets/en/Documents/configuring-low-latency-environments-on-dell-poweredge-12g-servers.pdf

HP Systems

Refer to the BIOS configuration guidelines recommended by HP’s white paper *“Configuring the HP ProLiant Server BIOS for Low-Latency Applications”* available from:

http://h20000.www2.hp.com/bc/docs/support/SupportManual/c01804533/c01804533.pdf

Although targeted at tuning for real-time operating systems, the recommendations equally apply to Windows Server platforms.

Other system vendors may publish similar recommendations. In general any BIOS settings guidelines that are targeted at increasing network performance whilst minimizing latency and jitter are applicable to all operating systems.

Intel® QuickData / NetDMA

On systems that support Intel I/OAT (I/O Acceleration Technology) features such as QuickData (a.k.a NetDMA), Solarflare recommend that these are enabled as they are rarely detrimental to performance.

Using Intel® QuickData Technology allows data copies to be performed by the system and not the operating system. This enables data to move more efficiently through the server and provide fast, scalable, and reliable throughput.

To enable NetDMA the `EnableTCPA` variable must be set to 1 in the `Tcpip\Parameters` registry key. Locate the following key in the registry:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters
```

The `EnableTCPA` value must be created if it is not present and set to 1:

```
EnableTCPA = 1
```

Intel Hyper-Threading Technology

On systems that support Intel Hyper-Threading Technology users should consider benchmarking or application performance data when deciding whether to adopt hyper-threading on a particular system and for a particular application. Solarflare have identified that hyper-threading is generally beneficial on systems fitted with Core i5, Core i7 and Xeon (Nehalem or later) CPUs.

TCP/IP Options

TCP timestamps, window scaling and selective acknowledgments are enabled by default on supported platforms, and include receive window tuning and congestion control algorithms that automatically adapt to 10 gigabit connections. There is therefore no need to change these settings.

Power Saving Mode

Modern processors utilize design features that enable a CPU core to drop into low power states when instructed by the operating system that the CPU core is idle. When the OS schedules work on the idle CPU core (or when other CPU cores or devices need to access data currently in the idle CPU core's data cache) the CPU core is signaled to return to the fully on power state. These changes in CPU core power states create additional network latency and jitter. Solarflare recommend to achieve the lowest latency and lowest jitter that the "C1E power state" or "CPU power saving mode" is disabled within the system BIOS.

In general the user should examine the system BIOS settings and identify settings that favor performance over power saving. In particular look for settings to disable:

- C states / Processor sleep/idle states
- Enhanced C1 CPU sleep state (C1E)
- Any deeper C states (C3 through to C6)
- P states / Processor throttling

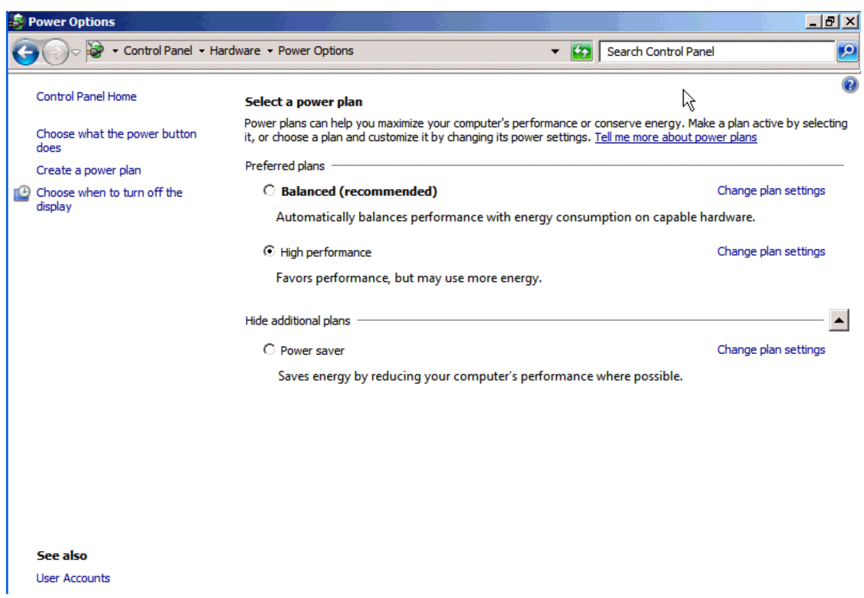
- Processor Turbo mode
- Ultra Low Power State
- PCIe Active State Power Management (ASPM)
- Unnecessary SMM/SMI features

The latency can be improved by selecting the highest performance power plan:

- Consider using the **Optimum performance** power plan added by the Solarflare driver package installer, which:
  - Disables all CPU idle states
  - Sets the OS device idle policy to favor performance over power savings
  - Disables PCIe ASPM (Active State Power Management)

  The installer does not enable this by default as, on certain systems, it might significantly increase energy usage.
- Otherwise, select the Always On or High performance power plan.

The power plan is configured from the Control Panel > Hardware > Power Options:



You can also use the `powercfg.exe` utility that is installed with Windows:

- List all power schemes in the current user's environment:

  ```
  powercfg /LIST
  ```
- Make the specified power scheme active on the computer:

  ```
  powercfg /SETACTIVE <GUID>
  ```

See https://technet.microsoft.com/en-us/library/cc748940%28v=ws.10%29.aspx.

Firewalls and anti-virus software

Depending on the system configuration, the following software may have a significant impact on throughput and CPU utilization, in particular when receiving multicast UDP traffic:

- the built-in Windows Firewall and Base Filtering Engine
- other third-party firewall or network security products
- anti-virus checkers.

This is the case even if the software has no rules configured but is still active.

Where high throughput is required on a particular port, the performance will be improved by disabling the software on that port:

NOTE: The Windows (or any third party) Firewall should be disabled with caution. The network administrator should be consulted before making any changes.

- Disable the Windows Firewall.
  - ***a)*** Set the **Startup Type** of the *Windows Firewall* service to **Disabled**.
  - ***b)*** Stop the service.
- On Windows Server 2008 R2 disable the Windows Filtering Platform (WFP).
  - ***a)*** Set the **Startup Type** of the *Base Filtering Engine* (BFE) service to **Disabled**.
  - ***b)*** Stop the service.

  For more information, see https://msdn.microsoft.com/en-us/Library/Windows/Hardware/dn653358%28v=vs.85%29.aspx.
- Disable (and if possible uninstall) any third-party firewall or network security products, and any anti-virus checkers.

  Remember that even the simplest anti-virus products can inspect network traffic to look for viruses in e-mails.

Configure network applications

Consider the options, logging and statistics provided by the networking applications that are being used.

Benchmarks

Throughput Benchmark using Ntttcp

The following example shows results from running Microsoft's ntttcp. It is suggested that first, **Large Receive Offload** (LRO) or **Receive Segment Coalescing** (RSC) is enabled.

1 On server run ntttcpr:

```
ntttcpr.exe -rb 500000 -a 24 -n 100000 -l 524288 -m
1,1,<server_adapter_IP_interface>
```

2 On client run ntttcps test:

```
ntttcps.exe -rb 500000 -a 24 -n 100000 -l 524288 -m
1,1,<server_adapter_IP_interface>
```

```
C:\ > ntttcps.exe -rb 500000 -a 24 -n 100000 -l 524288 -m 1,1,<server adapter IP
interface>
Copyright Version 2.4
Network activity progressing...
Thread Realtime(s) Throughput(KB/s) Throughput(Mbit/s)
====== =========== ================
0 44.767 1170961.007 9367.688
Total Bytes(MEG) Realtime(s) Average Frame Size Total Throughput(Mbit/s)
================ =========== ==================
52420.411392 44.767 1459.846 9367.688
Total Buffers Throughput(Buffers/s) Pkts(sent/intr) Intr(count/s) Cycles/Byte
============= =====================
99984.000 2233.431 27 29187.48 0.8
Packets Sent Packets Received Total Retransmits Total Errors Avg. CPU %
============ ================ =================
```

Tuning Recommendations

The following tables provide recommendations for tuning settings for different application characteristics.

- Throughput - Table 58 on page 239
- Latency - Table 59 on page 240

Table 58: Throughput Tuning Settings

Tuning Parameter	How?
Intel QuickData (Intel chipsets only)	Enable in BIOS and configure as described in guide.
Interrupt Moderation	Leave at default (Enabled).
Adaptive Interrupt Moderation	Leave at default (Enabled).
Interrupt Moderation Time	Leave at default (60µs).
Large Receive Offloads	**Enable** in Network Adapter Advanced Properties.
Large Send Offloads	Leave at default (Enabled).
Max Frame Size	Configure to maximum supported by network in Network Adapter's Advanced Properties.
Memory bandwidth	Ensure memory utilizes all memory channels on system motherboard.
Offload Checksums	Leave at default.
PCI Express Lane Configuration	Ensure the adapter is in an x8 slot (2.0 or later), and that current speed (not the supported speed) reads back as “x8 and 5GT/s”, or “x8 and 8GT/s”, or “x8 and Unknown”.
Power Saving Mode	Leave at default.
Receive Segment Coalescing	Leave at default (Enabled).
Receive Side Scaling (RSS)	Leave at default.
RSS NUMA Node	Leave at default (All).
TCP Protocol Tuning	Leave at default (install with “Optimize Windows TCP/IP protocol settings for 10G networking” option selected).

Table 59: Latency Tuning Settings

Tuning Parameter	How?
Intel QuickData (Intel chipsets only)	Enable in BIOS and configure as described in guide.
Interrupt Moderation	**Disable** in Network Adapter’s Advanced Properties.
Adaptive Interrupt Moderation	Leave at default. This setting is ignored when interrupt moderation is disabled.
Interrupt Moderation Time	Leave at default (60µs). This setting is ignored when interrupt moderation is disabled.
Large Receive Offloads	Disable in Network Adapter’s Advanced Properties.
Large Send Offloads	Leave at default (Enabled).
Max Frame Size	Configure to maximum supported by network in Network Adapter's Advanced Properties.
Memory bandwidth	Ensure memory utilizes all memory channels on system motherboard.
Offload Checksums	Leave at default (Enabled).
PCI Express Lane Configuration	Ensure the adapter is in an x8 slot (2.0 or later), and that current speed (not the supported speed) reads back as “x8 and 5GT/s”, or “x8 and 8GT/s”, or “x8 and Unknown”.
Power Saving Mode	Disable C1E and other CPU sleep modes to prevent OS from putting CPUs into lowering power modes when idle.
Receive Segment Coalescing	Disable in Network Adapter’s Advanced Properties.
Receive Side Scaling	Application dependent
RSS NUMA Node	Leave at default (All).
TCP Protocol Tuning	Leave at default (install with “Optimize Windows TCP/IP protocol settings for 10G networking” option selected).
TCP/IP Checksum Offload	Leave at default

4.28 Windows Event Log Error Messages

The following tables list the various error messages that can be added to the event log, along with a description and action that should be taken.

Driver Status Codes

Table 60: Driver Status Codes

Value	Error Message	Description	Severity	Notes
0x60000001L	BUS_STATUS_DRIVER_VERSION	The driver version information.	Informational	No action required.
0x60000002L	BUS_STATUS_DRIVER_LOAD_FAILURE	The driver failed to load.	Informational	
0xA0000004L	BUS_STATUS_DRIVER_NOT_ADDING_DEVICE	The driver can't add a device due to the system being started in safe mode (SAFEMODE_MINIMAL).	Warning	
0xA0000005L	BUS_STATUS_DRIVER_NUMA_ALLOCATION_FAILED	The driver could not allocate memory on a specific NUMA node.	Warning	For maximum performance all NUMA nodes should be populated. Install additional memory.

Device Status Codes

Table 61: Device Status Codes

Value	Error Message	Description	Severity	Action
0x6001000BL	BUS_STATUS_DEVICE_MTU_CHANGE	The MTU on the device was changed.	Informational	No action required.
0x6001001BL	BUS_STATUS_DEVICE_MCDI_VERSION	Hardware MCDI version	Informational	None required.

Table 61: Device Status Codes

Value	Error Message	Description	Severity	Action
0xA0010004L	BUS_STATUS_DEVICE_ LINK_WIDTH	The device does not have sufficient PCIe lanes to reach full bandwidth.	Warning	Move the adapter into a PCIe slot with more lanes. See PCI Express Lane Configurations on page 233.
0xA001000CL	BUS_STATUS_DEVICE_ TX_WATCHDOG	The transmit watchdog fired	Warning	
0xA001000DL	BUS_STATUS_DEVICE_ UNEXPECTED_EVENT	An unexpected event was received from the device.	Warning	No action required.
0xA0010010L	BUS_STATUS_DEVICE_ WRONG_RX_EVENT	A non-contiguous RX event was received from the device.	Warning	
0xA0010011L	BUS_STATUS_DEVICE_ TEMPERATURE_ WARNING	The device has exceeded the maximum supported temperature limit.	Warning	Improve the server cooling.
0xA0010013L	BUS_STATUS_DEVICE_ COOLING_ERROR	The device cooling has failed.	Warning	
0xA0010014L	BUS_STATUS_DEVICE_ VOLTAGE_WARNING	One of the device voltage supplies is outside of the supported voltage range.	Warning	The adapter or server maybe faulty.
0xA0010017L	BUS_STATUS_DEVICE_ MCDI_ERR	Hardware MCDI communication suffered an error.	Warning	None required.
0xA0010019L	BUS_STATUS_DEVICE_ MCDI_BOOT_ERROR	Hardware MCDI boot from non-primary flash. Possible flash corruption.	Warning	Run sfupdate or update via SAM.
0xE0010002L	BUS_STATUS_DEVICE_ PHY_ZOMBIE	PHY firmware has failed to start.	Error	Possible PHY firmware corruption. Run sfupdate or SAM to update.

Table 61: Device Status Codes

Value	Error Message	Description	Severity	Action
0xE0010005L	BUS_STATUS_DEVICE_ ADD_FAILURE	The device could not be added to the system.	Error	
0xE0010006L	BUS_STATUS_DEVICE_ INIT_INTERRUPTS_ DISABLED_FAILURE	The device could not be initialized with interrupts disabled.	Error	
0xE0010007L	BUS_STATUS_DEVICE_ INIT_INTERRUPTS_ ENABLED_FAILURE	The device could not be initialized with interrupts enabled.	Error	
0xE0010008L	BUS_STATUS_DEVICE_ START_FAILURE	The device could not be started.	Error	
0xE0010009L	BUS_STATUS_DEVICE_ RESET_FAILURE	The device could not be reset.	Error	
0xE001000AL	BUS_STATUS_DEVICE_ EFX_FAILURE	There was an EFX API failure.	Error	
0xE0010012L	BUS_STATUS_DEVICE_ TEMPERATURE_ERROR	The device has exceeded the critical temperature limit.	Error	Improve the server cooling.
0xE0010015L	BUS_STATUS_DEVICE_ VOLTAGE_ERROR	One of the device voltage supplies is outside of the critical voltage range.	Error	The adapter or server maybe faulty.
0xE0010016L	BUS_STATUS_DEVICE_ UNKNOWN_ SENSOREVT	A non-specified hardware monitor device has reported an error condition.	Error	
0xE0010018L	BUS_STATUS_DEVICE_ MCDI_TIMEOUT	Hardware MCDI communication timed out.	Error	None required.

NDIS Driver Status Codes

Table 62: NDIS Driver Status Codes

Value	Error Message	Description	Severity	Notes
0x60030001L	NDIS6_EVENT_ FAULT_RX_RESET_ OCCURRED	The driver has encountered a network adapter receive data path error and performed a network adapter reset to attempt recovery. Refer to the driver release notes and errata for more information.	Informational	
0xA0010001L	NDIS6_EVENT_ PARAMETER_ BREAKONENTRY_ MISSING	The parameter BreakOnEntry is missing from the registry. Using default value.	Warning	
0xA0010002L	NDIS6_EVENT_ PARAMETER_ BREAKONENTRY_ OUT_OF_RANGE	The parameter BreakOnEntry specified in the registry is out of range. Using default value.	Warning	
0xA0010003L	NDIS6_EVENT_ PARAMETER_ DEBUGBITS_ MISSING	The parameter DebugBits is missing from the registry. Using default value.	Warning	
0xA0010004L	NDIS6_EVENT_ PARAMETER_ DEBUGBITS_ OUT_OF_RANGE	The parameter DebugBits specified in the registry is out of range. Using default value.	Warning	
0xA0010005L	NDIS6_EVENT_ PARAMETER_ FLOWCONTROL_ MISSING	The parameter *FlowControl is missing from the registry. Using default value.	Warning	

Table 62: NDIS Driver Status Codes

Value	Error Message	Description	Severity	Notes
0xA0010006L	NDIS6_EVENT_ PARAMETER_ FLOWCONTROL_ OUT_OF_RANGE	The parameter *FlowControl specified in the registry is out of range. Using default value.	Warning	
0xA0010007L	NDIS6_EVENT_ PARAMETER_ IFTYPE_ MISSING	The parameter *IfType is missing from the registry. Using default value.	Warning	
0xA0010008L	NDIS6_EVENT_ PARAMETER_ IFTYPE_ OUT_OF_RANGE	The parameter *IfType specified in the registry is out of range. Using default value.	Warning	
0xA0010009L	NDIS6_EVENT_ PARAMETER_ INTERRUPT MODERATION_ MISSING	The parameter *InterruptModeration is missing from the registry. Using default value.	Warning	
0xA001000AL	NDIS6_EVENT_ PARAMETER_ INTERRUPT MODERATION_ OUT_OF_RANGE	The parameter *InterruptModeration specified in the registry is out of range. Using default value.	Warning	
0xA001000BL	NDIS6_EVENT_ PARAMETER_ IPCHKSUMOFFV4_ MISSING	The parameter *IPChksumOffv4 is missing from the registry. Using default value.	Warning	
0xA001000CL	NDIS6_EVENT_ PARAMETER_ IPCHKSUMOFFV4_ OUT_OF_RANGE	The parameter *IPChksumOffv4 specified in the registry is out of range. Using default value.	Warning	
0xA001000DL	NDIS6_EVENT_ PARAMETER_ JUMBOPACKET_ MISSING	The parameter *JumboPacket is missing from the registry. Using default value.	Warning	

Table 62: NDIS Driver Status Codes

Value	Error Message	Description	Severity	Notes
0xA001000EL	NDIS6_EVENT_ PARAMETER_ JUMBOPACKET_ OUT_OF_RANGE	The parameter *JumboPacket specified in the registry is out of range. Using default value.	Warning	
0xA001000FL	NDIS6_EVENT_ PARAMETER_ LROIPV4_ MISSING	The parameter LROIPv4 is missing from the registry. Using default value.	Warning	
0xA0010010L	NDIS6_EVENT_ PARAMETER_ LROIPV4_ OUT_OF_RANGE	The parameter LROIPv4 specified in the registry is out of range. Using default value.	Warning	
0xA0010011L	NDIS6_EVENT_ PARAMETER_ LSOV2IPV4_ MISSING	The parameter *LSOv2IPv4 is missing from the registry. Using default value.	Warning	
0xA0010012L	NDIS6_EVENT_ PARAMETER_ LSOV2IPV4_ OUT_OF_RANGE	The parameter *LSOv2IPv4 specified in the registry is out of range. Using default value.	Warning	
0xA0010013L	NDIS6_EVENT_ PARAMETER_ MAXEVENTSPER RECEIVEINTERRUPT_ MISSING	The parameter MaxEventsPerReceive Interrupt is missing from the registry. Using default value.	Warning	
0xA0010014L	NDIS6_EVENT_ PARAMETER_ MAXEVENTSPER RECEIVEINTERRUPT_ OUT_OF_RANGE	The parameter MaxEventsPerReceive Interrupt specified in the registry is out of range. Using default value.	Warning	
0xA0010015L	NDIS6_EVENT_ PARAMETER_ MEDIATYPE_ MISSING	The parameter *MediaType is missing from the registry. Using default value.	Warning	

Table 62: NDIS Driver Status Codes

Value	Error Message	Description	Severity	Notes
0xA0010016L	NDIS6_EVENT_ PARAMETER_ MEDIATYPE_ OUT_OF_RANGE	The parameter *MediaType specified in the registry is out of range. Using default value.	Warning	
0xA0010017L	NDIS6_EVENT_ PARAMETER_ NETWORKADDRESS_ MISSING	The parameter *NetworkAddress is missing from the registry. Using default value.	Warning	
0xA0010018L	NDIS6_EVENT_ PARAMETER_ NETWORKADDRESS_ OUT_OF_RANGE	The parameter *NetworkAddress specified in the registry is invalid. Using default value.	Warning	
0xA0010019L	NDIS6_EVENT_ PARAMETER_ PHYSICALMEDIATYPE_ MISSING	The parameter *PhysicalMediaType is missing from the registry. Using default value.	Warning	
0xA001001AL	NDIS6_EVENT_ PARAMETER_ PHYSICALMEDIATYPE_ OUT_OF_RANGE	The parameter *PhysicalMediaType specified in the registry is out of range. Using default value.	Warning	
0xA001001BL	NDIS6_EVENT_ PARAMETER_ PRIORITYVLANTAG_ MISSING	The parameter *PriorityVLANTag is missing from the registry. Using default value.	Warning	
0xA001001CL	NDIS6_EVENT_ PARAMETER_ PRIORITYVLANTAG_ OUT_OF_RANGE	The parameter *PriorityVLANTag specified in the registry is out of range. Using default value.	Warning	
0xA001001DL	NDIS6_EVENT_ PARAMETER_ RECEIVEBUFFERS_ MISSING	The parameter *ReceiveBuffers is missing from the registry. Using default value.	Warning	

Table 62: NDIS Driver Status Codes

Value	Error Message	Description	Severity	Notes
0xA001001EL	NDIS6_EVENT_ PARAMETER_ RECEIVEBUFFERS_ OUT_OF_RANGE	The parameter *ReceiveBuffers specified in the registry is out of range. Using default value.	Warning	
0xA001001FL	NDIS6_EVENT_ PARAMETER_ RECEIVEINTERRUPT MODERATIONVALUE_ MISSING	The parameter ReceiveInterrupt ModerationValue is missing from the registry. Using default value.	Warning	
0xA0010020L	NDIS6_EVENT_ PARAMETER_ RECEIVEINTERRUPT MODERATIONVALUE_ OUT_OF_RANGE	The parameter ReceiveInterrupt ModerationValue specified in the registry is out of range. Using default value.	Warning	
0xA0010021L	NDIS6_EVENT_ PARAMETER_ SPEEDDUPLEX_ MISSING	The parameter *SpeedDuplex is missing from the registry. Using default value.	Warning	
0xA0010022L	NDIS6_EVENT_ PARAMETER_ SPEEDDUPLEX_ OUT_OF_RANGE	The parameter *SpeedDuplex specified in the registry is out of range. Using default value.	Warning	
0xA0010023L	NDIS6_EVENT_ PARAMETER_ TCPCHKSUMOFFV4_ MISSING	The parameter *TCPChksumOffv4 is missing from the registry. Using default value.	Warning	
0xA0010024L	NDIS6_EVENT_ PARAMETER_ TCPCHKSUMOFFV4_ OUT_OF_RANGE	The parameter *TCPChksumOffv4 specified in the registry is out of range. Using default value.	Warning	

Table 62: NDIS Driver Status Codes

Value	Error Message	Description	Severity	Notes
0xA0010025L	NDIS6_EVENT_ PARAMETER_ TRANSMITBUFFERS_ MISSING	The parameter *TransmitBuffers is missing from the registry. Using default value.	Warning	
0xA0010026L	NDIS6_EVENT_ PARAMETER_ TRANSMITBUFFERS_ OUT_OF_RANGE	The parameter *TransmitBuffers specified in the registry is out of range. Using default value.	Warning	
0xA0010027L	NDIS6_EVENT_ PARAMETER_ UDPCHKSUMOFFV4_ MISSING	The parameter *UDPChksumOffv4 is missing from the registry. Using default value.	Warning	
0xA0010028L	NDIS6_EVENT_ PARAMETER_ UDPCHKSUMOFFV4_ OUT_OF_RANGE	The parameter *UDPChksumOffv4 specified in the registry is out of range. Using default value.	Warning	
0xA0010029L	NDIS6_EVENT_ PARAMETER_ VLANID_ MISSING	The parameter VlanId is missing from the registry. Using default value.	Warning	
0xA001002AL	NDIS6_EVENT_ PARAMETER_ VLANID_ OUT_OF_RANGE	The parameter VlanId specified in the registry is out of range. Using default value.	Warning	
0xA001002BL	NDIS6_EVENT_ PARAMETER_ WARNLEVEL_ MISSING	The parameter WarnLevel is missing from the registry. Using default value.	Warning	
0xA001002CL	NDIS6_EVENT_ PARAMETER_ WARNLEVEL_ OUT_OF_RANGE	The parameter WarnLevel specified in the registry is out of range. Using default value.	Warning	

Table 62: NDIS Driver Status Codes

Value	Error Message	Description	Severity	Notes
0xA001002DL	NDIS6_EVENT_ PARAMETER_ RSS_ MISSING	The parameter *RSS is missing from the registry. Using default value.	Warning	
0xA001002EL	NDIS6_EVENT_ PARAMETER_ RSS_ OUT_OF_RANGE	The parameter *RSS specified in the registry is invalid. Using default value.	Warning	
0xA001002FL	NDIS6_EVENT_ PARAMETER_ IPCHECKSUM OFFLOADIPV4_ MISSING	The parameter *IPChecksumOffload IPv4 is missing from the registry. Using default value.	Warning	
0xA0010030L	NDIS6_EVENT_ PARAMETER_ IPCHECKSUM OFFLOADIPV4_ OUT_OF_RANGE	The parameter *IPChecksumOffload IPv4 specified in the registry is out of range. Using default value.	Warning	
0xA0010031L	NDIS6_EVENT_ PARAMETER_ TCPCHECKSUM OFFLOADIPV4_ MISSING	The parameter *TCPChecksumOffload IPv4 is missing from the registry. Using default value.	Warning	
0xA0010032L	NDIS6_EVENT_ PARAMETER_ TCPCHECKSUM OFFLOADIPV4_ OUT_OF_RANGE	The parameter *TCPChecksumOffload IPv4 specified in the registry is out of range. Using default value.	Warning	
0xA0010033L	NDIS6_EVENT_ PARAMETER_ UDPCHECKSUM OFFLOADIPV4_ MISSING	The parameter *UDPChecksum OffloadIPv4 is missing from the registry. Using default value.	Warning	
0xA0010034L	NDIS6_EVENT_ PARAMETER_ UDPCHECKSUM OFFLOADIPV4_ OUT_OF_RANGE	The parameter *UDPChecksum OffloadIPv4 specified in the registry is out of range. Using default value.	Warning	

Table 62: NDIS Driver Status Codes

Value	Error Message	Description	Severity	Notes
0xA0010035L	NDIS6_EVENT_ PARAMETER_ TCPCONNECTION OFFLOADIPV4_ MISSING	The parameter *TCPConnection OffloadIPv4 is missing from the registry. Using default value.	Warning	
0xA0010036L	NDIS6_EVENT_ PARAMETER_ TCPCONNECTION OFFLOADIPV4_ OUT_OF_RANGE	The parameter *TCPConnection OffloadIPv4 specified in the registry is out of range. Using default value.	Warning	
0xA0010037L	NDIS6_EVENT_ PARAMETER_ MAXRSSPROCESSORS_ MISSING	The parameter *MaxRssProcessors is missing from the registry. Using default value.	Warning	
0xA0010038L	NDIS6_EVENT_ PARAMETER_ MAXRSSPROCESSORS_ OUT_OF_RANGE	The parameter *MaxRssProcessors specified in the registry is invalid. Using default value.	Warning	
0xA0010039L	NDIS6_EVENT_ PARAMETER_ INTERRUPT MODERATIONTIME_ MISSING	The parameter InterruptModeration Time is missing from the registry. Using default value.	Warning	
0xA001003AL	NDIS6_EVENT_ PARAMETER_ INTERRUPT MODERATIONTIME_ OUT_OF_RANGE	The parameter InterruptModeration Time specified in the registry is invalid. Using default value.	Warning	
0xA001003BL	NDIS6_EVENT_ PARAMETER_ TCPCHECKSUM OFFLOADIPV6_ MISSING	The parameter *TCPChecksum OffloadIPv6 is missing from the registry. Using default value.	Warning	

Table 62: NDIS Driver Status Codes

Value	Error Message	Description	Severity	Notes
0xA001003CL	NDIS6_EVENT_ PARAMETER_ TCPCHECKSUM OFFLOADIPV6_ OUT_OF_RANGE	The parameter *TCPChecksumOffload IPv6 specified in the registry is out of range. Using default value.	Warning	
0xA001003DL	NDIS6_EVENT_ PARAMETER_ UDPCHECKSUM OFFLOADIPV6_ MISSING	The parameter *UDPChecksum OffloadIPv6 is missing from the registry. Using default value.	Warning	
0xA001003EL	NDIS6_EVENT_ PARAMETER_ UDPCHECKSUM OFFLOADIPV6_ OUT_OF_RANGE	The parameter *UDPChecksum OffloadIPv6 specified in the registry is out of range. Using default value.	Warning	
0xA001003FL	NDIS6_EVENT_ PARAMETER_ WAKEONPATTERN_ MISSING	The parameter *WakeOnPattern is missing from the registry. Using default value.	Warning	
0xA0010040L	NDIS6_EVENT_ PARAMETER_ WAKEONPATTERN_ OUT_OF_RANGE	The parameter *WakeOnPattern specified in the registry is out of range. Using default value.	Warning	
0xA0010041L	NDIS6_EVENT_ PARAMETER_ WAKEON MAGICPACKET_ MISSING	The parameter *WakeOnMagicPacket is missing from the registry. Using default value.	Warning	
0xA0010042L	NDIS6_EVENT_ PARAMETER_ WAKEON MAGICPACKET_ OUT_OF_RANGE	The parameter *WakeOnMagicPacket specified in the registry is out of range. Using default value.	Warning	

Table 62: NDIS Driver Status Codes

Value	Error Message	Description	Severity	Notes
0xA0010043L	NDIS6_EVENT_ PARAMETER_ DEVICESLEEP ONDISCONNECT_ MISSING	The parameter *DeviceSleepOn Disconnect is missing from the registry. Using default value.	Warning	
0xA0010044L	NDIS6_EVENT_ PARAMETER_ DEVICESLEEP ONDISCONNECT_ OUT_OF_RANGE	The parameter *DeviceSleepOn Disconnect specified in the registry is out of range. Using default value.	Warning	
0xA0010045L	NDIS6_EVENT_ PARAMETER_ PMARPOFFLOAD_ MISSING	The parameter *PMARPOffload is missing from the registry. Using default value.	Warning	
0xA0010046L	NDIS6_EVENT_ PARAMETER_ PMARPOFFLOAD_ OUT_OF_RANGE	The parameter *PMARPOffload specified in the registry is out of range. Using default value.	Warning	
0xA0010047L	NDIS6_EVENT_ PARAMETER_ PMNSOFFLOAD_ MISSING	The parameter *PMNSOffload is missing from the registry. Using default value.	Warning	
0xA0010048L	NDIS6_EVENT_ PARAMETER_ PMNSOFFLOAD_ OUT_OF_RANGE	The parameter *PMNSOffload specified in the registry is out of range. Using default value.	Warning	
0xA0010049L	NDIS6_EVENT_ PARAMETER_ LSOV2IPV6_ MISSING	The parameter *LSOv2IPv6 is missing from the registry. Using default value.	Warning	
0xA001004AL	NDIS6_EVENT_ PARAMETER_ LSOV2IPV6_ OUT_OF_RANGE	The parameter *LSOv2IPv6 specified in the registry is out of range. Using default value.	Warning	

Table 62: NDIS Driver Status Codes

Value	Error Message	Description	Severity	Notes
0xA001004BL	NDIS6_EVENT_ PARAMETER_ LROIPV6_ MISSING	The parameter LROIPv6 is missing from the registry. Using default value.	Warning	
0xA001004CL	NDIS6_EVENT_ PARAMETER_ LROIPV6_ OUT_OF_RANGE	The parameter LROIPv6 specified in the registry is out of range. Using default value.	Warning	
0xA001004DL	NDIS6_EVENT_ PARAMETER_ FLOWCONTROL AUTONEGOTIATION_ MISSING	The parameter FlowControlAuto Negotiation is missing from the registry. Using default value.	Warning	
0xA001004EL	NDIS6_EVENT_ PARAMETER_ FLOWCONTROL AUTONEGOTIATION_ OUT_OF_RANGE	The parameter FlowControlAuto Negotiation specified in the registry is out of range. Using default value.	Warning	
0xA001004FL	NDIS6_EVENT_ PARAMETER_ ADAPTIVEINTERRUPT MODERATION_ MISSING	The parameter AdaptiveInterrupt Moderation is missing from the registry. Using default value.	Warning	
0xA0010050L	NDIS6_EVENT_ PARAMETER_ ADAPTIVEINTERRUPT MODERATION_ OUT_OF_RANGE	The parameter AdaptiveInterrupt Moderation specified in the registry is out of range. Using default value.	Warning	
0xA0010051L	NDIS6_EVENT_ PARAMETER_ NUMANODEID_ MISSING	The parameter *NumaNodeId is missing from the registry. Using default value.	Warning	

Table 62: NDIS Driver Status Codes

Value	Error Message	Description	Severity	Notes
0xA0010052L	NDIS6_EVENT_ PARAMETER_ NUMANODEID_ OUT_OF_RANGE	The parameter *NumaNodeId specified in the registry is out of range. Using default value.	Warning	
0xA0010053L	NDIS6_EVENT_ PARAMETER_ VMQ_ MISSING	The parameter *VMQ is missing from the registry. Using default value.	Warning	
0xA0010054L	NDIS6_EVENT_ PARAMETER_ VMQ_ OUT_OF_RANGE	The parameter *VMQ specified in the registry is out of range. Using default value.	Warning	
0xA0010055L	NDIS6_EVENT_ PARAMETER_ VMQLOOKAHEAD SPLIT_ MISSING	The parameter *VMQLookaheadSplit is missing from the registry. Using default value.	Warning	
0xA0010056L	NDIS6_EVENT_ PARAMETER_ VMQLOOKAHEAD SPLIT_ OUT_OF_RANGE	The parameter *VMQLookaheadSplit specified in the registry is out of range. Using default value.	Warning	
0xA0010057L	NDIS6_EVENT_ PARAMETER_ VMQVLANFILTERING_ MISSING	The parameter *VMQVlanFiltering is missing from the registry. Using default value.	Warning	
0xA0010058L	NDIS6_EVENT_ PARAMETER_ VMQVLANFILTERING_ OUT_OF_RANGE	The parameter *VMQVlanFiltering specified in the registry is out of range. Using default value.	Warning	
0xA0010059L	NDIS6_EVENT_ PARAMETER_ RSSQUEUECOUNT_ MISSING	The parameter RssQueueCount is missing from the registry. Using default value.	Warning	

Table 62: NDIS Driver Status Codes

Value	Error Message	Description	Severity	Notes
0xA001005AL	NDIS6_EVENT_ PARAMETER_ RSSQUEUECOUNT_ OUT_OF_RANGE	The parameter RssQueueCount specified in the registry is out of range. Using default value.	Warning	
0xA001005BL	NDIS6_EVENT_ PARAMETER_ RSSALGORITHM_ MISSING	The parameter RssAlgorithm is missing from the registry. Using default value.	Warning	
0xA001005CL	NDIS6_EVENT_ PARAMETER_ RSSALGORITHM_ OUT_OF_RANGE	The parameter RssAlgorithm specified in the registry is out of range. Using default value.	Warning	
0xA001005DL	NDIS6_EVENT_ PARAMETER_ NUMRSSQUEUES_ MISSING	The parameter *NumRSSQueues is missing from the registry. Using default value.	Warning	
0xA001005EL	NDIS6_EVENT_ PARAMETER_ NUMRSSQUEUES_ OUT_OF_RANGE	The parameter *NumRSSQueues specified in the registry is out of range. Using default value.	Warning	
0xA001005FL	NDIS6_EVENT_ PARAMETER_ RSSPROFILE_ MISSING	The parameter *RssProfile is missing from the registry. Using default value.	Warning	
0xA0010060L	NDIS6_EVENT_ PARAMETER_ RSSPROFILE_ OUT_OF_RANGE	The parameter *RssProfile specified in the registry is out of range. Using default value.	Warning	
0xA0010061L	NDIS6_EVENT_ PARAMETER_ RSSBASEPROCGROUP_ MISSING	The parameter *RssBaseProcGroup is missing from the registry. Using default value.	Warning	

Table 62: NDIS Driver Status Codes

Value	Error Message	Description	Severity	Notes
0xA0010062L	NDIS6_EVENT_ PARAMETER_ RSSBASEPROCGROUP_ OUT_OF_RANGE	The parameter *RssBaseProcGroup specified in the registry is out of range. Using default value.	Warning	
0xA0010063L	NDIS6_EVENT_ PARAMETER_ RSSBASEPROC NUMBER_ MISSING	The parameter *RssBaseProcNumber is missing from the registry. Using default value.	Warning	
0xA0010064L	NDIS6_EVENT_ PARAMETER_ RSSBASEPROC NUMBER_ OUT_OF_RANGE	The parameter *RssBaseProcNumber specified in the registry is out of range. Using default value.	Warning	
0xA0010065L	NDIS6_EVENT_ PARAMETER_ RSSMAXPROCGROUP_ MISSING	The parameter *RSSMaxProcGroup is missing from the registry. Using default value.	Warning	
0xA0010066L	NDIS6_EVENT_ PARAMETER_ RSSMAXPROCGROUP_ OUT_OF_RANGE	The parameter *RSSMaxProcGroup specified in the registry is out of range. Using default value.	Warning	
0xA0010067L	NDIS6_EVENT_ PARAMETER_ RSSMAXPROC NUMBER_ MISSING	The parameter *RssMaxProcNumber is missing from the registry. Using default value.	Warning	
0xA0010068L	NDIS6_EVENT_ PARAMETER_ RSSMAXPROC NUMBER_ OUT_OF_RANGE	The parameter *RssMaxProcNumber specified in the registry is out of range. Using default value.	Warning	
0xA0010069L	NDIS6_EVENT_ PARAMETER_ RSCIPV4_ MISSING	The parameter *RscIPv4 is missing from the registry. Using default value.	Warning	

Table 62: NDIS Driver Status Codes

Value	Error Message	Description	Severity	Notes
0xA001006AL	NDIS6_EVENT_ PARAMETER_ RSCIPV4_ OUT_OF_RANGE	The parameter *RscIPv4 specified in the registry is out of range. Using default value.	Warning	
0xA001006BL	NDIS6_EVENT_ PARAMETER_ RSCIPV6_ MISSING	The parameter *RscIPv6 is missing from the registry. Using default value.	Warning	
0xA001006CL	NDIS6_EVENT_ PARAMETER_ RSCIPV6_ OUT_OF_RANGE	The parameter *RscIPv6 specified in the registry is out of range. Using default value.	Warning	
0xA0020001L	NDIS6_EVENT_ TX_CHECKSUM_ OFFLOAD_ NOT_SUPPORTED	The driver has determined that the network adapter does not support transmit checksum offloads and therefore has overridden the *IPChksumOffv4, *TCPChksumOffv4 and *UDPChksumOffv4 registry parameters.	Warning	
0xA0020002L	NDIS6_EVENT_ LARGE_SEND_ OFFLOAD_ NOT_SUPPORTED	The driver has determined that the network adapter does not support Large Send Offload and therefore has overridden the *LSOv2IPv4 and *LSOv2IPv6 registry parameters.	Warning	

Table 62: NDIS Driver Status Codes

Value	Error Message	Description	Severity	Notes
0xA0020003L	NDIS6_EVENT_ LARGE_RECEIVE_ OFFLOAD_ INCOMPATIBLE_WITH_ TAGGING	The driver has disabled Large Receive Offload (overriding the LROIPv4 registry parameter) as it is incompatible with IEEE 802.1p/802.1Q tagging. In order to use Large Receive Offload tagging must be disabled.	Warning	
0xA0020004L	NDIS6_EVENT_ LARGE_RECEIVE_ OFFLOAD_ UNAVAILABLE	IPv4 Large Receive Offload support will be unavailable whilst IPv4 receive checksum offload or TCP receive checksum offload is disabled.	Warning	
0xA0020005L	NDIS6_EVENT_ LARGE_SEND_ OFFLOAD_ UNAVAILABLE	IPv4 Large Send Offload support will be unavailable whilst IPv4 transmit checksum offload or TCP transmit checksum offload is disabled.	Warning	
0xA0020006L	NDIS6_EVENT_ DESIRED_ MAX_FRAME_SIZE_ UNAVAILABLE	The driver has determined that the network adapter's maximum frame size is not configured correctly and therefore has overridden *JumboFrames registry parameter. This issue may be resolved after a system restart.	Warning	

Table 62: NDIS Driver Status Codes

Value	Error Message	Description	Severity	Notes
0xA0020008L	NDIS6_EVENT_ TCP_CONNECTION_ OFFLOAD_ NOT_SUPPORTED	The driver has determined that the network adapter does not support TCP Connection Offload and therefore has overridden the *TCPConnection OffloadIPv4 registry parameter.	Warning	
0xA0020009L	NDIS6_EVENT_ TCP_CONNECTION_ OFFLOAD_ INCOMPATIBLE_WITH_ TAGGING	The driver has disabled TCP Connection Offload (overriding the *TCPConnection OffloadIPv4 registry parameter) as it is incompatible with IEEE 802.1p/802.1Q tagging. In order to use TCP Connection Offload tagging must be disabled.	Warning	
0xA002000AL	NDIS6_EVENT_ MAC_ADDRESS_ CHANGING_ NOT_SUPPORTED	The driver has determined that the network adapter does not support changing the current MAC address and therefore has overridden the *NetworkAddress registry parameter.	Warning	
0xA002000BL	NDIS6_EVENT_ TCP_CONNECTION_ OFFLOAD_ INCOMPATIBLE_WITH_ JUMBO_FRAMES	The driver has disabled TCP Connection Offload (overriding the *TCPConnection OffloadIPv4 registry parameter) as it is incompatible with MTU values greater than 1500 bytes. Refer to the driver release notes and errata for more information.	Warning	

Table 62: NDIS Driver Status Codes

Value	Error Message	Description	Severity	Notes
0xA002000CL	NDIS6_EVENT_ IPV6_CHECKSUM_ OFFLOAD_ NOT_SUPPORTED	The driver has determined that the network adapter does not support IPv6 checksum offloads and therefore has overridden the *TCPChksumOffv6 and *UDPChksumOffv6 registry parameters.	Warning	
0xA002000DL	NDIS6_EVENT_ WAKE_ON_LAN_ NOT_SUPPORTED	The driver has determined that the network adapter does not support Wake on LAN and therefore has overridden the *WakeOnPattern, *WakeOnMagicPacket and *DeviceSleepOn Disconnect registry parameters.	Warning	
0xA002000EL	NDIS6_EVENT_ PROTOCOL_OFFLOAD_ NOT_SUPPORTED	The driver has determined that the network adapter does not support protocol offload whilst the system is in a sleep state and therefore has overridden the *PMARPOffload and *PMNSOffload registry parameters.	Warning	
0xA002000FL	NDIS6_EVENT_ IPV6_LARGE_SEND_ OFFLOAD_ NOT_SUPPORTED	The driver has determined that the network adapter does not support IPv6 Large Send Offload and therefore has overridden the *LSOv2IPv6 registry parameter.	Warning	

Table 62: NDIS Driver Status Codes

Value	Error Message	Description	Severity	Notes
0xA0020010L	NDIS6_EVENT_ IPV6_LARGE_RECEIVE_ OFFLOAD_ UNAVAILABLE	IPv6 Large Receive Offload support will be unavailable whilst IPv6 TCP receive checksum offload is disabled.	Warning	
0xA0020011L	NDIS6_EVENT_ IPV6_LARGE_SEND_ OFFLOAD_ UNAVAILABLE	IPv6 Large Send Offload support will be unavailable whilst IPv6 TCP transmit checksum offload is disabled.	Warning	
0xA0020012L	NDIS6_EVENT_ VIRTUAL_MACHINE_ QUEUES_ NOT_SUPPORTED	The driver has determined that the network adapter does not support Virtual Machine Queues and therefore has overridden the *VMQ registry parameter.	Warning	
0xA0020013L	NDIS6_EVENT_ INSUFFICIENT_MSIX_ MESSAGES_TO_ SUPPORT_VIRTUAL_ MACHINE_QUEUES	The driver has been allocated insufficient MSI-X messages to support Virtual Machine Queues and therefore has overridden the *VMQ registry parameter.	Warning	
0xA0020014L	NDIS6_EVENT_ IPV4_RECEIVE_ SEGMENT_ COALESCING_ UNAVAILABLE	IPv4 Receive Segment Coalescing support will be unavailable whilst IPv4 receive checksum offload or TCP receive checksum offload is disabled.	Warning	
0xA0020015L	NDIS6_EVENT_ IPV6_RECEIVE_ SEGMENT_ COALESCING_ UNAVAILABLE	IPv6 Receive Segment Coalescing support will be unavailable whilst TCP receive checksum offload is disabled.	Warning	

Table 62: NDIS Driver Status Codes

Value	Error Message	Description	Severity	Notes
0xA0020016L	NDIS6_EVENT_ RECEIVE_SIDE_ SCALING_ UNAVAILABLE	The driver has determined that the network adapter does not support Receive Side Scaling and therefore has overridden the *Rss registry parameter.	Warning	
0xE0000001L	NDIS6_EVENT_ OUT_OF_RESOURCE	The driver could not allocate the resources necessary for operation.	Error	
0xE0000002L	NDIS6_EVENT_ HARDWARE_FAILURE	The driver has determined that the network adapter is not functioning properly.	Error	
0xE0000003L	NDIS6_EVENT_ DRIVER_FAILURE	The driver has encountered an internal error and has failed.	Error	
0xE0000004L	NDIS6_EVENT_ DRIVER_INTERFACE_ NOT_FOUND	The driver could not connect to the NDIS bus interface and has failed.	Error	
0xE0000005L	NDIS6_EVENT_ TCP_CONNECTION_ OFFLOAD_INTERFACE_ NOT_FOUND	The driver could not connect to the NDIS TCP Connection Offload interface and has failed.	Error	
0xE0030002L	NDIS6_EVENT_ FAULT_RX_LOCKED	The driver has encountered an unrecoverable network adapter receive data path error; a full system restart is required. Refer to the driver release notes and errata for more information.	Error	

5 Solarflare Adapters on VMware

This chapter includes procedures for installation and configuration of Solarflare adapters on VMware®. For details of SR-IOV and Virtualization refer to SR-IOV Virtualization Using ESXi on page 402.

5.1 System Requirements

Refer to Software Driver Support on page 13 for supported VMware host platforms.

5.2 VMware Feature Set

Table 63 lists the features available from the VMware host. The following options can also be configured on the guest operating system:

- Jumbo Frames
- Task Offloads
- Virtual LANs (VLANs)

Table 63: VMware Host Feature Set

Jumbo frames	Support for MTUs (Maximum Transmission Units) from 1500 bytes to 9000 bytes. • See Adapter MTU (Maximum Transmission Unit) on page 287
Task offloads	Support for TCP Segmentation Offload (TSO), Large Receive Offload (LRO), and TCP/UDP/IP checksum offload for improved adapter performance and reduced CPU processing requirements. • See TCP/IP Checksum Offload on page 288 • See TCP Segmentation Offload (TSO) on page 289
NetQueue	Support for NetQueue, a performance technology that significantly improves performance in 10 Gigabit Ethernet virtualized environments. • See VMware ESX NetQueue on page 286
Teaming	Improve server reliability by creating teams on either the host vSwitch, Guest OS or physical switch to act as a single adapter, providing redundancy against single adapter failure. • See Configuring Teams on page 267
Virtual LANs (VLANs)	Support for VLANs on the host, guest OS and virtual switch. • See Configuring VLANs on page 268
PXE booting	Support for diskless booting to a target operating system via PXE boot. • See Sfboot: Command Line Options on page 272 • See Solarflare Boot ROM Agent on page 435

Table 63: VMware Host Feature Set

Fault diagnostics	Support for comprehensive adapter and cable fault diagnostics and system reports. • See Running Adapter Diagnostics on page 269
Firmware updates	Support for Boot ROM and Phy transceiver firmware upgrades for in-field upgradable adapters. • See Upgrading Adapter Firmware with sfupdate on page 283

5.3 Installing Solarflare Drivers and Utilities on VMware

Using the VMware ESX Service Console

The service console is the VMware ESX Server command-line interface. It provides access to the VMware ESX Server management tools, includes a command prompt for direct management of the Server, and keeps track of all the virtual machines on the server as well as their configurations.

Installing on VMware ESX 5.0, ESXi 5.1, 5.5 and 6.0

To install or update the .VIB through the CLI:

esxcli software vib install -v <absolute PATH to the .vib>

To install or update the offline bundle

esxcli software vib install -d <absolute PATH to the .zip>

To install through the Update Manager

Import the package in to the Update Manager and add to a baseline, then follow the normal update process. To install a new package on to a host deploy the package as part of a Host Extension type baseline rather than a Host Upgrade type.

Granting access to the NIC from the Virtual Machine

To allow guest operating systems access to the Solarflare NIC, you will need to connect the device to a vSwitch to which the guest also has a connection. You can either connect to an existing vSwitch, or create a new vSwitch for this purpose. To create a new vSwitch:

1. Log in to the **VMware Infrastructure Client**.
2. Select the host from the inventory panel.
3. Select the **Configuration** tab.
4. Choose **Networking** from the **Hardware** box on the left of the resulting panel.
5. Click **Add Networking** on the top right.
6. Select **Virtual Machine** connection type and click **Next**.
7. Choose **Create a Virtual Switch** or **Use vSwitchX** as desired.
8. Follow the remaining on-screen instructions.

5.4 Configuring Teams

A team allows two or more network adapters to be connected to a virtual switch (vSwitch). The main benefits of creating a team are:

- Increased network capacity for the virtual switch hosting the team.
- Passive failover in the event one of the adapters in the team fails.

NOTE: The VMware ESX host only supports NIC teaming on a single physical switch or stacked switches.

To create a team

1. From the host, select the **Configuration** tab.
2. Select **Networking** from the **Hardware** section.
3. Select **Properties** for the Virtual Switch you want to create the team for.
4. Select the **vSwitch** from the dialog box and click **Edit**.
5. Select **NIC Teaming**.

You can configure the following settings:

- Load Balancing
- Network Failover Detection
- Notify Switches
- Failover
- Failover Order

5.5 Configuring VLANs

There are three methods for creating VLANs on VMware ESX:

1 Virtual Switch Tagging (VST)

2 External Switch Tagging (EST)

3 Virtual Guest Tagging (VGT)

For EST and VGT tagging, consult the documentation for the switch or for the guest OS.

To Configure Virtual Switch Tagging (VST)

With vSwitch tagging:

- All VLAN tagging of packets is performed by the virtual switch, before leaving the VMware ESX host.
- The host network adapters must be connected to trunk ports on the physical switch.
- The port groups connected to the virtual switch must have an appropriate VLAN ID specified.

NOTE: VMware recommend that you create or amend VLAN details from the physical console of the server, not via the Infrastructure Client, to prevent potential disconnections.

1 From the host, select the **Configuration** tab.

2 Select **Networking** from the **Hardware** section.

3 Select **Properties** for the Virtual Switch you want to create the team for.

4 Select a **Port Group** and click **Edit**.

5 Enter a valid VLAN ID (0 equals no VLAN).

6 Click **OK**.

Further Reading

- NIC teaming in VMware ESX Server:

  http://kb.vmware.com/selfservice/microsites/search.do?cmd=displayKC&docType=kc&externalId=1004088&sliceId=1&docTypeID=DT_KB_1_1&dialogID=40304190&stateId=0%200%2037866989

- VMware ESX Server host requirements for link aggregation:

  http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1001938

- VLAN Configuration on Virtual Switch, Physical Switch, and virtual machines:
  http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1003806

5.6 Running Adapter Diagnostics

You can use Ethtool to run adapter diagnostic tests. Tests can be run offline (default) or online. Offline runs the full set of tests, possibly causing normal operation interruption during testing. Online performs a limited set of tests without affecting normal adapter operation.

As root user, enter the following command:

```
ethtool --test vmnicX offline|online
```

The tests run by the command are as follows:

Table 64: Adapter Diagnostic Tests

Diagnostic Test	Purpose
core.nvram	Verifies the flash memory 'board configuration' area by parsing and examining checksums.
core.registers	Verifies the adapter registers by attempting to modify the writable bits in a selection of registers.
core.interrupt	Examines the available hardware interrupts by forcing the controller to generate an interrupt and verifying that the interrupt has been processed by the network driver.
tx/rx.loopback	Verifies that the network driver is able to pass packets to and from the network adapter using the MAC and Phy loopback layers.
core.memory	Verifies SRAM memory by writing various data patterns (incrementing bytes, all bit on and off, alternating bits on and off) to each memory location, reading back the data and comparing it to the written value.
core.mdio	Verifies the MII registers by reading from PHY ID registers and checking the data is valid (not all zeros or all ones). Verifies the MMD response bits by checking each of the MMDs in the Phy is present and responding.

Table 64: Adapter Diagnostic Tests

Diagnostic Test	Purpose
chanX eventq.poll	Verifies the adapter's event handling capabilities by posting a software event on each event queue created by the driver and checking it is delivered correctly. The driver utilizes multiple event queues to spread the load over multiple CPU cores (RSS).
phy.bist	Examines the PHY by initializing it and causing any available built-in self tests to run.

5.7 Solarflare Utilities Package

The Solarflare Linux and VMware ESX Utilities Source package (SF-105095-LS) is available from the Solarflare support download site: https://support.solarflare.com/ - (Downloads > Linux > Misc).

The utilities (sfboot, sfupdate, sfkey, sfctool) can be built on a standard Linux OS server i.e. RHEL, before the binary utilities are extracted from the RPM package and copied to the ESXi host.

1 Download package SF-105095-LS to a temp directory (e.g /tmp) on a standard Linux machine:

```
SF-105095-LS-Solarflare_Linux_and_VMware_ESX_Utilities_Source.zip
```

2 Unzip and build the binary RPM in the temp directory from the source package:

```
# cd /tmp
# unzip SF-105095-LS-Solarflare_Linux_and_VMware_ESX_Utilities_Source.zip
# rpmbuild --rebuild sfutils-<version>.src.rpm
```

The "Wrote..." line at the end of the build process identifies the location of the binary RPM:

```
Wrote: /root/rpmbuild/RPMS/x86_64/sfutils-<version>.x86_64.rpm
```

3 Identify where the binary tools would be installed (but don't actually install them):

```
# rpm -qlp /root/rpmbuild/RPMS/x86_64/sfutils-<version>.x86_64.rpm
/usr/sbin/sfaoepackage
/usr/sbin/sfboot
/usr/sbin/sfctool
/usr/sbin/sfkey
/usr/sbin/sfupdate
```

4 From the binary extract the utility tools (sfupdate example):

```
# rpm2cpio /root/rpmbuild/RPMS/x86_64/sfutils-<version>.x86_64.rpm \
  | cpio -ivd ./usr/sbin/sfupdate
```

NOTE the . (dot) before the `/usr/sbin/sfupdate` to create the directories and copy the file within the current directory.

5 Copy the binary utilities to the ESXi machine and run them.

By default remote command execution is disabled on an ESXi host so SSH access must be enabled from the console or from the vSphere client to allow remote login and secure copy - refer to the following documentation for SSH enabling procedures:

https://pubs.vmware.com/vsphere-51/index.jsp?topic=%2Fcom.vmware.vcli.getstart.doc%2Fcli_jumpstart.3.6.html

5.8 Configuring the Boot ROM with Sfboot

Sfboot is a command line utility for configuring Solarflare adapter Boot ROM options, including options for PXE and iSCSI booting. Using sfboot is an alternative to using **Ctrl+B** to access the Boot Rom agent during server startup.

See Configuring the Solarflare Boot ROM Agent on page 436 for more information on the Boot Rom agent.

Sfboot: Command Usage

Log in to the VMware Service Console as root, and enter the following command:

```
sfboot [--adapter=vmnicX] [options] [parameters]
```

Note that without `--adapter`, the `sfboot` command applies to all adapters that are present in the target host.

The format for the parameters are:

```
<parameter>=<value>
```


Sfboot: Command Line Options

Table 65 lists the options for `sfboot.exe`, Table 66 lists the available global parameters, and Table 67 lists the available per-adapter parameters. Note that command line options are case insensitive and may be abbreviated.

NOTE: Abbreviations in scripts should be avoided, since future updates to the application may render abbreviated scripts invalid.

Table 65: Sfboot Options

Option	Description
`-?`, `-h`, `--help`	Displays command line syntax and provides a description of each sfboot option.
`-V`, `--version`	Shows detailed version information and exits.
`--nologo`	Hide the version and copyright message at startup.
`-v`, `--verbose`	Shows extended output information for the command entered.
`-y`, `--yes`	Update without prompting.
`-s`, `--quiet` Aliases: `--silent`	Suppresses all output, including warnings and errors; no user interaction. You should query the completion code to determine the outcome of commands when operating silently (see Performance Tuning on Windows on page 225).
`--log <filename>`	Logs output to the specified file in the current folder or an existing folder. Specify `--silent` to suppress simultaneous output to screen, if required.
`--computer <computer_name>`	Performs the operation on a specified remote computer. Administrator rights on the remote computer is required.
`--list`	Lists all available Solarflare adapters. This option shows the ifname and MAC address. Note: this option may not be used in conjunction with any other option. If this option is used with configuration parameters, those parameters will be silently ignored.
-i, --adapter =<vmnicX>	Performs the action on the identified Solarflare network adapter. The adapter identifier `vmnicX` can be the name or MAC address, as output by the `--list` option. If `--adapter` is not included, the action will apply to all installed Solarflare adapters.
--clear	Resets all options to their default values. If an adapter is specified, options for the given adapter are reset, but global options (shown in Table 66) are not reset. Note that `--clear` can also be used with parameters, allowing you to reset to default values, and then apply the parameters specified.

The following global parameters in Table 66 are used to control the configurable parameters for the Boot ROM driver when running prior to the operating system booting.

Table 66: Sfboot Global Parameters

Parameter	Description
`boot-image=` `all|optionrom|uefi|disabled`	Specifies which boot firmware images are served-up to the BIOS during start-up. This parameter can not be used if the `--adapter` option has been specified.
`port-mode=` `default|1x10G|2x10G|4x10G|` `2x40G`	Configure the port mode to use. This is for SFN7000 series adapters only. The values specify the connectors available after using any splitter cables. The usable values are adapter-dependent: • SFN7*xx*2F: 1x10G, 2x10G (default) • SFN7*xx*4F: 2x10G, 4x10G (default) • SFN7*xx*2Q: 2x10G, 4x10G, 2x40G (default) Changes to this setting with `sfboot` require a cold reboot to become effective. MAC address assignments may change after altering this setting.
`firmware-variant=` `full-feature|ultra-low-latency|` `capture-packed-stream|auto`	Configure the firmware variant to use. This is for SFN7000 series adapters only: • the SFN7002F adapter is factory set to full-feature • all other adapters are factory set to auto. Default value = auto - means the driver will select a variant that meets its needs: • the VMware driver always uses full-feature • otherwise, ultra-low-latency is used. The ultra-low-latency variant produces best latency without support for TX VLAN insertion or RX VLAN stripping (not currently used features). It is recommended that Onload customers use the ultra-low-latency variant.
`insecure-filters=` `enabled|disabled`	If enabled bypass filter security on non-privileged functions. This is for SFN7000 series adapters only. This reduces security in virtualized environments. The default is disabled. When enabled a function (PF or VF) can insert filters not qualified by their own permanent MAC address. This is a requirement when using Onload or when using bonded interfaces.

Table 66: Sfboot Global Parameters

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>mac-spoofing=enabled|disabled</td><td>If enabled, non-privileged functions can create unicast filters for MAC addresses that are not associated with them. This is for SFN7000 series adapters only.

The default is disabled.

Changes to this setting with sfboot require a cold reboot to become effective.</td></tr>
<tr><td>rx-dc-size=8|16|32|64</td><td>Specifies the size of the descriptor cache for each receive queue. This is for SFN7000 series adapters only. The default is: 16 if the port-mode supports the maximum number of connectors for the adaptor 32 if the port-mode supports a reduced number of connectors.</td></tr>
<tr><td>tx-dc-size=8|16|32|64</td><td>Specifies the size of the descriptor cache for each transmit queue. This is for SFN7000 series adapters only. The default is: 32 if the port-mode supports the maximum number of connectors for the adaptor 64 if the port-mode supports a reduced number of connectors.</td></tr>
<tr><td>vi-count=<vi count></td><td>Sets the total number of virtual interfaces that will be available on the NIC.</td></tr>
</table>

The following per-adapter parameters in Table 67 are used to control the configurable parameters for the Boot ROM driver when running prior to the operating system booting.

Table 67: Sfboot Per-adapter Parameters

Parameter	Description
`link-speed=auto|10g|1g|100m`	Specifies the network link speed of the adapter used by the Boot ROM. The default is `auto`. On the 10GBASE-T adapters, `auto` instructs the adapter to negotiate the highest speed supported in common with its link partner. On SFP+ adapters, `auto` instructs the adapter to use the highest link speed supported by the inserted SFP+ module. On 10GBASE-T and SFP+ adapters, any other value specified will fix the link at that speed, regardless of the capabilities of the link partner, which may result in an inability to establish the link. `auto` Auto-negotiate link speed (default) `10G` 10G bit/sec `1G` 1G bit/sec `100M` 100M bit/sec
`linkup-delay=` `<delay time in seconds>`	Specifies the delay (in seconds) the adapter defers its first connection attempt after booting, allowing time for the network to come up following a power failure or other restart. This can be used to wait for spanning tree protocol on a connected switch to unblock the switch port after the physical network link is established. The default is 5 seconds.
`banner-delay=` `<delay time in seconds>`	Specifies the wait period for `Ctrl-B` to be pressed to enter adapter configuration tool. `<delay time in seconds>` = 0-256
`bootskip-delay=` `<delay time in seconds>`	Specifies the time allowed for `Esc` to be pressed to skip adapter booting. `<delay time in seconds>` = 0-256
`boot-type=pxe|iscsi|disabled`	Sets the adapter boot type – effective on next boot. `pxe` – PXE (Preboot eXecution Environment) booting `iscsi` – iSCSI (Internet Small Computer System Interface) booting `disabled` – Disable adapter booting

Table 67: Sfboot Per-adapter Parameters

Parameter	Description
initiator-dhcp=enabled|disabled	Enables or disables DHCP address discovery for the adapter by the Boot ROM except for the Initiator IQN (see initiator-iqn-dhcp). This option is only valid if iSCSI booting is enabled (boot-type=iscsi). If initiator-DHCP is set to disabled, the following options will need to be specified: initiator-ip=<IPv4 address> netmask=<IPv4 subnet mask> The following options may also be needed: gateway=<ip_address> primary-dns=<ip_address>
initiator-ip=<IPv4 address>	Specifies the IPv4 address (in standard "." notation form) to be used by the adapter when initiator-dhcp is disabled. Note that this option is only valid if iSCSI booting is enabled (boot-type=iscsi). Example: sfboot boot-type=iscsi initiator-dhcp=disabled initiator-ip=<192.168.1.3>
netmask=<IPv4 subnet mask>	Specifies the IPv4 subnet mask (in standard "." notation form) to be used by the adapter when initiator-dhcp is disabled. Note that this option is only valid if iSCSI booting is enabled (boot-type=iscsi). Example: sfboot boot-type=iscsi initiator-dhcp=disabled netmask=255.255.255.0
gateway=<IPv4 address>	Specifies the IPv4 subnet mask (in standard "." notation form) to be used by the adapter when initiator-dhcp is disabled. Note that this option is only valid if iSCSI booting is enabled (boot-type=iscsi). Example: sfboot boot-type=iscsi initiator-dhcp=disabled gateway=192.168.0.10

Table 67: Sfboot Per-adapter Parameters

Parameter	Description
`primary-dns=<IPv4 address>`	Specifies the IPv4 address (in standard "." notation form) of the Primary DNS to be used by the adapter when `initiator-dhcp` is disabled. This option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot boot-type=iscsi initiator-dhcp=disabled primary-dns=192.168.0.3`
`initiator-iqn-dhcp= enabled|disabled`	Enables or disables use of DHCP for the initiator IQN only.
`initiator-iqn=<IQN>`	Specifies the IQN (iSCSI Qualified Name) to be used by the adapter when `initiator-iqn-dhcp` is disabled. The IQN is a symbolic name in the "." notation form; for example: iqn.2009.01.com.solarflare, and is a maximum of 223 characters long. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot initiator-iqn-dhcp=disabled initiator-iqn=iqn.2009.01.com.solarflare adapter=2`
`lun-retry-count=<retry count>`	Specifies the number of times the adapter attempts to access and login to the Logical Unit Number (LUN) on the iSCSI Target before failing. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot lun-retry-count=3`

`target-dhcp=enabled|disabled`	Enables or disables the use of DHCP to discover iSCSI target parameters on the adapter.
	If `target-dhcp` is disabled, you must specify the following options:
	`target-server=<DNS name or IPv4 address>`
	`target-iqn=<IQN>`
	`target-port=<port number>`
	`target-lun=<LUN>`
	Example - Enable the use of DHCP to configure iSCSI Target settings:
	`sfboot boot-type=iscsi target-dhcp=enabled`

Table 67: Sfboot Per-adapter Parameters

Parameter	Description
`target-server=` `<DNS name or IPv4 address>`	Specifies the iSCSI target's DNS name or IPv4 address to be used by the adapter when `target-dhcp` is disabled. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot boot-type=iscsi target-dhcp=disabled target-server=192.168.2.2`
`target-port=<port number>`	Specifies the Port number to be used by the iSCSI target when `target-dhcp` is disabled. The default Port number is Port 3260. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot boot-type=iscsi target-dhcp=disabled target-port=3262` This option should only be used if your target is using a non-standard TCP Port.
`target-lun=<LUN>`	Specifies the Logical Unit Number (LUN) to be used by the iSCSI target when `target-dhcp` is disabled. The default LUN is 0. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`).
`target-iqn=<IQN>`	Specifies the IQN of the iSCSI target when `target-dhcp` is disabled. Maximum of 223 characters. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Note that if there are spaces contained in `<IQN>`, then the IQN must be wrapped in double quotes (“”). Example: `sfboot target-dhcp=disabled target-iqn=iqn.2009.01.com.solarflare adapter=2`
`vendor-id=<vendor identifier>`	Specifies the device vendor ID to be advertised to the DHCP server. This must match the vendor id configured at the DHCP server when using DHCP option 43 to obtain the iSCSI target.

Table 67: Sfboot Per-adapter Parameters

Parameter	Description
chap=enabled|disabled	Enables or disables the use of Challenge Handshake Protocol (CHAP) to authenticate the iSCSI connection. Note that this option is only valid if iSCSI booting is enabled (boot-type=iscsi). To be valid, this option also requires the following sub-options to be specified: username=<initiator username> secret=<initiator password> Example: sfboot boot-type=iscsi chap=enabled username=initiatorusername secret=initiatorsecret
username=<username>	Specifies the CHAP initiator username (maximum 64 characters). Note that this option is required if either CHAP or Mutual CHAP is enabled (chap=enabled, mutual-chap=enabled). Note that if there are spaces contained in <username>, then it must be wrapped in double quotes (“”). Example: sfboot boot-type=iscsi chap=enabled username=username
secret=<secret>	Specifies the CHAP initiator secret (minimum 12 characters, maximum 20 characters). Note that this option is valid if either CHAP or Mutual CHAP is enabled (chap=enabled, mutual-chap=enabled). Note that if there are spaces contained in <secret>, then it must be wrapped in double quotes (“”). Example: sfboot boot-type=iscsi chap=enabled username=username secret=veryverysecret

Table 67: Sfboot Per-adapter Parameters

Parameter	Description

`mutual-chap=enabled|disabled`	Enables/disables Mutual CHAP authentication when iSCSI booting is enabled.
	
	This option also requires the following sub-options to be specified:
	
	`target-username=<username>`
	`target-secret=<password>`
	`username=<username>`
	`secret=<password>`
	
	Example:
	
	`sfboot boot-type=iscsi mutual-chap=enabled username=username secret=veryverysecret target-username=targetusername target-secret=anothersecret`

`target-username=<username>`	Specifies the username that has been configured on the iSCSI target (maximum 64 characters). Note that this option is necessary if Mutual CHAP is enabled on the adapter (`mutual-chap=enabled`). Note that if there are spaces contained in `<username>`, then it must be wrapped in double quotes (“”).
`target-secret=<secret>`	Specifies the secret that has been configured on the iSCSi target (minimum 12 characters; maximum 20 characters). Note: This option is necessary if Mutual CHAP is enabled on the adapter (`mutual-chap=enabled`). Note that if there are spaces contained in `<secret>`, then it must be wrapped in double quotes (“”).
`mpio-priority=<MPIO priority>`	Specifies the Multipath I/O (MPIO) priority for the adapter. This option is only valid for iSCSI booting over multi-port adapters, where it can be used to establish adapter port priority. The range is 1- 255, with 1 being the highest priority.
`mpio-attempts=<attempt count>`	Specifies the number of times MPIO will try and use each port in turn to login to the iSCSI target before failing.
`pf-count=<pf count>`	This is the number of available PCIe PFs per physical network port. This setting is applied to all ports on the adapter. Changes to this setting with `sfboot` require a cold reboot to become effective. MAC address assignments may change after altering this setting.

Table 67: Sfboot Per-adapter Parameters

Parameter	Description
`msix-limit=` `8|16|32|64|128|256|512|1024`	Specifies the maximum number of MSI-X interrupts that each PF will use. The default is 32. Note: Using the incorrect setting can impact the performance of the adapter. Contact Solarflare technical support before changing this setting.
`sriov=enabled|disabled`	Enable SR-IOV support for operating systems that support this. Not required on SFN7000 series adapters.
`vf-count=<vf count>`	The number of virtual functions (VF) advertised to the operating system for each Physical Function on this physical network port. SFN7000 series adapters have a total limit of 2048 interrupts. Earlier adapters support a total limit of 127 virtual functions per port and a total of 1024 interrupts. Depending on the values of msix-limit and vf-msix-limit, some of these virtual functions may not be configured. Enabling all 127 VFs per port with more than one MSI-X interrupt per VF may not be supported by the host BIOS - in which case you may get 127 VFs on one port and none on others. Contact your BIOS vendor or reduce the VF count. The sriov parameter is implied if vf-count is greater than zero. Changes to this setting with `sfboot` require a cold reboot to become effective.
`vf-msix-limit=` `1|2|4|8|16|32|64|128|256`	The maximum number of interrupts a virtual function may use.

Table 67: Sfboot Per-adapter Parameters

Parameter	Description
`pf-vlans=<tag>[,<tag>[,...]]|none`	Comma separated list of VLAN tags for each PF in the range 0-4094 - see sfboot --help for details.
`switch-mode= default|sriov|partitioning| partitioning-with-sriov|pfiov`	Specifies the mode of operation that the port will be used in: `default` - single PF created, zero VFs created. `sriov` - SR-IOV enabled, single PF created, VFs configured with `vf-count`. `partitioning` - PFs configured with `pf-count`, VFs configured with `vf-count`. See NIC Partitioning on page 58 for details. `partitioning-with-sriov` - SR-IOV enabled, PFs configured with `pf-count`, VFs configured with `vf-count`. See NIC Partitioning on page 58 for details. `pfiov` - PFIOV enabled, PFs configured with `pf-count`, VFs not supported. Changes to this setting with `sfboot` require a cold reboot to become effective.

Sfboot: Examples

- Show the current boot configuration for all adapters:

  `Sfboot`

```
Solarflare boot configuration utility [v3.0.3]
Copyright Solarflare Communications 2006-2010, Level 5 Networks 2002-2005

eth1:
  Boot image                        Option ROM and UEFI
    Link speed                      Negotiated automatically
    Link-up delay time              5 seconds
    Banner delay time               2 seconds
    Boot skip delay time            5 seconds
    Boot type                       Disabled
  MSI-X interrupt limit             32

eth2:
  Boot image                        Option ROM and UEFI
    Link speed                      Negotiated automatically
    Link-up delay time              5 seconds
    Banner delay time               2 seconds
    Boot skip delay time            5 seconds
    Boot type                       Disabled
  MSI-X interrupt limit             32
```


5.9 Upgrading Adapter Firmware with sfupdate

Sfupdate is a command line utility used to manage and upgrade the Solarflare adapter Boot ROM, Phy and adapter firmware. Embedded within the sfupdate executable is firmware images for various Solarflare adapters - the exact updates available via sfupdate are therefore depend on your adapter.

Sfupdate: Command Usage

Log in to the VMware Service Console as root, and enter the following command:

```
sfupdate [--adapter=vmnicX] [options]
```

where:

- `vmnicX` is the interface name of the Solarflare adapter you want to upgrade. Specifying the adapter is optional - if it is not included the command is applied to all Solarflare adapters in the machine.
- `option` is one of the command options listed in Table 68.

The format for the options are:

```
<option>=<parameter>
```

Running the command `sfupdate` with no additional parameters will show the current firmware version for all Solarflare adapters and whether the firmware within sfupdate is more up to date. To update the firmware for all Solarflare adapters run the command `sfupdate --write`

Solarflare recommend that you use sfupdate in the following way:

1. Run `sfupdate` to check that the firmware on all your adapters are up to date.
2. Run `sfupdate --write` to update the firmware on all adapters.

Sfupdate: Command Line Options

Table 68 lists the options for `sfupdate`.

Table 68: Sfupdate Options

Option	Description
`-h, --help`	Shows help for the available options and command line syntax.
`-v, --verbose`	Verbose output mode.
`-s, --silent`	Suppress all output except errors. Useful for scripting.
`-V, --version`	DIsplay version number information and exit.
`-i, --adapter=vmnicX`	Specifies the target adapter when more than one adapter is installed in the local host. **vmnicX** = Adapter interface name or MAC address (as obtained with `--list`).
`--list`	Shows the adapter ID, adapter name and MAC address of each adapter installed in the local host, or on the target when `--computer` is specified.
`--write`	Writes the firmware from the images embedded in `sfupdate`. To use an external image, specify `--image=<filename>` in the command. `--write` fails if the embedded image is the same or a previous version to that in the adapter. To force a write in this case, specify `--force` in the command.
`--force`	Force update of all firmware, even if the installed firmware version is the same or more recent. If required, use this option with `--write`.
`--image=<filename>`	Specifies a specific firmware image. This option is not normally required and is only necessary if you need to provide writing the `sfupdate` embedded image file.
`-y, --yes`	Prompts for user confirmation before writing the firmware.

Sfupdate: Examples

- List all Solarflare adapters installed on the host with the installed firmware:

  `sfupdate`

```
Solarflare firmware update utility [v3.0.3]
Copyright Solarflare Communications 2006-2010, Level 5 Networks 2002-2005

eth1 - MAC: 00-0F-53-01-39-70
    Firmware version:   v3.0.3
    PHY type:           QT2025C
```

```
    PHY version:        v2.0.2.5
    Controller type:    Solarflare SFC4000
    Controller version: v3.0.3.2127
    BootROM version:    v3.0.3.2127

The PHY firmware is up to date
The BootROM firmware is up to date
The controller firmware is up to date

eth2 - MAC: 00-0F-53-01-39-71
    Firmware version:   v3.0.2
    PHY type:           QT2025C
    PHY version:        v2.0.2.5

The PHY firmware is up to date
```

5.10 Performance Tuning on VMware

- Introduction on page 285
- Tuning Settings on page 286
- Other Considerations on page 290

Introduction

The Solarflare family of network adapters are designed for high-performance network applications. The adapter driver is pre-configured with default performance settings that have been designed to give good performance across a broad class of applications. In many cases, application performance can be improved by tuning these settings to best suit the application.

There are three metrics that should be considered when tuning an adapter:

- Throughput
- Latency
- CPU utilization

Different applications may be more or less affected by improvements in these three metrics. For example, transactional (request-response) network applications can be very sensitive to latency whereas bulk data transfer applications are likely to be more dependent on throughput.

The purpose of this section is to highlight adapter driver settings that affect the performance metrics described. This guide covers the tuning of all members of the Solarflare family of adapters. Performance between adapters should be identical, with the exception of latency measurements.

Latency will be affected by the type of physical medium used: CX4, XFP, 10GBase-T or SFP+. This is because the physical media interface chip (PHY) used on the adapter can introduce additional latency.

Tuning Settings

Install VMware Tools in the Guest Platform

Installing VMware tools will give greatly improved networking performance in the guest. If VMware Tools are not installed, ESX emulates a PC-Lance device in the guest. If VMware Tools are installed, the guest will see a virtual adapter of type vmxnet.

To check that VMware Tools are installed:

1. From the **VMware Infrastructure Client,** power on the virtual machine and click the **Summary** tab.
2. In the **General** panel, check the status of **VMware Tools**.

To install VMware Tools:

1. Power on the virtual machine
2. From the **Inventory > Virtual Machine** menu, select **Install/Upgrade VMware Tools**.

This will mount a virtual CD-ROM in the guest OS. If the guest OS is Windows, it can autorun the CD and install tools (if not, navigate to the CD-ROM device and run the setup program yourself). If the guest is a Linux OS, you must mount the CD, install the tools, and configure them. For example, if the guest is Red Hat:

```
# mount /dev/cdrom /mnt
# rpm -i /mnt/VMwareTools*.rpm
# vmware-tools-config.pl
```

VMware ESX NetQueue

Solarflare adapters supports VMware's NetQueue technology, which accelerates network performance in 10 Gigabit Ethernet virtualized environments. NetQueue is enabled by default in VMware versions. There is usually no reason not to enable NetQueue.

NOTE: VMware NetQueue accelerates both receive and transmit traffic.

Binding NetQueue queues and Virtual Machines to CPUs

Depending on the workload, NetQueue can show improved performance if each of the queues' associated interrupt and the virtual machine are pinned to the same CPU. This is particularly true of workloads where sustained high bandwidth is evenly distributed across multiple virtual machines (such as you might do when benchmarking). To pin a Virtual Machine to one or more CPUs:

1. Log in to the VMware Infrastructure Client.
2. Expand the host and select the virtual machine to pin from the inventory panel.
3. Select the **Summary** tab for that virtual machine.
4. Click **Edit Settings**.

5 From the resulting dialog box select the **Resources** tab

6 Click **Advanced CPU** on the left.

7 Select the CPU(s) to which the virtual machine is to be bound (on the right hand side of the dialog box).

To bind a queue's interrupt to a CPU, from the VMware ESX console OS enter:

```
# echo move $IRQVEC $CPU > /proc/vmware/intr-tracker
```

(Where `$IRQVEC` is the interrupt vector in hex, and `$CPU` is the CPU number in decimal.)

To determine the value for `$IRQVEC` enter:

```
# cat /proc/vmware/interrupts
```

Locate the interrupts associated with the Solarflare adapter (e.g. vmnic2). Interrupts are listed in order: the first interrupt will be for the **default** queue, the second interrupt for the queue dedicated to the first virtual machine to have been started, the third interrupt for the queue dedicated to the second virtual machine to have been started, and so on.

If there are more virtual machine's than CPUs on the host, optimal performance is obtained by pinning each virtual machine and associated interrupt to the same CPU. If there are fewer virtual machines than CPUs, optimal results are obtained by pinning the virtual machine and associated interrupt respectively to two cores which share an L2 cache.

Adapter MTU (Maximum Transmission Unit)

The default MTU of 1500 bytes ensures that the adapter is compatible with legacy 10/100Mbps Ethernet endpoints. However if a larger MTU is used, adapter throughput and CPU utilization can be improved. CPU utilization is improved because it takes fewer packets to send and receive the same amount of data. Solarflare adapters support frame sizes up to 9216 bytes (this does not include the Ethernet preamble or frame-CRC).

Since the MTU should ideally be matched across all endpoints in the same LAN (VLAN), and since the LAN switch infrastructure must be able to forward such packets, the decision to deploy a larger than default MTU requires careful consideration. It is recommended that experimentation with MTU be done in a controlled test environment.

To change the MTU of the vSwitch, from the VMware Console OS enter:

```
# esxcfg-vswitch --mtu <size> <vSwitch>
```

To verify the MTU settings, as well as obtaining a list of vSwitches installed on the host, enter:

```
# esxcfg-vswitch --list
```

The change in MTU size of the vSwitch will persist across reboots of the VMware ESX host.

Interrupt Moderation (Interrupt Coalescing)

Interrupt moderation controls the number of interrupts generated by the adapter by adjusting the extent to which receive packet processing events are coalesced. Interrupt moderation may coalesce more than one packet-reception or transmit-completion event into a single interrupt.

By default, adaptive moderation is enabled. Adaptive moderation means that the network driver software adapts the interrupt moderation setting according to the traffic and workloads it sees.

Alternatively, you can set the moderation interval manually. You would normally only do this if you are interested in reducing latency. To do this you must first disable adaptive moderation with the following command, where `vmnicX` is the interface name.

```
ethtool -C <vmnicX> adaptive-rx off
```

NOTE: `adaptive-rx` may already have been disabled. Consult your VMware documentation for details.

Interrupt moderation can be changed using ethtool, where vmnicX is the interface name and interval is the moderation setting in microseconds (µs). Specifying 0 as the interval parameter will turn interrupt moderation off:

```
ethtool –C <vmnicX> rx-usecs-irq <interval>
```

Verification of the moderation settings may be performed by running `ethtool –c`

This parameter is critical for tuning adapter latency. Increasing the moderation value will increase latency, but reduce CPU utilization and improve peak throughput, if the CPU is fully utilized. Decreasing the moderation value or turning it off will decrease latency at the expense of CPU utilization and peak throughput. However, for many transaction request-response type network applications, the benefit of reduced latency to overall application performance can be considerable. Such benefits may outweigh the cost of increased CPU utilization.

NOTE: The interrupt moderation time dictates the minimum gap between two consecutive interrupts. It does not mandate a delay on the triggering of an interrupt on the reception of every packet. For example, an interrupt moderation setting of 30µs will not delay the reception of the first packet received, but the interrupt for any following packets will be delayed until 30µs after the reception of that first packet.

TCP/IP Checksum Offload

Checksum offload moves calculation and verification of IP Header, TCP and UDP packet checksums to the adapter. The driver by default has all checksum offload features enabled. Therefore, there is no opportunity to improve performance from the default.

Checksum offload is controlled using ethtool:

Receive Checksum:

```
# /sbin/ethtool –K <vmnicX> rx <on|off>
```


Transmit Checksum:

```
# /sbin/ethtool -K <vmnicX> tx <on|off>
```

Verification of the checksum settings may be performed by running ethtool with the –k option. Solarflare recommend you do not disable checksum offload.

For advice on configuring checksum offload in the guest, consult the relevant Solarflare section for that guest, or the documentation for the guest operating system.

TCP Segmentation Offload (TSO)

TCP Segmentation offload (TSO) offloads the splitting of outgoing TCP data into packets to the adapter. TCP segmentation offload benefits applications using TCP. Non TCP protocol applications will not benefit (but will not suffer) from TSO.

Enabling TCP segmentation offload will reduce CPU utilization on the transmit side of a TCP connection, and so improve peak throughput, if the CPU is fully utilized. Since TSO has no effect on latency, it can be enabled at all times. The driver has TSO enabled by default. Therefore, there is no opportunity to improve performance from the default.

NOTE: TSO cannot be controlled via the host on VMware ESX. It can only be controlled via the guest Operating System.

TCP Large Receive Offload (LRO)

TCP Large Receive Offload (LRO) is a feature whereby the adapter coalesces multiple packets received on a TCP connection into a single call to the operating system TCP Stack. This reduces CPU utilization, and so improves peak throughput when the CPU is fully utilized.

LRO should not be enabled if you are using the host to forward packets from one interface to another; for example if the host is performing IP routing or acting as a layer2 bridge.

LRO is supported, and enabled by default, on VMware versions later than ESX 3.5.

TCP Protocol Tuning

TCP Performance can also be improved by tuning kernel TCP settings. Settings include adjusting send and receive buffer sizes, connection backlog, congestion control, etc.

Typically it is sufficient to tune just the max buffer value. It defines the largest size the buffer can grow to. Suggested alternate values are max=500000 (1/2 Mbyte). Factors such as link latency, packet loss and CPU cache size all influence the affect of the max buffer size values. The minimum and default values can be left at their defaults minimum=4096 and default=87380.

For advice on tuning the guest TCP stack consult the documentation for the guest operating system.

Receive Side Scaling (RSS)

Solarflare adapters support Receive Side Scaling (RSS). RSS enables packet receive-processing to scale with the number of available CPU cores. RSS requires a platform that supports MSI-X interrupts. RSS is enabled by default.

When RSS is enabled the controller uses multiple receive queues into which to deliver incoming packets. The receive queue selected for an incoming packet is chosen in such a way as to ensure that packets within a TCP stream are all sent to the same receive queue – this ensures that packet-ordering within each stream is maintained. Each receive queue has its own dedicated MSI-X interrupt which ideally should be tied to a dedicated CPU core. This allows the receive side TCP processing to be distributed amongst the available CPU cores, providing a considerable performance advantage over a conventional adapter architecture in which all received packets for a given interface are processed by just one CPU core.

RSS will be enabled whenever NetQueue is not and Solarflare recommend using NetQueue on VMware ESX hosts.

Other Considerations

PCI Express Lane Configurations

The PCI Express (PCIe) interface used to connect the adapter to the server can function at different widths. This is independent of the physical slot size used to connect the adapter. The possible widths are multiples x1, x2, x4, x8 and x16 lanes of (2.5Gbps for PCIe Gen. 1, 5.0 Gbps for PCIe Gen. 2) in each direction. Solarflare Adapters are designed for x8 lane operation.

On some server motherboards, choice of PCIe slot is important. This is because some slots (including ones that are physically x8 or x16 lanes) may only electrically support x4 lanes. In x4 lane slots, Solarflare PCIe adapters will continue to operate, but not at full speed. The Solarflare driver will warn you if it detects the adapter is plugged into a PCIe slot which electrically has fewer than x8 lanes.

For SFN5xxxx adapters, which require a PCIe Gen. 2 slot for optimal operation, a warning will be given if they are installed in a PCIe Gen. 1 slot. Warning messages can be viewed in `dmesg` from `/var/log/messages`.

Memory bandwidth

Many chipsets/CPUs use multiple channels to access main system memory. Maximum memory performance is only achieved when the server can make use of all channels simultaneously. This should be taken into account when selecting the number of DIMMs to populate in the server. Consult your motherboard documentation for details.

Intel® QuickData

Intel® QuickData Technology allows VMware ESX to data copy by the chipset instead of the CPU, to move data more efficiently through the server and provide fast, scalable, and reliable throughput.

I/O AT can be enabled on the host and on guest operating systems. For advice on enabling I/OAT in the guest, consult the relevant Solarflare section for that guest, or the documentation for the guest operating system. I/OAT must be enabled on the host if it is to be used in the guests.

To enable I/OAT on the VMware ESX host:

On some systems the hardware associated with I/OAT must first be enabled in the BIOS

Log in to the ConsoleOS on the VMware ESX host, and enter:

```
# esxcfg-advcfg -s 1 /Net/TcpipUseIoat
```

Reboot the VMware ESX host

To verify I/OAT is enabled, from the ConsoleOS enter:

```
# vmkload_mod -l | grep -i ioat
```

NOTE: The following VMware KB article should be read when enabling I/OAT:

http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1003712

Server Motherboard, Server BIOS, Chipset Drivers

Tuning or enabling other system capabilities may further enhance adapter performance. Readers should consult their server user guide. Possible opportunities include tuning PCIe memory controller (PCIe Latency Timer setting available in some BIOS versions).

6 Solarflare Adapters on Solaris

This chapter covers the following topics on the Solaris platform:

6.1 System Requirements

Refer to Software Driver Support on page 13 for details of supported Solaris distributions.

Solarflare drivers for Solaris support the GLDv3 API, but do not support the Crossbow API framework.

6.2 Solaris Platform Feature Set

Table 69 lists the features supported by Solarflare adapters on Solaris.

Table 69: Solaris Feature Set

Jumbo frames	Support for MTUs (Maximum Transmission Units) to 9000 bytes. • See Configuring Jumbo Frames on page 297
Task offloads	Support for TCP Segmentation Offload (TSO), Large Receive Offload (LRO), and TCP/UDP/IP checksum offload for improved adapter performance and reduced CPU processing requirements. • See Configuring Task Offloading on page 297
Receive Side Scaling (RSS)	Support for RSS multi-core load distribution technology. • See Receive Side Scaling (RSS) on page 316
Virtual LANs (VLANs)	Support for multiple VLANs per adapter. • See Setting Up VLANs on page 298
PXE and booting	Support for diskless booting to a target operating system via PXE or iSCSI boot. • See Configuring the Boot ROM with sfboot on page 299 • See Solarflare Boot ROM Agent on page 435
Firmware updates	Support for Boot ROM, PHY transceiver and adapter firmware upgrades. • See Upgrading Adapter Firmware with sfupdate on page 310

6.3 Installing Solarflare Drivers

The Solaris drivers for Solarflare are available in a binary package for both 32 and 64 bit platforms.

- A driver package (pkg format) is available for Solaris 10.8, 10.9 and 10.10.
- A driver package (pkg format) is available for Solaris 11.0.

NOTE: The Solarflare adapter should be physically installed in the host computer before you attempt to install drivers. You must have root permissions to install the adapter drivers.

1 As a root user enter:

```
pkgadd -d SFCsfxge_sol10_i386_<version>.pkg SFCsfxge
```

or

```
pkgadd -d SFCsfxge_sol11_i386_<version>.pkg SFCsfxge
```

Output similar to the following will be displayed:

```
Solarflare 10GE NIC Driver(i386) <DRIVER VERSION>
<LICENSE INFO>
This package contains scripts which will be executed with super-user
permission during the process of installing this package.
Do you want to continue with the installation of <SFCsfxge> [y,n,?]
```

2 Enter 'y'. The installation will continue.

3 The following information will be displayed:

```
Installing Solarflare 10GE NIC Driver as <SFCsfxge>
## Installing part 1 of 1.
/kernel/drv/amd64/sfxge
/kernel/drv/sfxge
[ verifying class <none> ]
## Executing postinstall script.
Installation of <SFCsfxge> was successful.
```

6.4 Unattended Installation Solaris 10

Unattended installations of Solaris 10 are done via JumpStart. For general information on JumpStart see:

http://www.oracle.com/technetwork/articles/servers-storage-admin/installjumpstartons11ex-219229.html

The process for using JumpStart is as follows:

- Create a JumpStart installation server
- Create the client configuration files
- Share the client tftpboot files
- Configure and run the DHCP Server
- Perform a hands-off JumpStart installation

These processes are documented here:

http://docs.oracle.com/cd/E19253-01/819-6397/819-6397.pdf

NOTE: The Solarflare server adapter can be used to PXE boot the installer, but as there is no driver, the adapter can not be used during installation.

To install the Solarflare Solaris package as part of an unattended installation, it must be added using the **package** command to the JumpStart machine profile. The package can reside on a local disk or on a HTTP or NFS server. For more information, see:

http://search.oracle.com/sear ch/search?start=1&search_p_main_operator=all&q=package+command&group=Technology+Network

The following are example lines for a JumpStart profile:

```
package SFCsfxge add local_device <device> <path> <file_system_type>
package SFCsfxge add http://<server_name>[:<port>] <path> [<options>]
package SFCsfxge add nfs://<server_name>:/<path> [retry <n>]
```

Table 70 shows an example time line for an unattended installation.

Table 70: Installation Stages

In Control	Stages of Boot	Setup needed
BIOS	PXE code on the adapter runs.	Adapter must be in PXE boot mode. See PXE Support on page 436.
SF Boot ROM (PXE)	DHCP request from PXE (SF Boot ROM).	Jumpstart server must be installed and configured.
SF Boot ROM (PXE)	TFTP request for filename to next-server, e.g. pxegrub.0	
pxegrub	TFTP retrieval of grub configuration.	
pxegrub	TFTP menu retrieval of Solaris kernel image.	
Solaris kernel/installer	Installer retrieves configuration.	
Installation occurs	Machine reboots	
Target Solaris kernel	kernel reconfigures network adapters.	DHCP server.

6.5 Unattended Installation Solaris 11

Please refer to the Oracle Solaris 11 documentation for details of transitioning from Solaris 10 to 11 or for details of the Automated Installer feature for Solaris 11.

https://blogs.oracle.com/unixman/entry/how_to_get_started_with

https://blogs.oracle.com/unixman/entry/migrating_from_jumpstart_to_automated

6.6 Configuring the Solarflare Adapter

NOTE: The examples below demonstrate the *Solaris 10.x* configuration command. *Solaris 11* users should refer to Solaris documentation for the equivalent Solaris 11 configuration commands.

The drivers will be loaded as part of the as part of the installation. However the adapter will not be plumbed (implement the TCP/IP stack) or configured (adding IP address and netmask).

Each Solarflare network adapter interface will be named `sfxge<x>` where `<x>` is a unique identifier. There will be one interface per physical port on the Solarflare adapter.

To plumb an interface enter the following:

```
ifconfig sfxge<x> plumb
```

You then need to configure the interface and bring it up to allow data to pass. Enter the following:

```
ifconfig sfxge<x> <IPv4 address> netmask <netmask> up
```

This configures the interface and initializes it with the up command.

NOTE: This method of plumbing and configuring is temporary. If you reboot your computer the settings will be lost. To make these settings permanent, create the configuration files as described below.

Using IPv6

To plumb and configure using IPv6, enter the following:

```
ifconfig sfxge<x> inet6 plumb
ifconfig sfxge<x> inet6 up
```

Then create an IPv6 interface sfxge<x> interface with a link local IPv6 address by entering:

```
ifconfig sfxge<x> inet6 addif <IPv6 address>/<ipv6 prefix length> up
```

This will give an IPv6 interface name of `sfxge<x>:1`

Using Configuration Files with IPv4

There are three options when using a configuration file with IPv4:

1 Using a static IPv4 address. To use this option, add `<IPv4 address> <netmask>` to:

   `/etc/hostname.sfxge<x>`

2 Using a static IPv4 hostname. To use this option, add `<hostname>` to:

   `/etc/hostname.sfxge<x>`

   and modify `/etc/hosts` and `/etc/netmasks`

3 Using DHCP. To use this option, enter:

```
touch /etc/hostname.sfxge<4>
```

and:

```
touch /etc/dhcp.sfxge<4>
```

Using Configuration files with IPv6

To make the interface settings permanent, you need to create the following file per interface:

```
/etc/hostname6.sfxge<x>
```

This enables the interface to be plumbed and configured when the computer is booted. For example:

```
touch /hostname6.sfxge<x>
```

For a static IP address, add your IPv6 address to `/etc/hostname6.sfxge<x>`.

Or add your hostname to `/etc/hostname6.sfxge<x>` and edit the following:

```
/etc/hosts
```

DHCP and IPv6

Unlike for IPv4, no file is required for DHCP in. The DHCP Daemons are automatically started. Consult the `man dhcp` pages for more details.

Configuring Task Offloading

Solarflare adapters support IPv4 TCP and UDP transmit (Tx) and receive (Rx) checksum offload, as well as TCP segmentation offload. To ensure maximum performance from the adapter, all task offloads should be enabled, which is the default setting on the adapter. For more information, see Performance Tuning on Solaris on page 313.

Configuring Jumbo Frames

The maximum driver MTU size can be set in `sfxge.conf`. This setting is applied across all Solarflare adapters. The default setting in `sfxge.conf` is 1500.

Solarflare adapters support frame sizes from 1500 bytes to 9000 bytes. For example, to set a new frame size (MTU) of 9000 bytes, enter the following command:

```
$ ifconfig sfxge<x> mtu 9000
```

To view the current MTU, enter:

```
$ ifconfig sfxge<x>
sfxge0: flags=1001000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 9000
```

If you want to have an MTU configured when the interface is brought up add `mtu` to the single line of configuration data in `/etc/hostname.sfxge<X>`. For example:

```
[<IP address>] mtu <size>
```


6.7 Setting Up VLANs

VLANs offer a method of dividing one physical network into multiple broadcast domains. In enterprise networks, these broadcast domains usually match with IP subnet boundaries, so that each subnet has its own VLAN. The advantages of VLANs include:

- Performance
- Ease of management
- Security
- Trunks
- You don't have to configure any hardware device, when physically moving your server to another location.

To have a single interface exist on multiple VLANs (if the port on the connected switch is set to “trunked” mode) see the “How to Configure a VLAN” section in the following documentation:

http://docs.oracle.com/cd/E19253-01/816-4554/fpdga/index.html

6.8 Solaris Utilities Package

The Solarflare Solaris Utilities package, supplied as a 32 bit SVR package and available from https://support.solarflare.com/ contains the following utilities:

Table 71: Utilities Package

Utility File	Description
sfupdate	A command line utility that contains an adapter firmware version which can update Solarflare adapter firmware.
sfboot	A command line utility for configuring Solarflare adapter Boot ROM options, including PXE and iSCSI booting.
sfreport	A command line utility that generates a diagnostic log file providing diagnostic data about the server and Solarflare adapters.

Once installed (`pkgadd -d SFCutils_i386_v<version>.pkg`), by default, utility files are located in the `/opt/SFCutils/bin` directory.

6.9 Configuring the Boot ROM with sfboot

- Sfboot: Command Usage on page 299
- Sfboot: Command Line Options on page 299
- Sfboot: Examples on page 309

Sfboot is a command line utility for configuring Solarflare adapter Boot ROM options, including PXE and iSCSI booting. Using sfboot is an alternative to using **Ctrl + B** to access the Boot Rom agent during server startup.

See Configuring the Solarflare Boot ROM Agent on page 436 for more information on the Boot Rom agent.

Sfboot: Command Usage

The general usage for sfboot is as follows (as root):

```
sfboot [--adapter=sfxge<x>] [options] [parameters]
```

Note that without `--adapter`, the `sfboot` command applies to all adapters that are present in the target host.

The format for the parameters are:

```
<parameter>=<value>
```

Sfboot: Command Line Options

Table 72 lists the options for `sfboot`, Table 73 lists the available global parameters, and Table 74 lists the available per-adapter parameters. Note that command line options are case insensitive and may be abbreviated.

NOTE: Abbreviations in scripts should be avoided, since future updates to the application may render abbreviated scripts invalid.

Table 72: Sfboot Options

Option	Description
`-h, --help`	Displays command line syntax and provides a description of each sfboot option.
`-V, --version`	Shows detailed version information and exits.
`-v, --verbose`	Shows extended output information for the command entered.
`-s, --silent`	Suppresses all output, including warnings and errors; no user interaction. You should query the completion code to determine the outcome of commands when operating silently.
`--log <filename>`	Logs output to the specified file in the current folder or an existing folder. Specify `--silent` to suppress simultaneous output to screen, if required.

Table 72: Sfboot Options

Option	Description
`--computer <computer_name>`	Performs the operation on a specified remote computer. Administrator rights on the remote computer is required.
`--list`	Lists all available Solarflare adapters. This option shows the adapter's ID number, ifname and MAC address. Note: this option may not be used in conjunction with any other option. If this option is used with configuration parameters, those parameters will be silently ignored.
`-d, --adapter =<sfxge<N>>`	Performs the action on the identified Solarflare network adapter. The adapter identifier `sfxge` can be the adapter ID number, ifname or MAC address, as output by the `--list` option. If `--adapter` is not included, the action will apply to all installed Solarflare adapters.
`--clear`	Resets all options to their default values. If an adapter is specified, options for the given adapter are reset, but global options (shown in Table 73) are not reset. Note that `--clear` can also be used with parameters, allowing you to reset to default values, and then apply the parameters specified.

The following global parameters in Table 73 are used to control the configurable parameters for the Boot ROM driver when running prior to the operating system booting.

Table 73: Sfboot Global Parameters

Parameter	Description
`boot-image=` `all|optionrom|uefi|disabled`	Specifies which boot firmware images are served-up to the BIOS during start-up. This parameter can not be used if the `--adapter` option has been specified.

`port-mode=`
`default|1x10G|2x10G|4x10G|`
`2x40G`
	Configure the port mode to use. This is for SFN7000 series adapters only. The values specify the connectors available after using any splitter cables. The usable values are adapter-dependent:
	• SFN7*xx*2F: 1x10G, 2x10G (default)
	• SFN7*xx*4F: 2x10G, 4x10G (default)
	• SFN7*xx*2Q: 2x10G, 4x10G, 2x40G (default)
	Changes to this setting with `sfboot` require a cold reboot to become effective. MAC address assignments may change after altering this setting.

Table 73: Sfboot Global Parameters

Parameter	Description

`firmware-variant=`
`full-feature|ultra-low-latency|`
`capture-packed-stream|auto`
	Configure the firmware variant to use. This is for SFN7000 series adapters only:
	• the SFN7002F adapter is factory set to full-feature
	• all other adapters are factory set to auto.
	Default value = auto - means the driver will select a variant that meets its needs:
	• the VMware driver always uses full-feature
	• otherwise, ultra-low-latency is used.
	The ultra-low-latency variant produces best latency without support for TX VLAN insertion or RX VLAN stripping (not currently used features). It is recommended that Onload customers use the ultra-low-latency variant.

`insecure-filters=` `enabled|disabled`	If enabled bypass filter security on non-privileged functions. This is for SFN7000 series adapters only. This reduces security in virtualized environments. The default is disabled. When enabled a function (PF or VF) can insert filters not qualified by their own permanent MAC address. This is a requirement when using Onload or when using bonded interfaces.
`mac-spoofing=enabled|disabled`	If enabled, non-privileged functions can create unicast filters for MAC addresses that are not associated with them. This is for SFN7000 series adapters only. The default is disabled. Changes to this setting with `sfboot` require a cold reboot to become effective.
`rx-dc-size=8|16|32|64`	Specifies the size of the descriptor cache for each receive queue. This is for SFN7000 series adapters only. The default is: • 16 if the `port-mode` supports the maximum number of connectors for the adaptor • 32 if the `port-mode` supports a reduced number of connectors.

Table 73: Sfboot Global Parameters

Parameter	Description
`tx-dc-size=8|16|32|64`	Specifies the size of the descriptor cache for each transmit queue. This is for SFN7000 series adapters only. The default is: • 32 if the `port-mode` supports the maximum number of connectors for the adaptor • 64 if the `port-mode` supports a reduced number of connectors.
`vi-count=<vi count>`	Sets the total number of virtual interfaces that will be available on the NIC.

The following per-adapter parameters in Table 74 are used to control the configurable parameters for the Boot ROM driver when running prior to the operating system booting.

Table 74: Sfboot Per-adapter Parameters

Parameter	Description
`link-speed=auto|10g|1g|100m`	Specifies the network link speed of the adapter used by the Boot ROM. The default is `auto`. On the 10GBASE-T adapters, `auto` instructs the adapter to negotiate the highest speed supported in common with its link partner. On SFP+ adapters, `auto` instructs the adapter to use the highest link speed supported by the inserted SFP+ module. On 10GBASE-T and SFP+ adapters, any other value specified will fix the link at that speed, regardless of the capabilities of the link partner, which may result in an inability to establish the link. `auto` Auto-negotiate link speed (default) `10G` 10G bit/sec `1G` 1G bit/sec `100M` 100M bit/sec
`linkup-delay=` `<delay time in seconds>`	Specifies the delay (in seconds) the adapter defers its first connection attempt after booting, allowing time for the network to come up following a power failure or other restart. This can be used to wait for spanning tree protocol on a connected switch to unblock the switch port after the physical network link is established. The default is 5 seconds.

Table 74: Sfboot Per-adapter Parameters

Parameter	Description
`banner-delay=` `<delay time in seconds>`	Specifies the wait period for `Ctrl-B` to be pressed to enter adapter configuration tool. `<delay time in seconds>` = 0-256
`bootskip-delay=` `<delay time in seconds>`	Specifies the time allowed for `Esc` to be pressed to skip adapter booting. `<delay time in seconds>` = 0-256
`boot-type=pxe|iscsi|disabled`	Sets the adapter boot type – effective on next boot. `pxe` – PXE (Preboot eXecution Environment) booting `iscsi` – iSCSI (Internet Small Computer System Interface) booting `disabled` – Disable adapter booting

`initiator-dhcp=enabled|disabled`	Enables or disables DHCP address discovery for the adapter by the Boot ROM except for the Initiator IQN (see `initiator-iqn-dhcp`). This option is only valid if iSCSI booting is enabled (`boot-type=iscsi`).
	
	If initiator-DHCP is set to disabled, the following options will need to be specified:
	
	`initiator-ip=<IPv4 address>`
	`netmask=<IPv4 subnet mask>`
	
	The following options may also be needed:
	
	`gateway=<ip_address>`
	`primary-dns=<ip_address>`

`initiator-ip=<IPv4 address>`	Specifies the IPv4 address (in standard "." notation form) to be used by the adapter when `initiator-dhcp` is disabled. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot boot-type=iscsi initiator-dhcp=disabled initiator-ip=<192.168.1.3>`
`netmask=<IPv4 subnet mask>`	Specifies the IPv4 subnet mask (in standard "." notation form) to be used by the adapter when `initiator-dhcp` is disabled. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot boot-type=iscsi initiator-dhcp=disabled netmask=255.255.255.0`

Table 74: Sfboot Per-adapter Parameters

Parameter	Description
`gateway=<IPv4 address>`	Specifies the IPv4 subnet mask (in standard "." notation form) to be used by the adapter when `initiator-dhcp` is disabled. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot boot-type=iscsi initiator-dhcp=disabled gateway=192.168.0.10`
`primary-dns=<IPv4 address>`	Specifies the IPv4 address (in standard "." notation form) of the Primary DNS to be used by the adapter when `initiator-dhcp` is disabled. This option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot boot-type=iscsi initiator-dhcp=disabled primary-dns=192.168.0.3`
`initiator-iqn-dhcp= enabled|disabled`	Enables or disables use of DHCP for the initiator IQN only.
`initiator-iqn=<IQN>`	Specifies the IQN (iSCSI Qualified Name) to be used by the adapter when `initiator-iqn-dhcp` is disabled. The IQN is a symbolic name in the "." notation form; for example: iqn.2009.01.com.solarflare, and is a maximum of 223 characters long. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot initiator-iqn-dhcp=disabled initiator-`iqn=iqn.2009.01.com.solarflare adapter=2
`lun-retry-count=<retry count>`	Specifies the number of times the adapter attempts to access and login to the Logical Unit Number (LUN) on the iSCSI Target before failing. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot lun-retry-count=3`

Table 74: Sfboot Per-adapter Parameters

Parameter	Description

`target-dhcp=enabled|disabled`	Enables or disables the use of DHCP to discover iSCSI target parameters on the adapter.
	
	If `target-dhcp` is disabled, you must specify the following options:
	
	`target-server=<DNS name or IPv4 address>`
	`target-iqn=<IQN>`
	`target-port=<port number>`
	`target-lun=<LUN>`
	
	Example - Enable the use of DHCP to configure iSCSI Target settings:
	
	`sfboot boot-type=iscsi target-dhcp=enabled`

`target-server=` `<DNS name or IPv4 address>`	Specifies the iSCSI target's DNS name or IPv4 address to be used by the adapter when `target-dhcp` is disabled. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot boot-type=iscsi target-dhcp=disabled target-server=192.168.2.2`
`target-port=<port number>`	Specifies the Port number to be used by the iSCSI target when `target-dhcp` is disabled. The default Port number is Port 3260. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot boot-type=iscsi target-dhcp=disabled target-port=3262` This option should only be used if your target is using a non-standard TCP Port.
`target-lun=<LUN>`	Specifies the Logical Unit Number (LUN) to be used by the iSCSI target when `target-dhcp` is disabled. The default LUN is 0. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`).

Table 74: Sfboot Per-adapter Parameters

Parameter	Description
`target-iqn=<IQN>`	Specifies the IQN of the iSCSI target when `target-dhcp` is disabled. Maximum of 223 characters. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Note that if there are spaces contained in `<IQN>`, then the IQN must be wrapped in double quotes (“”). Example: `sfboot target-dhcp=disabled target-iqn=iqn.2009.01.com.solarflare adapter=2`
`vendor-id=<vendor identifier>`	Specifies the device vendor ID to be advertised to the DHCP server. This must match the vendor id configured at the DHCP server when using DHCP option 43 to obtain the iSCSI target.

`chap=enabled|disabled`	Enables or disables the use of Challenge Handshake Protocol (CHAP) to authenticate the iSCSI connection.
	Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`).
	To be valid, this option also requires the following sub-options to be specified:
	`username=<initiator username>`
	`secret=<initiator password>`
	Example:
	`sfboot boot-type=iscsi chap=enabled username=initiatorusername secret=initiatorsecret`

`username=<username>`	Specifies the CHAP initiator username (maximum 64 characters). Note that this option is required if either CHAP or Mutual CHAP is enabled (`chap=enabled, mutual-chap=enabled`). Note that if there are spaces contained in `<username>`, then it must be wrapped in double quotes (“”). Example: `sfboot boot-type=iscsi chap=enabled username=username`

Table 74: Sfboot Per-adapter Parameters

Parameter	Description
`secret=<secret>`	Specifies the CHAP initiator secret (minimum 12 characters, maximum 20 characters). Note that this option is valid if either CHAP or Mutual CHAP is enabled (chap=enabled, `mutual-chap=enabled`). Note that if there are spaces contained in `<secret>`, then it must be wrapped in double quotes (“”). Example: `sfboot boot-type=iscsi chap=enabled username=username secret=veryverysecret`

`mutual-chap=enabled|disabled`	Enables/disables Mutual CHAP authentication when iSCSI booting is enabled.
	
	This option also requires the following sub-options to be specified:
	
	`target-username=<username>`
	`target-secret=<password>`
	`username=<username>`
	`secret=<password>`
	
	Example:
	
	`sfboot boot-type=iscsi mutual-chap=enabled username=username secret=veryverysecret target-username=targetusername target-secret=anothersecret`

`target-username=<username>`	Specifies the username that has been configured on the iSCSI target (maximum 64 characters). Note that this option is necessary if Mutual CHAP is enabled on the adapter (`mutual-chap=enabled`). Note that if there are spaces contained in `<username>`, then it must be wrapped in double quotes (“”).
`target-secret=<secret>`	Specifies the secret that has been configured on the iSCSi target (minimum 12 characters; maximum 20 characters). Note: This option is necessary if Mutual CHAP is enabled on the adapter (`mutual-chap=enabled`). Note that if there are spaces contained in `<secret>`, then it must be wrapped in double quotes (“”).
`mpio-priority=<MPIO priority>`	Specifies the Multipath I/O (MPIO) priority for the adapter. This option is only valid for iSCSI booting over multi-port adapters, where it can be used to establish adapter port priority. The range is 1- 255, with 1 being the highest priority.

Table 74: Sfboot Per-adapter Parameters

Parameter	Description
`mpio-attempts=<attempt count>`	Specifies the number of times MPIO will try and use each port in turn to login to the iSCSI target before failing.
`pf-count=<pf count>`	This is the number of available PCIe PFs per physical network port. This setting is applied to all ports on the adapter. Changes to this setting with `sfboot` require a cold reboot to become effective. MAC address assignments may change after altering this setting.
`msix-limit=` `8|16|32|64|128|256|512|1024`	Specifies the maximum number of MSI-X interrupts that each PF will use. The default is 32. Note: Using the incorrect setting can impact the performance of the adapter. Contact Solarflare technical support before changing this setting.
`sriov=enabled|disabled`	Enable SR-IOV support for operating systems that support this. Not required on SFN7000 series adapters.
`vf-count=<vf count>`	The number of virtual functions (VF) advertised to the operating system for each Physical Function on this physical network port. SFN7000 series adapters have a total limit of 2048 interrupts. Earlier adapters support a total limit of 127 virtual functions per port and a total of 1024 interrupts. Depending on the values of msix-limit and vf-msix-limit, some of these virtual functions may not be configured. Enabling all 127 VFs per port with more than one MSI-X interrupt per VF may not be supported by the host BIOS - in which case you may get 127 VFs on one port and none on others. Contact your BIOS vendor or reduce the VF count. The sriov parameter is implied if vf-count is greater than zero. Changes to this setting with `sfboot` require a cold reboot to become effective.
`vf-msix-limit=` `1|2|4|8|16|32|64|128|256`	The maximum number of interrupts a virtual function may use.
`pf-vlans=<tag>[,<tag>[,...]]|none`	Comma separated list of VLAN tags for each PF in the range 0-4094 - see sfboot --help for details.

Table 74: Sfboot Per-adapter Parameters

Parameter	Description
`switch-mode= default|sriov|partitioning| partitioning-with-sriov|pfiov`	Specifies the mode of operation that the port will be used in: `default` - single PF created, zero VFs created. `sriov` - SR-IOV enabled, single PF created, VFs configured with `vf-count`. `partitioning` - PFs configured with `pf-count`, VFs configured with `vf-count`. See NIC Partitioning on page 58 for details. `partitioning-with-sriov` - SR-IOV enabled, PFs configured with `pf-count`, VFs configured with `vf-count`. See NIC Partitioning on page 58 for details. `pfiov` - PFIOV enabled, PFs configured with `pf-count`, VFs not supported. Changes to this setting with `sfboot` require a cold reboot to become effective.

Sfboot: Examples

- Show the current boot configuration for all adapters:

  `sfboot`

```
Solarflare boot configuration utility [v3.0.5]
Copyright Solarflare Communications 2006-2010, Level 5 Networks 2002-2005

sfxge0:
  Boot image                       Disabled
  MSI-X interrupt limit            32

sfxge1:
  Boot image                       Disabled
  MSI-X interrupt limit            32
```

- List all Solarflare adapters installed on the localhost:

  `sfboot --list`

```
Solarflare boot configuration utility [v3.0.5]
Copyright Solarflare Communications 2006-2010, Level 5 Networks 2002-2005
sfxge0 - 00-0F-53-01-38-40
sfxge1 - 00-0F-53-01-38-41
```

6.10 Upgrading Adapter Firmware with sfupdate

To Update Adapter Firmware

As a root user enter:

```
pkgadd -d SFCutils_i386_v<version>.pkg
```

Once installed, by default, utility tools are located in the `/opt/SFCutils/bin` directory.

Sfupdate: Command Usage

The general usage for sfupdate is as follows (as root):

```
sfupdate [--adapter=sfxge<x>] [options]
```

where:

- `sfgxe<x>` is the interface name of the Solarflare adapter you want to upgrade.
- `option` is one of the command options listed in Table 75.

The format for the options are:

```
--<option>=<parameter>
```

Running the command `sfupdate` with no additional parameters will show the current firmware version for all Solarflare adapters and whether the firmware within sfupdate is more up to date. To update the firmware for all Solarflare adapters run the command `sfupdate --write`

Solarflare recommend that you use sfupdate in the following way:

1 Run `sfupdate` to check that the firmware on all your adapters are up to date.

2 Run `sfupdate --write` to update the firmware on all adapters.

Sfupdate: Command Line Options

Table 75 lists the options for `sfupdate`.

Table 75: Sfupdate Options

Option	Description
`-h, --help`	Shows help for the available options and command line syntax.
`-v, --verbose`	Enable verbose output mode.
`-s, --silent`	Suppress all output except for errors. Useful for scripts.
`-V, --version`	Display version information and exit.
`-i, --adapter=sfxge<x>`	Specifies the target adapter when more than one adapter is installed in the machine. `sfxge<x>` = Adapter ifname or MAC address (as obtained with `--list`).
`--list`	Shows the adapter ID, adapter name and MAC address of each adapter installed in the machine.
`--write`	Re-writes the firmware from the images embedded in the `sfupdate` tool. To re-write using an external image, specify `--image=<filename>` in the command. `--write` fails if the embedded image is the same or a previous version. To force a write in this case, specify the option `--force`.
`--force`	Force update of all firmware, even if the installed firmware version is the same or more recent than the images embedded in the utility.
`--image=(filename)`	Update the firmware using the image contained in the specified file, rather than the image embedded in the utility. Use with the `--write` and, if needed, `--force` options.
`-y, --yes`	Prompts for user confirmation before re-writing the firmware.

Sfupdate: Examples

- Display firmware versions for all adapters:

  `sfupdate`

```
sfupdate: Solarflare Firmware Update Utility [v3.0.5.2164]
Copyright Solarflare Communications 2006-2010, Level 5 Networks 2002-2005
Network adapter driver version: v3.0.5.2163

sfxge0 - MAC: 00:0F:53:01:38:90
    Firmware   version: v3.0.5
    Boot ROM   version: v3.0.5.2163
    PHY        version: v2.0.2.5
    Controller version: v3.0.5.2161

The Boot ROM firmware is up to date
The PHY firmware is up to date
The image contains a more recent version of the Controller [v3.0.5.2163]
vs [v3.0.5.2161]
Use the -w|--write option to perform an update

sfxge1 - MAC: 00:0F:53:01:38:91
    Firmware   version: v3.0.5
    Boot ROM   version: v3.0.5.2163
    PHY        version: v2.0.2.5
    Controller version: v3.0.5.2161

The Boot ROM firmware is up to date
The PHY firmware is up to date
The image contains a more recent version of the Controller [v3.0.5.2163]
vs [v3.0.5.2161]
Use the -w|--write option to perform an update
```


6.11 Performance Tuning on Solaris

Introduction

The Solarflare family of network adapters are designed for high-performance network applications. The adapter driver is pre-configured with default performance settings that have been chosen to give good performance across a broad class of applications. In many cases, application performance can be improved by tuning these settings to best suit the application.

There are three metrics that should be considered when tuning an adapter:

- Throughput
- Latency
- CPU utilization

Different applications may be more or less affected by improvements in these three metrics. For example, transactional (request-response) network applications can be very sensitive to latency whereas bulk data transfer applications are likely to be more dependent on throughput.

The purpose of this guide is to highlight adapter driver settings that affect the performance metrics described. This guide covers the tuning of all of the Solarflare family of adapters.

In addition to this guide, you may need to consider other issues influencing performance such as application settings, server motherboard chipset, additional software installed on the system, such as a firewall, and the specification and configuration of the LAN. Consideration of such issues is not within the scope of this guide.

Tuning settings

Port mode

The selected port mode for SFN7000 series adapters should correspond to the speed and number of connectors in use, after using any splitter cables. If a restricted set of connectors is configured, the driver can then transfer resources from the unused connectors to those configured, potentially improving performance.

Adapter MTU (Maximum Transmission Unit)

The default MTU of 1500 bytes ensures that the adapter is compatible with legacy 10/100Mbps Ethernet endpoints. However if a larger MTU is used, adapter throughput and CPU utilization can be improved. CPU utilization is improved because it takes fewer packets to send and receive the same amount of data. Solarflare adapters support frame sizes up to 9000 bytes (this does not include the Ethernet preamble or frame-CRC).

Since the MTU should ideally be matched across all endpoints in the same LAN (VLAN), and since the LAN switch infrastructure must be able to forward such packets, the decision to deploy a larger than default MTU requires careful consideration. It is recommended that experimentation with MTU be done in a controlled test environment.

The MTU can be changed dynamically using `ifconfig` if the maximum MTU size has been set in the `sxfge_conf` file (see Configuring Jumbo Frames on page 297), where `sfxge<X>` is the interface name and size is the MTU size in bytes:

```
$ ifconfig sfxge<X> mtu <size>
```

Verification of the MTU setting may be performed by running `$ ifconfig sfxge<X>` with no options and checking the MTU value associated with the interface. If you want to have an MTU configure when the interface is brought up, add `mtu` to the single line of configuration data in:

```
/etc/hostname.sfxge<X>
```

For example:

```
[<IP address>] mtu <size>
```

Interrupt Moderation (Interrupt Coalescing)

Interrupt moderation controls the number of interrupts generated by the adapter by adjusting the extent to which receive packet processing events are coalesced. Interrupt moderation may coalesce more than one packet-reception or transmit-completion event into a single interrupt.

This parameter is critical for tuning adapter latency. Increasing the moderation value will increase latency, but reduce CPU utilization and improve peak throughput, if the CPU is fully utilized. Decreasing the moderation value or turning it off will decrease latency at the expense of CPU utilization and peak throughput. However, for many transaction request-response type network applications, the benefit of reduced latency to overall application performance can be considerable. Such benefits may outweigh the cost of increased CPU utilization.

NOTE: The interrupt moderation time dictates the minimum gap between two consecutive interrupts. It does not mandate a delay on the triggering of an interrupt on the reception of every packet. For example, an interrupt moderation setting of 30µs will not delay the reception of the first packet received, but the interrupt for any following packets will be delayed until 30µs after the reception of that first packet.

TCP/IP Checksum Offload

Checksum offload moves calculation and verification of IP Header, TCP and UDP packet checksums to the adapter. The driver by default has all checksum offload features enabled. Therefore, there is no opportunity to improve performance from the default.

TCP Segmentation Offload (TSO)

TCP Segmentation offload (TSO) offloads the splitting of outgoing TCP data into packets to the adapter. TCP segmentation offload benefits applications using TCP. Non TCP protocol applications will not benefit (but will not suffer) from TSO.

The Solaris TCP/IP stack provides a large TCP segment to the driver, which splits the data into MSS size, each with adjusted sequence space and a hardware calculated checksum.

TCP Large Receive Offload (LRO) / RX Coalescing

LRO (called rx coalescing on Solaris) is a feature whereby the adapter coalesces multiple packets received on a TCP connection into a single call to the operating system TCP Stack. This reduces CPU utilization, and so improves peak throughput when the CPU is fully utilized.

LRO should not be enabled if you are using the host to forward packets from one interface to another; for example if the host is performing IP routing or acting as a layer2 bridge. The driver has LRO disabled by default.

TCP Protocol Tuning

TCP Performance can also be improved by tuning kernel TCP settings. Settings include adjusting send and receive buffer sizes, connection backlog, congestion control, etc.

The transmit and receive buffer sizes may already be explicitly controlled by an application calling `setsockopt()` to set SO_SNDBUF or SO_RCVBUF.

```
ndd -set /dev/tcp tcp_xmit_hiwat 524288
ndd -set /dev/tcp tcp_recv_hiwat 524288
```

The following settings may help if there are a large number of connections being made:

```
ndd -set /dev/tcp tcp_time_wait_interval 1000 # 1 sec for time-wait (min)
ndd -set /dev/tcp tcp_conn_req_max_q 4096 # increase accept queue
ndd -set /dev/tcp tcp_conn_req_max_q0 4096 # increase accept queue
ndd -set /dev/tcp tcp_conn_req_min 1024 # increase minimum accept queue
ndd -set /dev/tcp tcp_rst_sent_rate_enabled 0 # disable rst rate limiting
```

See the Internet Protocol Suite Tunable Parameters chapter of the Tunable Parameters Reference Guide for more details:

http://docs.oracle.com/cd/E19082-01/819-2724/6n50b07lr/index.html

NOTE: You can also add these settings to `/etc/system`

Receive Side Scaling (RSS)

Solarflare adapters support Receive Side Scaling (RSS). RSS enables packet receive-processing to scale with the number of available CPU cores. RSS requires a platform that supports MSI-X interrupts.

When RSS is enabled the controller uses multiple receive queues into which to deliver incoming packets. The receive queue selected for an incoming packet is chosen in such a way as to ensure that packets within a TCP stream are all sent to the same receive queue – this ensures that packet-ordering within each stream is maintained. Each receive queue has its own dedicated MSI-X interrupt which ideally should be tied to a dedicated CPU core. This allows the receive side TCP processing to be distributed amongst the available CPU cores, providing a considerable performance advantage over a conventional adapter architecture in which all received packets for a given interface are processed by just one CPU core.

RSS is enabled by default in the sfxge driver. To limit or disable RSS:

Uncomment the following line in: `/kernel/drv/sfxge.conf`

```
rx_scale_count=<number of MSI-X interrupts requested>
```

Limitations of Solaris MSI-X interrupt allocation are:

- All network drivers share 32 MSI-X interrupts.
- A single NIC can only use 2 MSI-X interrupts (this restriction can be lifted with the `ddi_msix_alloc_limit` setting below).

To lift the restriction of 2 MSI-X interrupts, add the following line to `/etc/system` and reboot.

```
set ddi_msix_alloc_limit=8
```

If no MSI/MSI-X interrupts are available then the driver will fall-back to use a single legacy interrupt. RSS will be unavailable for that port.

Other RSS Settings

You should add the following lines to `/etc/system`:

```
set pcplusmp:apic_intr_policy=1
```

This sets the interrupt distribution method to round robin.

```
set ip:ip_squeue_fanout=1
```

This determines the mode for associating TCP/IP connections with queues. For details refer to the following:

http://docs.oracle.com/cd/E19082-01/819-2724/chapter4-7/index.html

NOTE: RSS also works for UDP packets. For UDP traffic the Solarflare adapter will select the Receive CPU based on IP source and destination addresses. SFN5xxx adapters support IPv4 and IPv6 RSS, while SFN4xxx adapters support just IPv4 RSS.

Other Considerations

PCI Express Lane Configurations

The PCI Express (PCIe) interface used to connect the adapter to the server can function at different widths. This is independent of the physical slot size used to connect the adapter. The possible widths are multiples x1, x2, x4, x8 and x16 lanes of (2.5Gbps for PCIe Gen 1, 5.0 Gbps for PCIe Gen 2) in each direction. Solarflare Adapters are designed for x8 lane operation.

On some server motherboards, choice of PCIe slot is important. This is because some slots (including ones that are physically x8 or x16 lanes) may only electrically support x4 lanes. In x4 lane slots, Solarflare PCIe adapters will continue to operate, but not at full speed. The Solarflare driver will warn you if it detects the adapter is plugged into a PCIe slot which electrically has fewer than x8 lanes.

For SFN5xxxx adapters, which require a PCIe Gen 2 slot for optimal operation, a warning will be given if they are installed in a PCIe Gen 1 slot.

CPU Power Management

This feature monitors CPU utilization and lowers the CPU frequency when utilization is low. This reduces the power consumption of the CPU. For latency sensitive applications, where the application switches between having packets to process and having periods of idle time waiting to receive a packet, dynamic clock speed control may increase packet latencies. There therefore can be a benefit to disabling the service.

The service can be disabled temporarily with the configuration in `/etc/power.conf` file and restarting the service. For example:

```
cpupm disable
system-threshold always-on
cpu-threshold always-on
cpu_deep_idle disable
```

The service can be disabled across reboots with:

```
svcadm disable  svc:/system/power:default
```

See http://docs.oracle.com/cd/E19253-01/817-0547/gfgmu/index.html

Memory bandwidth

Many chipsets use multiple channels to access main system memory. Maximum memory performance is only achieved when the chipset can make use of all channels simultaneously. This should be taken into account when selecting the number of DIMMs to populate in the server. Consult the motherboard documentation for details.

Server Motherboard, Server BIOS, Chipset Drivers

Tuning or enabling other system capabilities may further enhance adapter performance. Readers should consult their server user guide. Possible opportunities include tuning PCIe memory controller (PCIe Latency Timer setting available in some BIOS versions).

Tuning Recommendations

The following tables provide recommendations for tuning settings for different applications.

Throughput - Table 76

Latency - Table 77

Forwarding - Table 78

Recommended Throughput Tuning

Table 76: Throughput Tuning Settings

Tuning Parameter	How?
MTU Size to maximum supported by network	`ifconfig sfxge<x> mtu <size>`
Interrupt moderation	Leave at default
TCP/IP Checksum Offload	Leave at default
TCP Segmentation Offload	Leave at default
TCP Large Receive Offload	Leave at default
TCP Protocol Tuning	Leave at default
Receive Side Scaling (RSS)	Application dependent
Buffer Allocation Method	Leave at default. Some applications may benefit from specific setting.
PCI Express Lane Configuration	Ensure current speed (not the supported speed) reads back as “x8 and 5Gb/s” Or “x8 and Unknown”
CPU Power Management	Leave enabled
Memory bandwidth	Ensure memory utilizes all memory channels on system motherboard

Recommended Latency Tuning

Table 77: Latency Tuning Settings

Tuning Parameter	How?
MTU Size to maximum supported by network	Leave at default
Interrupt moderation	Disable with: `sfxge.conf`
TCP/IP Checksum Offload	Leave at default
TCP Segmentation Offload	Leave at default
TCP Large Receive Offload	Leave at default
TCP Protocol Tuning	Leave at default, but changing does not impact latency
Receive Side Scaling	Application dependent
Buffer Allocation Method	Leave at default
PCI Express Lane Configuration	Ensure current speed (not the supported speed) reads back as “x8 and 5Gb/s” Or “x8 and Unknown”
CPU Power Management	Disable with: `/etc/power.conf`
Memory bandwidth	Ensure memory utilizes all memory channels on system motherboard

Recommended Forwarding Tuning

Table 78: Forwarding Tuning Settings

Tuning Parameter	How?
MTU Size to maximum supported by network	`ifconfig sfxge<X> mtu <size>`
Interrupt moderation	Leave at default
TCP/IP Checksum Offload	Leave at default
TCP Segmentation Offload	Leave at default
TCP Large Receive Offload	Leave at default
TCP Protocol Tuning	Can leave at default
Receive Side Scaling (RSS)	`sfxge.conf`

Table 78: Forwarding Tuning Settings

Tuning Parameter	How?
Buffer Allocation Method	Leave at default
PCI Express Lane Configuration	Ensure current speed (not the supported speed) reads back as “x8 and 5Gb/s” Or “x8 and Unknown”
CPU Speed service (cpuspeed)	Leave enabled
Memory bandwidth	Ensure memory utilizes all memory channels on system motherboard

6.12 Module Parameters

The normal syntax when using parameters is `param=<value>`.

Table 79 lists the available parameters in the Solarflare Solaris driver module (`sfxge.conf`):

Table 79: Driver Module Parameters

Parameter	Description	Values	Default
rx_scale_count	Maximum number of RSS channels to use per port. The actual number may be lower due to availability of MSI-X interrupts. There is a maximum of 32 MSI-X interrupts across all network devices. To use more than 2 MSI-X interrupts you need to add e.g. `set ddi_msix_alloc_limit=8` in `/etc/system`.		128
rx_coalesce_mode	Coalesce RX packets (Large Receive Offload).	0 = off 1 = on 2 = on, respecting TCP PSH boundaries.	0
intr_moderation	Interrupt moderation in µs. Decreasing this reduces latency but increases interrupt rate and therefore CPU usage.		30
mtu	Maximum MTU of an sfxge interface in bytes (excludes ethernet framing)	1500 - 9000	1500

Table 79: Driver Module Parameters

Parameter	Description	Values	Default
rxq_size	Number of descriptors in the receive queue descriptor ring	512 - 4096 must be a power of 2 value	1024
rx_pkt_mem_max	Per port memory limit for receive packet buffers		64Mb
action_on_hw_err	Controls the action taken on hardware error.	0 - recover adapter to a working state. 1- do not advertise to the kernel that the link is down during the reset. 2. reset the hardware, but do not use it again - useful for failover mechanisms to ensure this adapter does not become the active link again.	0
rx_prealloc_pkt_buffers	Number of pkt buffers to allocate at start of a receive queue and maintain a free pkt pool of at least this many buffers.	limited by available system memory.	512

6.13 Kernel and Network Adapter Statistics

Statistical data originating from the MAC on Solarflare network adapters can be gathered using the Solaris Kernel Statistics command (`kstat`). The following tables identify kernel and adapter statistics returned from the `kstat` command:

kstat -m sfxge

To read individual classes use the -c option or by name use the -n option, for example:

kstat -m sfxge -c net

kstat -m sfxge -n mac

The kstats statistics are listed by 'names' in the following tables

Table 80: Name mac

Parameter	Description
adv_cap_1000fdx	Advertise 1000 Mbps full duplex capacity 1 = true, 0 = false.
adv_cap_1000hdx	Advertise 1000 Mbps half duplex capacity 1 = true, 0 = false.
adv_cap_100fdx	Advertise 100 Mbps full duplex capacity 1 = true, 0 = false.
adv_cap_100hdx	Advertise 100 Mbps half duplex capacity 1 = true, 0 = false.
adv_cap_10fdx	Advertise 10 Mbps full duplex capacity 1 = true, 0 = false.
adv_cap_10hdx	Advertise 10 Mbps half duplex capacity 1 = true, 0 = false.
adv_cap_asmpause	Advertise asymmetric pause capability 1 = true, 0 = false.
adv_cap_autoneg	Advertise auto-negotiation capability when auto-negotiation is enabled. When set to zero, the highest priority speed and duplex mode is used for forced mode.

Table 80: Name mac

Parameter	Description
adv_cap_pause	Depends on the value of adv_cap_asmpause. If adv_cap_asmpause = 1 then: 1 = send pause frames when there is receive congestion. 0 = pause transmission when a pause frame is received. If adv_cap_asmpause = 0 then: 1 = send pause frames when there is receive congestion and pause transmission when a pause frame is received. 0 = pause capability is not available in either direction.
align_errors	Number of occurrences of frame alignment error.
brdcstrcv	Number of broadcast packets received.
brdcstxmt	Number of broadcast packets transmitted.
cap_1000fdx	Capable of 1000 Mbps full duplex 1 = true, 0 = false.
cap_1000hdx	Capable of 1000 Mbps half duplex 1 = true, 0 = false.
cap_100fdx	Capable of 100 Mbps full duplex 1 = true, 0 = false.
cap_100hdx	Capable of 100 Mbps half duplex 1 = true, 0 = false.
cap_10fdx	Capable of 10 Mbps full duplex 1 = true, 0 = false.
cap_10hdx	Capable of 10 Mbps half duplex 1 = true, 0 = false.
cap_asmpause	Asymmetric pause capability 1 = true, 0 = false. This determines action taken by the cap_pause parameter - see below.
cap_autoneg	Capable of auto negotiation 1= true, 0 = false.

Table 80: Name mac

Parameter	Description
cap_pause	Direction depends on value of cap_asmpause. If cap_asmpause = 1 then: 0 = pause transmission when a pause frame is received. 1 = send pause frames when there is receive congestion. If cap_asmpause = 0 then: 0 = pause capability is not available in either direction. 1 = send pause frames when there is receive congestion and pause transmission when a pause frame is received.
collisions	Number of collisions detected when attempting to send.
crtime	Timestamp when samples were taken.
defer_xmts	Number of packets successfully transmitted after the network adapter defers transmission at least once when the medium is busy.
ex_collsions	Number of packets not transmitted due to excessive collisions which can occur on networks under heavy load or when too many devices contend for the collision domain. After 15 retransmission attempts plus the original transmission attempt the counter is incremented and the packet is dropped.
fcs_errors	Number of frames received which FCS errors.
first_collisions	0
ierrors	0
ifspeed	Adapter interface speed.
ipackets	Number of packets received (32 bit counter).
ipackets64	Number of packets received (64 bit counter).

Table 80: Name mac

Parameter	Description
link_asmpause	When adv_*pause_cap and lp_*pause_cap are compared following auto-negotiation, the flow control mechanism for the link depends on what is most meaningful: 0 = flow control in both directions when link_pause is set to one. 1 = flow control in one direction.
link_autoneg	Auto-negotiation, 0 = not enabled, 1 = enabled.
link_duplex	0 = down, 1 = half duplex, 2 = full duplex.
link_pause	Depends on link_asmpause. If link_asmpause = 1 then: 1 = flow control in both directions is available. 0 = no flow control on the link. If link_asmpause = 0 then: 1 = the local host will honor received pause frames by temporarily suspending transmission of further frames. 0 = in the event of receive congestion, the local host will transmit pause frames to the peer.
link_state	link status 0 = link down, 1 = link up.
link_up	1 =link is up, 0 = link is down.
lp_cap_1000fdx	Link partner advertises 1000 Mbps full duplex capability.
lp_cap_1000hdx	Link partner advertises 1000 Mbps half duplex capability.
lp_cap_100fdx	Link partner advertises 100 Mbps full duplex capability.
lp_cap_100hdx	Link partner advertises 100 Mbps half duplex capability.
lp_cap_10fdx	Link partner advertises 10 Mbps full duplex capability.
lp_cap_10hdx	Link partner advertises 10 Mbps half duplex capability.
lp_cap_asmpause	Asymmetric pause capability. 1 = true, 0 = false. This determines action taken by the lp_cap_pause parameter - see below.

Table 80: Name mac

Parameter	Description
lp_cap_autoneg	Link partner advertises auto-negotiation capability.
lp_cap_pause	Depends on value of lp_cap_asmpause. If lp_cap_asmpause = 1 then: 1 = send pause frames when there is receive congestion. 0 = pause transmission when a pause frame is received. If lp_cap_asmpause = 0 then: 1 = send pause frames when there is receive congestion and pause transmission when a pause frame is received. 0 = pause capability is not available in either direction.
macrcv_errors	Count the number of frames for which reception on a particular interface fails due to internal MAC error. Does not include too long frames, alignment error frames or FCS errors.
macxmt_errors	Count the number of frames for which transmission fails due to internal MAC error. Does not include failures due to late collisions, excessive collisions or carrier sense errors.
multi_collisions	Count the number of frames for which transmission fails due to multiple collisions.
multircv	Number of multicast packets received.
multixmt	Number of multicast packets transmitted.
norcvbuf	0
noxmtbuf	0
obytes	Number of bytes output (32 bit counter).
obytes64	Number of bytes output (64 bit counter).
oerrors	Number of outbound packets not transmitted due to error.
opackets	Number of outbound packets (32 bit counter).
opackets64	Number of outbound packets (64 bit counter)
promisc	Promiscuous Mode, 0 = not enabled, 1 = enabled

Table 80: Name mac

Parameter	Description
rbytes	Number of bytes received (32 bit counter)
rbytes64	Number of bytes received (64 bit counter)
snaptime	178761.398854604
sqe_errors	Count of number of times the SQE_TEST_ERROR message is generated for an interface.
toolong_errors	Count the number of frames received that exceed the maximum permitted frame size.
tx_late_collsions	A sending device may detect a collision as it attempts to transmit a frame or before it completes sending the entire frame. If a collision is detected after the device has completed sending the entire frame, the device will assume that the collision occurred because of a different frame. Late collisions can occur if the length of the network segment is greater than the standard allowed length. Collision occurred beyond the collision window (512 bit times). This should always be zero as Solarflare adapters operate in full duplex mode.
unknowns	0
xcvr_addr	MII address in the 0-31 range of the physical layer device in use for a given Ethernet device.
xcvr_id	MII transceiver manufacturer and device ID.
xcvr_inuse	MII transceiver type: 0 = undefined 1 = none, MII present but nothing connected 2 = 10 Mbps Manchester encoding 3 = 100BaseT4, 100 Mbps 8B/6T 4 = 100BaseX, 100 Mbps 4B/5B 5 = 100BaseT2 100 Mbps PAM5X5 6 = 1000BaseT, 1000 Mbps 4D-PAM5

Table 81: Name sfxge_cfg

Parameter	Description
crtime	Timestamp when samples were taken.
mac	Adapter hardware address.
version	Solarflare sfxge driver version

Table 82: Name sfxge_mac

Parameter	Description
crtime	Timestamp when samples were taken.
link_duplex	0 = down, 1 = half duplex, 2 = full duplex.
link_speed	10000 (Mbps).
link_up	1 = link is up, 0 = link is down.
rx_1024_to_15xx_pkts	Number of packets received where the length is between 1024 and 15xx bytes. 1518(non VLAN), 1522(VLAN).
rx_128_to_255_pkts	Number of packets received where the length is between 128 and 255 bytes.
rx_256_to_511_pkts	Number of packets received where the length is between 256 and 511 bytes.
rx_512_to_1023_pkts	Number of packets received where the length is between 512 and 1023 bytes.
rx_65_to_127_pkts	Number of packets received where the length is between 65 and 127 bytes.
rx_align_errors	Number of occurrences of frame alignment error.
rx_brdcst_pkts	Number of broadcast packets received.
rx_drop_events	Number of packets dropped by adapter driver.
rx_errors	Number of packet received with bad FCS.
rx_false_carrier_errors	Count of the instances of false carrier detected. False carrier is activity on the receive channel that does not result in a packet receive attempt being made.
rx_fcs_errors	Number of packets received with FCS errors - these are dropped by the Solarflare driver.
rx_ge_15xx_pkts	Number of packets received with payload size greater than 1518 bytes (1522 bytes VLAN).

Table 82: Name sfxge_mac

Parameter	Description
rx_internal_errors	Number of frames that could not be received due to a MAC internal error condition. e.g. frames not received by the MAC due to FIFO overflow condition.
rx_lane0_char_err	0
rx_lane0_disp_err	0
rx_lane1_char_err	0
rx_lane1_disp_err	0
rx_lane2_char_err	0
rx_lane2_disp_err	0
rx_lane3_char_err	0
rx_lane3_disp_err	0
rx_le_64_pkts	Number of packets received where the length is exactly 64 bytes.
rx_match_fault	Number of packets received which did not match a filter.
rx_multicst_pkts	Number of multicast packets received.
rx_nodesc_drop_cnt	Number of packets dropped by the network adapter because of a lack of RX descriptors in the RX queue. Packets can be dropped by the NIC when there are insufficient RX descriptors in the RX queue to allocate to the packet. This problem can occur if the receive rate is very high and the network adapter is not able to allocate memory and refill the RX descriptor ring quickly enough to keep up with the incoming packet rate. A number of different steps can be tried to resolve this issue: 1. Disable the irqbalance daemon in the OS. 2. Distribute the traffic load across the available CPU/ cores by setting rss_cpus=cores. Refer to Receive Side Scaling section. 3. Increase receive queue size using ethtool.
rx_octets	Total number of octets received.

Table 82: Name sfxge_mac

Parameter	Description
rx_pause_pkts	Number of pause packets received with valid pause op_code.
rx_pkts	Total number of packets received.
rx_symbol_errors	Count of the number of times the receiving media is non-idle (the time between the Start of Packet Delimiter and the End of Packet Delimiter) for a period of time equal to or greater than minimum frame size, and during which there was at least one occurrence of an event that causes the PHY to indicate Receive Error on the MII.
rx_unicst_pkts	Number of unicast packets received.
tx_1024_to_15xx_pkts	Number of packets transmitted where the length is between 1024 and 15xx bytes. 1518(non VLAN), 1522(VLAN).
tx_128_to_255_pkts	Number of packets transmitted where the length is between 128 and 255 bytes.
tx_256_to_511_pkts	Number of packets transmitted where the length is between 256 and 511 bytes.
tx_512_to_1023_pkts	Number of packets transmitted where the length is between 512 and 1023 bytes.
tx_65_to_127_pkts	Number of packets transmitted where the length is between 65 and 127 bytes.
tx_brdcst_pkts	Number of broadcast packets transmitted.
tx_def_pkts	The number of packets successfully transmitted after the network adapter defers transmission at least once when the medium is busy.
tx_errors	Number of packets transmitted with incorrect FCS.
tx_ex_col_pkts	Number of packets not transmitted due to excessive collisions. Excessive collisions occur on a network under heavy load or when too many devices contend for the collision domain. After 15 retransmission attempts + the original transmission attempt the counter is incremented and the packet is discarded.
tx_ex_def_pkts	Number of packets for which transmission is deferred for an excessive period of time.

Table 82: Name sfxge_mac

Parameter	Description
tx_ge_15xx_pkts	Number of packets transmitted where length is between 15xx and 9000 bytes. 1518(non VLAN), 1522(VLAN).
tx_late_col_pkts	A sending device may detect a collision as it attempts to transmit a frame or before it completes sending the entire frame. If a collision is detected after the device has completed sending the entire frame, the device will assume that the collision occurred because of a different frame. Late collisions can occur if the length of the network segment is greater than the standard allowed length. Collision occurred beyond the collision window (512 bit times). This should always be zero as Solarflare adapters operate in full duplex mode.
tx_le_64_pkts	Number of frames transmitted where the length is less than 64 bytes.
tx_mult_col_pkts	Number of packets transmitted after being subject to multiple collisions.
tx_multicst_pkts	Number of multicast packets transmitted. Includes flow control packets.
tx_octets	Number of octets transmitted.
tx_pause_pkts	Number of pause packets transmitted.
tx_pkts	Number of packets transmitted.
tx_sgl_col_pkts	Number of occurrences when a single collision delayed immediate transmission of a packet.
tx_unicst_pkts	Number of unicast packets transmitted. Includes packets that exceed that maximum length.

Table 83: Name sfxge_ndd

Parameter	Description
adv_cap_1000fdx	Refer to the corresponding field in the MAC statistics in Table 80 above. The adv_cap_* parameters represent a mirror image of the mac adv_*_cap parameter list for an Ethernet device. The parameters are also a subset of the cap_* statistics. If the cap_* value is 0, the corresponding adv_cap_* must also be 0, except for adv_cap_asmpause and adv_cap_pause parameters.
adv_cap_1000hdx	
adv_cap_100fdx	
adv_cap_100hdx	
adv_cap_10fdx	
adv_cap_10gfdx	
adv_cap_10hdx	
adv_cap_asm_pause	
adv_cap_autoneg	
adv_cap_pause	
cap_1000fdx	
cap_1000hdx	
cap_100fdx	
cap_100hdx	
cap_10fdx	
cap_10gfdx	
cap_10hdx	
cap_asm_pause	
cap_autoneg	
cap_pause	
crtime	Timestamp when samples were taken.
fcntl_generate	Flow control. When 1 generate pause frames.
fcntl_respond	Flow control -When 1 - pause transmission on receipt of pause frames.
intr_moderation	Interrupt moderation interval (microseconds) maximum value 20000 µs

Table 83: Name sfxge_ndd

Parameter	Description
lp_cap_1000fdx	Refer to the corresponding link partner field in the MAC statistics in Table 80 above. The adv_cap_* parameters represent a mirror image of the mac adv_*_cap parameter list for an Ethernet device. The parameters are also a subset of the cap_* statistics. If the cap_* value is 0, the corresponding adv_cap_* must also be 0, except for adv_cap_asmpause and adv_cap_pause parameters.0
lp_cap_1000hdx	
lp_cap_100fdx	
lp_cap_100hdx	
lp_cap_10fdx	
lp_cap_10gfdx	
lp_cap_10hdx	
lp_cap_asm_pause	
lp_cap_autoneg	
lp_cap_pause	
rx_coalesce_mode	Large Receive Offload. 0 = disabled (default), 1 = enabled, 2 = enabled - respecting TCP PSH boundaries.
rx_scale_count	Number of RSS channels to use per port. Default is 128, Minimum is 1.

Table 84: Name sfxge_rss

Parameter	Description
crtime	Timestamp when samples were taken.
evq0000_count	Number of RSS table entries for this event queue.
scale	Actual number of MSI-X interrupts.

Table 85: Name sfxge_rxq0000

Parameter	Description
crtime	Timestamp when samples were taken.
dma_alloc_fail	Memory allocation failure.
dma_alloc_nomem	Memory allocation failure.
dma_bind_fail	Memory allocation failure.
dma_bind_nomem	Memory allocation failure.
kcache_alloc_nomem	Memory allocation failure.

Table 85: Name sfxge_rxq0000

Parameter	Description
rx_pkt_mem_limit	Per interface memory limit for RX packet buffers.
rxq_empty_discard	Number of times the RX descriptor ring was empty causing a received packet to be discarded.

Table 86: Name sfxge_txq0000

Parameter	Description
crtime	Timestamp when samples were taken.
post	Number of packets posted to the transmit queue.
dpl_get_full_count	Number of times the Deferred Packet List limit was reached.
dpl_get_pkt_limit	Deferred Packet max packet limit
dpl_put_full_count	Number of times the Deferred Packet List limit was reached
dpl_put_pkt_limit	Deferred Packet max packet limit.
unaligned_split	Always 0.

Table 87: Name sfxge_vpd

Parameter	Description
crtime	Timestamp when samples were taken.
EC	Engineering change data.
ID	Solarflare adapter type.
PN	Solarflare adapter part number.
SN	Solarflare adapter serial number.
VD	Adapter firmware version.

7 Solarflare Adapters on FreeBSD

This chapter covers the following topics on the FreeBSD platform:

7.1 System Requirements

Refer to Software Driver Support on page 13 for details of supported FreeBSD distributions.

NOTE: FreeBSD includes a previous version of the Solarflare adapter driver that does not support all features of this version. To update the supplied driver, see Installing Solarflare Drivers on page 336.

7.2 FreeBSD Platform Feature Set

Table 88 lists the features supported by Solarflare adapters on FreeBSD.

Table 88: FreeBSD Feature Set

Jumbo frames	Support for MTUs (Maximum Transmission Units) to 9000 bytes. • See Configuring Jumbo Frames on page 341
Task offloads	Support for TCP Segmentation Offload (TSO), Large Receive Offload (LRO), and TCP/UDP/IP checksum offload for improved adapter performance and reduced CPU processing requirements. • See Configuring Task Offloading on page 341
Receive Side Scaling (RSS)	Support for RSS multi-core load distribution technology. • See Receive Side Scaling (RSS) on page 361
Virtual LANs (VLANs)	Support for multiple VLANs per adapter. • See Setting Up VLANs on page 341
PXE and booting	Support for diskless booting to a target operating system via PXE or iSCSI boot. • See Configuring the Boot ROM with sfboot on page 343 • See Solarflare Boot ROM Agent on page 435
Firmware updates	Support for Boot ROM, PHY transceiver and adapter firmware upgrades. • See Upgrading Adapter Firmware with sfupdate on page 354

7.3 Installing Solarflare Drivers

The FreeBSD drivers for Solarflare are available in a source package.

- A package is available for FreeBSD 10.0 and 10.1:
  - this package might perform correctly with other FreeBSD kernels, but has not been tested with them
  - for further details see the *Release Notes*.

NOTE: The Solarflare adapter should be physically installed in the host computer before you attempt to install drivers. You must have root permissions to install the adapter drivers.

This source can be used:

- To compile and install a driver on a development machine.

  The development machine must have the following installed:

  - development tools
  - the *ports* system, and its Makefiles
  - the kernel source.

- To create a binary driver package, for installing on other target machines.

  A target machine:

  - must have the same kernel and architecture as the development machine that built the package
  - does not require any of the development tools or source.

  To install the driver, use pkg add or pkg_add.

The following instructions assume that the source package has been downloaded to the `/tmp` directory.

1 Ensure you are the root user. If not:

```
su -
```

2 To avoid using the previous driver that is distributed with the OS, rename it:

```
mv /boot/kernel/sfxge.ko /boot/kernel/sfxge.ko_default
```

- If desired, it can instead be removed:

  ```
  rm /boot/kernel/sfxge.ko
  ```

3 Unpack the downloaded source:

```
cd /tmp
tar xvf sfxge-freebsd-<version_no>.txz
```

For example:

```
cd /tmp
tar xvf sfxge-freebsd-4.5.3.1002.txz
```

4 Change directory into the source:

```
cd sfxge-freebsd-<version_no>
```

For example:

```
cd sfxge-freebsd-4.5.3.1002
```

5 Build the package:

```
make package
```

- If you do not have the ports system installed you will see this error:

  ```
  The ports system must be installed first.
  ```

- You can install the ports system by running:

  ```
  portsnap fetch extract
  ```

6 Install the package:

```
make install
```

- the driver (for use on this machine) is installed in `/boot/modules/sfxge.ko`
- the binary driver package (for use on other machines) is installed in the build directory in `work/package/sfxge-kmod-<version_no>.txz`, and in `/usr/ports/packages/All/sfxge-kmod-<version_no>.txz`

7 Load the driver:

```
kldload /boot/modules/sfxge.ko
```

For information on configuring this network interface see Configuring the Solarflare Adapter on page 340.

7.4 Unattended Installation

Unattended installations of FreeBSD can be performed by PXE booting over the network, and using the `bsdinstall` command. Set this up as follows:

- Ensure that DHCP is available, with PXE boot options.
- Ensure that a TFTP server is available.
- Ensure that a FreeBSD server is available.

  This is required only to generate the FreeBSD PXE boot image:

  - Download the `mfsbsd` utility (available from http://mfsbsd.vx.sk/).
    Install it on the FreeBSD server.[1]
  - Download the ISO image for the required FreeBSD release.
    Mount this boot image on the FreeBSD server.
  - Configure and customize the boot image.
    Change any PXE boot settings as necessary.
    Modify the `/etc/installerconfig` file in the boot image to add any post-install tasks.
  - Use the `mfsbsd` utility to build the PXE boot image, using the modified boot image as source.
- Copy the PXE boot image to the TFTP server

  Add a FreeBSD option to its pxelinux boot menu.
- PXE boot the target server and select the FreeBSD image.
- The FreeBSD server that was used to generate the FreeBSD PXE boot image can now be re-used.

1. The `mfsbsd` utility runs only under FreeBSD.

FreeBSD install and booting is documented as follows:

- For information on booting a FreeBSD system over the network, see:
  https://www.freebsd.org/cgi/man.cgi?query=diskless&sektion=8
- For general information on using `bsdinstall`, see:
  https://www.freebsd.org/doc/handbook/bsdinstall.html
- For a reference description of the `bsdinstall` command, see:
  https://www.freebsd.org/cgi/man.cgi?query=bsdinstall&sektion=8
  especially the *SCRIPTING* section.

Table 89 shows an example time line for an unattended installation.

Table 89: Installation Stages

In Control	Stages of Boot	Setup needed
BIOS	PXE code on the adapter runs.	Adapter must be in PXE boot mode. See PXE Support on page 436.
SF Boot ROM (PXE)	DHCP request from PXE (SF Boot ROM).	DHCP server filename and next-server options.
SF Boot ROM (PXE)	TFTP request for filename to next-server, e.g. pxelinux.0	TFTP server.
pxelinux	TFTP retrieval of pxelinux configuration.	pxelinux configuration on TFTP server.
pxelinux	TFTP menu retrieval of FreeBSD kernel image.	pxelinux configuration Kernel, kernel command
FreeBSD kernel/ installer	Installer retrieves configuration.	FreeBSD image
Installation occurs	Machine reboots	`/etc/installerconfig` file
Target FreeBSD kernel	kernel reconfigures network adapters.	DHCP server.

7.5 Configuring the Solarflare Adapter

The drivers will be loaded as part of the as part of the installation. However the adapter will not be configured (adding IP address and netmask).

Each Solarflare network adapter interface will be named `sfxge<n>` where `<n>` is a unique identifier. There will be one interface per physical port on the Solarflare adapter.

To configure the interface and bring it up to allow data to pass, enter the following:

```
ifconfig sfxge<n> inet <IPv4 address> netmask <netmask> up
```

This configures the interface and initializes it with the up command.

NOTE: This method of configuring is temporary. If you reboot your computer the settings will be lost. To make these settings permanent, create entries in the configuration file as described below.

Using IPv6

To configure using IPv6, create an IPv6 interface sfxge<n> interface with a link local IPv6 address by entering:

```
ifconfig sfxge<n> inet6 <IPv6 address> prefixlen <IPv6 prefix length>
```

This uses automatic link-local address configuration, which is enabled by default in FreeBSD. It will give an IPv6 interface name of `sfxge<n>:1`

Using a Configuration File with IPv4

Configuration is set in the `/etc/rc.conf` file. There are three options with IPv4:

- Using a static IPv4 address. To use this option, add:

  ```
  ifconfig_sfxge<n>="inet <IPv4 address> netmask <netmask>"
  ```

- Using a hostname. To use this option, add:

  ```
  ifconfig_sfxge<n>="inet <hostname>"
  ```

  and modify `/etc/hosts` and `/etc/netmasks`

- Using DHCP. To use this option, add:

  ```
  ifconfig_sfxge<n>="DHCP"
  ```

Using Configuration files with IPv6

Configuration is set in the `/etc/rc.conf` file:

- For automatic configuration by StateLess Address AutoConfiguration (SLAAC), add:

  ```
  ifconfig_sfxge<n>_ipv6="inet6 accept_rtadv"
  ```


Configuring Task Offloading

Solarflare adapters support transmit (Tx) and receive (Rx) checksum offload, as well as TCP segmentation offload. To ensure maximum performance from the adapter, all task offloads should be enabled, which is the default setting on the adapter. For more information, see Performance Tuning on FreeBSD on page 356.

Configuring Jumbo Frames

Solarflare adapters support a frame size (MTU) from 1500 bytes to 9000 bytes.

The default maximum driver MTU size is 1500 bytes. For example, to set a new frame size (MTU) of 9000 bytes, enter the following command:

```
ifconfig sfxge<n> inet mtu 9000
```

To view the current MTU, enter:

```
# ifconfig sfxge<n>
sfxge0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        ...
```

If you want to have an MTU configured when the interface is brought up, add an `mtu` parameter to the single line of interface configuration data in the `/etc/rc.conf` file. For example:

```
ifconfig_sfxge<n>="inet <IPv4 address> netmask <netmask> mtu <MTU size>"
```

7.6 Setting Up VLANs

VLANs offer a method of dividing one physical network into multiple broadcast domains. In enterprise networks, these broadcast domains usually match with IP subnet boundaries, so that each subnet has its own VLAN. The advantages of VLANs include:

- Performance
- Ease of management
- Security
- Trunks
- You don't have to configure any hardware device, when physically moving your server to another location.

To have a single interface exist on multiple VLANs (if the port on the connected switch is set to “trunked” mode) see the following documentation:

http://people.freebsd.org/~arved/vlan/vlan_en.html

7.7 FreeBSD Utilities Package

The Solarflare FreeBSD Utilities package is supplied as a source package or a 64 bit binary package, and is available from https://support.solarflare.com/. It contains the following utilities:

Table 90: Utilities Package

Utility File	Description
sfupdate	A command line utility that contains an adapter firmware version which can update Solarflare adapter firmware.
sfboot	A command line utility for configuring Solarflare adapter Boot ROM options, including PXE and iSCSI booting.
sfreport	A command line utility that generates a diagnostic log file providing diagnostic data about the server and Solarflare adapters.

By default, `sfboot` and `sfupdate` are installed to `/usr/local/sbin`, and `sfreport` is installed to `/usr/local/bin`.

Building and installing the source package

To build and install the source package:

1 Unpack the source package:

```
tar -xf <source package name>
```

2 Go to its directory:

```
cd <source package dir>
```

3 Build and install a binary package from the source:

```
make install
```

Alternatively, to build and install in separate steps:

a) Build a binary package from the source:

```
make package
```

b) Install the resulting binary package:

```
pkg install ./work/pkg/sfutils-<version>.txz
```

Installing the 64 bit binary package

1 Install the binary package:

```
pkg install <path to package file>
```


7.8 Configuring the Boot ROM with sfboot

Sfboot is a command line utility for configuring Solarflare adapter Boot ROM options, including PXE and iSCSI booting. Using sfboot is an alternative to using **Ctrl + B** to access the Boot Rom agent during server startup.

See Configuring the Solarflare Boot ROM Agent on page 436 for more information on the Boot Rom agent.

Sfboot: Command Usage

The general usage for sfboot is as follows (as root):

```
sfboot [--adapter=sfxge<n>] [options] [parameters]
```

Note that without `--adapter`, the `sfboot` command applies to all adapters that are present in the target host.

The format for the parameters are:

```
<parameter>=<value>
```

Sfboot: Command Line Options

Table 91 lists the options for `sfboot`, Table 92 lists the available global parameters, and Table 93 lists the available per-adapter parameters.

Table 91: Sfboot Options

Option	Description
`-h, --help`	Displays command line syntax and provides a description of each sfboot option.
`-V, --version`	Shows detailed version information and exits.
`-v, --verbose`	Shows extended output information for the command entered.
`-s, --silent`	Suppresses all output, including warnings and errors; no user interaction. You should query the completion code to determine the outcome of commands when operating silently.
`--log <filename>`	Logs output to the specified file in the current folder or an existing folder. Specify `--silent` to suppress simultaneous output to screen, if required.
`--computer <computer_name>`	Performs the operation on a specified remote computer. Administrator rights on the remote computer is required.

Table 91: Sfboot Options

Option	Description
`--list`	Lists all available Solarflare adapters. This option shows the adapter's ID number, ifname and MAC address. Note: this option may not be used in conjunction with any other option. If this option is used with configuration parameters, those parameters will be silently ignored.
`-d, --adapter =<sfxge<n>>`	Performs the action on the identified Solarflare network adapter. The adapter identifier `sfxge` can be the adapter ID number, ifname or MAC address, as output by the `--list` option. If `--adapter` is not included, the action will apply to all installed Solarflare adapters.
`--clear`	Resets all options to their default values. If an adapter is specified, options for the given adapter are reset, but global options (shown in Table 92) are not reset. Note that `--clear` can also be used with parameters, allowing you to reset to default values, and then apply the parameters specified.

The following global parameters in Table 92 are used to control the configurable parameters for the Boot ROM driver when running prior to the operating system booting.

Table 92: Sfboot Global Parameters

Parameter	Description
`boot-image=` `all|optionrom|uefi|disabled`	Specifies which boot firmware images are served-up to the BIOS during start-up. This parameter can not be used if the `--adapter` option has been specified.

`port-mode=`
`default|1x10G|2x10G|4x10G|`
`2x40G`
	Configure the port mode to use. This is for SFN7000 series adapters only. The values specify the connectors available after using any splitter cables. The usable values are adapter-dependent:
	• SFN7*xx*2F: 1x10G, 2x10G (default)
	• SFN7*xx*4F: 2x10G, 4x10G (default)
	• SFN7*xx*2Q: 2x10G, 4x10G, 2x40G (default)
	
	Changes to this setting with `sfboot` require a cold reboot to become effective. MAC address assignments may change after altering this setting.

Table 92: Sfboot Global Parameters

Parameter	Description
`firmware-variant= full-feature|ultra-low-latency| capture-packed-stream|auto`	Configure the firmware variant to use. This is for SFN7000 series adapters only: • the SFN7002F adapter is factory set to full-feature • all other adapters are factory set to auto. Default value = auto - means the driver will select a variant that meets its needs: • the VMware driver always uses full-feature • otherwise, ultra-low-latency is used. The ultra-low-latency variant produces best latency without support for TX VLAN insertion or RX VLAN stripping (not currently used features). It is recommended that Onload customers use the ultra-low-latency variant.
`insecure-filters= enabled|disabled`	If enabled bypass filter security on non-privileged functions. This is for SFN7000 series adapters only. This reduces security in virtualized environments. The default is disabled. When enabled a function (PF or VF) can insert filters not qualified by their own permanent MAC address. This is a requirement when using Onload or when using bonded interfaces.
`mac-spoofing=enabled|disabled`	If enabled, non-privileged functions can create unicast filters for MAC addresses that are not associated with them. This is for SFN7000 series adapters only. The default is disabled. Changes to this setting with `sfboot` require a cold reboot to become effective.
`rx-dc-size=8|16|32|64`	Specifies the size of the descriptor cache for each receive queue. This is for SFN7000 series adapters only. The default is: • 16 if the `port-mode` supports the maximum number of connectors for the adaptor • 32 if the `port-mode` supports a reduced number of connectors.

Table 92: Sfboot Global Parameters

Parameter	Description
`tx-dc-size=8|16|32|64`	Specifies the size of the descriptor cache for each transmit queue. This is for SFN7000 series adapters only. The default is: • `32` if the `port-mode` supports the maximum number of connectors for the adaptor • `64` if the `port-mode` supports a reduced number of connectors.
`vi-count=<vi count>`	Sets the total number of virtual interfaces that will be available on the NIC.

The following per-adapter parameters in Table 93 are used to control the configurable parameters for the Boot ROM driver when running prior to the operating system booting.

Table 93: Sfboot Per-adapter Parameters

Parameter	Description
`link-speed=auto|10g|1g|100m`	Specifies the network link speed of the adapter used by the Boot ROM. The default is `auto`. On the 10GBASE-T adapters, `auto` instructs the adapter to negotiate the highest speed supported in common with its link partner. On SFP+ adapters, `auto` instructs the adapter to use the highest link speed supported by the inserted SFP+ module. On 10GBASE-T and SFP+ adapters, any other value specified will fix the link at that speed, regardless of the capabilities of the link partner, which may result in an inability to establish the link. `auto` Auto-negotiate link speed (default) `10G` 10G bit/sec `1G` 1G bit/sec `100M` 100M bit/sec
`linkup-delay=` `<delay time in seconds>`	Specifies the delay (in seconds) the adapter defers its first connection attempt after booting, allowing time for the network to come up following a power failure or other restart. This can be used to wait for spanning tree protocol on a connected switch to unblock the switch port after the physical network link is established. The default is 5 seconds.

Table 93: Sfboot Per-adapter Parameters

Parameter	Description
`banner-delay= <delay time in seconds>`	Specifies the wait period for `Ctrl-B` to be pressed to enter adapter configuration tool. `<delay time in seconds>` = 0-256
`bootskip-delay= <delay time in seconds>`	Specifies the time allowed for `Esc` to be pressed to skip adapter booting. `<delay time in seconds>` = 0-256
`boot-type=pxe|iscsi|disabled`	Sets the adapter boot type – effective on next boot. pxe – PXE (Preboot eXecution Environment) booting `iscsi` – iSCSI (Internet Small Computer System Interface) booting `disabled` – Disable adapter booting

`initiator-dhcp=enabled|disabled`	Enables or disables DHCP address discovery for the adapter by the Boot ROM except for the Initiator IQN (see `initiator-iqn-dhcp`). This option is only valid if iSCSI booting is enabled (`boot-type=iscsi`).
	If initiator-DHCP is set to disabled, the following options will need to be specified:
	`initiator-ip=<IPv4 address>`
	`netmask=<IPv4 subnet mask>`
	The following options may also be needed:
	`gateway=<ip_address>`
	`primary-dns=<ip_address>`

`initiator-ip=<IPv4 address>`	Specifies the IPv4 address (in standard "." notation form) to be used by the adapter when `initiator-dhcp` is disabled. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot boot-type=iscsi initiator-dhcp=disabled initiator-ip=<192.168.1.3>`
`netmask=<IPv4 subnet mask>`	Specifies the IPv4 subnet mask (in standard "." notation form) to be used by the adapter when `initiator-dhcp` is disabled. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot boot-type=iscsi initiator-dhcp=disabled netmask=255.255.255.0`

Table 93: Sfboot Per-adapter Parameters

Parameter	Description
`gateway=<IPv4 address>`	Specifies the IPv4 subnet mask (in standard "." notation form) to be used by the adapter when `initiator-dhcp` is disabled. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot boot-type=iscsi initiator-dhcp=disabled gateway=192.168.0.10`
`primary-dns=<IPv4 address>`	Specifies the IPv4 address (in standard "." notation form) of the Primary DNS to be used by the adapter when `initiator-dhcp` is disabled. This option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot boot-type=iscsi initiator-dhcp=disabled primary-dns=192.168.0.3`
`initiator-iqn-dhcp= enabled|disabled`	Enables or disables use of DHCP for the initiator IQN only.
`initiator-iqn=<IQN>`	Specifies the IQN (iSCSI Qualified Name) to be used by the adapter when `initiator-iqn-dhcp` is disabled. The IQN is a symbolic name in the "." notation form; for example: iqn.2009.01.com.solarflare, and is a maximum of 223 characters long. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot initiator-iqn-dhcp=disabled` initiator-iqn=iqn.2009.01.com.solarflare adapter=2
`lun-retry-count=<retry count>`	Specifies the number of times the adapter attempts to access and login to the Logical Unit Number (LUN) on the iSCSI Target before failing. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot lun-retry-count=3`

Table 93: Sfboot Per-adapter Parameters

Parameter	Description

`target-dhcp=enabled|disabled`	Enables or disables the use of DHCP to discover iSCSI target parameters on the adapter.
	If `target-dhcp` is disabled, you must specify the following options:
	`target-server=<DNS name or IPv4 address>`
	`target-iqn=<IQN>`
	`target-port=<port number>`
	`target-lun=<LUN>`
	Example - Enable the use of DHCP to configure iSCSI Target settings:
	`sfboot boot-type=iscsi target-dhcp=enabled`

`target-server=` `<DNS name or IPv4 address>`	Specifies the iSCSI target's DNS name or IPv4 address to be used by the adapter when `target-dhcp` is disabled. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot boot-type=iscsi target-dhcp=disabled target-server=192.168.2.2`
`target-port=<port number>`	Specifies the Port number to be used by the iSCSI target when `target-dhcp` is disabled. The default Port number is Port 3260. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Example: `sfboot boot-type=iscsi target-dhcp=disabled target-port=3262` This option should only be used if your target is using a non-standard TCP Port.
`target-lun=<LUN>`	Specifies the Logical Unit Number (LUN) to be used by the iSCSI target when `target-dhcp` is disabled. The default LUN is 0. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`).

Table 93: Sfboot Per-adapter Parameters

Parameter	Description
`target-iqn=<IQN>`	Specifies the IQN of the iSCSI target when `target-dhcp` is disabled. Maximum of 223 characters. Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`). Note that if there are spaces contained in `<IQN>`, then the IQN must be wrapped in double quotes (“”). Example: `sfboot target-dhcp=disabled target-iqn=iqn.2009.01.com.solarflare adapter=2`
`vendor-id=<vendor identifier>`	Specifies the device vendor ID to be advertised to the DHCP server. This must match the vendor id configured at the DHCP server when using DHCP option 43 to obtain the iSCSI target.

`chap=enabled|disabled`	Enables or disables the use of Challenge Handshake Protocol (CHAP) to authenticate the iSCSI connection.
	
	Note that this option is only valid if iSCSI booting is enabled (`boot-type=iscsi`).
	
	To be valid, this option also requires the following sub-options to be specified:
	
	`username=<initiator username>`
	`secret=<initiator password>`
	
	Example:
	
	`sfboot boot-type=iscsi chap=enabled username=initiatorusername secret=initiatorsecret`

`username=<username>`	Specifies the CHAP initiator username (maximum 64 characters). Note that this option is required if either CHAP or Mutual CHAP is enabled (`chap=enabled, mutual-chap=enabled`). Note that if there are spaces contained in `<username>`, then it must be wrapped in double quotes (“”). Example: `sfboot boot-type=iscsi chap=enabled username=username`

Table 93: Sfboot Per-adapter Parameters

Parameter	Description
`secret=<secret>`	Specifies the CHAP initiator secret (minimum 12 characters, maximum 20 characters). Note that this option is valid if either CHAP or Mutual CHAP is enabled (chap=enabled, `mutual-chap=enabled`). Note that if there are spaces contained in `<secret>`, then it must be wrapped in double quotes (“”). Example: `sfboot boot-type=iscsi chap=enabled username=username secret=veryverysecret`

`mutual-chap=enabled|disabled`	Enables/disables Mutual CHAP authentication when iSCSI booting is enabled.
	
	This option also requires the following sub-options to be specified:
	
	`target-username=<username>`
	`target-secret=<password>`
	`username=<username>`
	`secret=<password>`
	
	Example:
	
	`sfboot boot-type=iscsi mutual-chap=enabled username=username secret=veryverysecret target-username=targetusername target-secret=anothersecret`

`target-username=<username>`	Specifies the username that has been configured on the iSCSI target (maximum 64 characters). Note that this option is necessary if Mutual CHAP is enabled on the adapter (`mutual-chap=enabled`). Note that if there are spaces contained in `<username>`, then it must be wrapped in double quotes (“”).
`target-secret=<secret>`	Specifies the secret that has been configured on the iSCSi target (minimum 12 characters; maximum 20 characters). Note: This option is necessary if Mutual CHAP is enabled on the adapter (`mutual-chap=enabled`). Note that if there are spaces contained in `<secret>`, then it must be wrapped in double quotes (“”).
`mpio-priority=<MPIO priority>`	Specifies the Multipath I/O (MPIO) priority for the adapter. This option is only valid for iSCSI booting over multi-port adapters, where it can be used to establish adapter port priority. The range is 1- 255, with 1 being the highest priority.

Table 93: Sfboot Per-adapter Parameters

Parameter	Description
`mpio-attempts=<attempt count>`	Specifies the number of times MPIO will try and use each port in turn to login to the iSCSI target before failing.
`pf-count=<pf count>`	This is the number of available PCIe PFs per physical network port. This setting is applied to all ports on the adapter. Changes to this setting with `sfboot` require a cold reboot to become effective. MAC address assignments may change after altering this setting.
`msix-limit=` `8|16|32|64|128|256|512|1024`	Specifies the maximum number of MSI-X interrupts that each PF will use. The default is 32. Note: Using the incorrect setting can impact the performance of the adapter. Contact Solarflare technical support before changing this setting.
`sriov=enabled|disabled`	Enable SR-IOV support for operating systems that support this. Not required on SFN7000 series adapters.
`vf-count=<vf count>`	The number of virtual functions (VF) advertised to the operating system for each Physical Function on this physical network port. SFN7000 series adapters have a total limit of 2048 interrupts. Earlier adapters support a total limit of 127 virtual functions per port and a total of 1024 interrupts. Depending on the values of msix-limit and vf-msix-limit, some of these virtual functions may not be configured. Enabling all 127 VFs per port with more than one MSI-X interrupt per VF may not be supported by the host BIOS - in which case you may get 127 VFs on one port and none on others. Contact your BIOS vendor or reduce the VF count. The sriov parameter is implied if vf-count is greater than zero. Changes to this setting with `sfboot` require a cold reboot to become effective.
`vf-msix-limit=` `1|2|4|8|16|32|64|128|256`	The maximum number of interrupts a virtual function may use.
`pf-vlans=<tag>[,<tag>[,...]]|none`	Comma separated list of VLAN tags for each PF in the range 0-4094 - see sfboot --help for details.

Table 93: Sfboot Per-adapter Parameters

Parameter	Description
`switch-mode=default|sriov|partitioning|partitioning-with-sriov|pfiov`	Specifies the mode of operation that the port will be used in: `default` - single PF created, zero VFs created. `sriov` - SR-IOV enabled, single PF created, VFs configured with `vf-count`. `partitioning` - PFs configured with `pf-count`, VFs configured with `vf-count`. See NIC Partitioning on page 58 for details. `partitioning-with-sriov` - SR-IOV enabled, PFs configured with `pf-count`, VFs configured with `vf-count`. See NIC Partitioning on page 58 for details. `pfiov` - PFIOV enabled, PFs configured with `pf-count`, VFs not supported. Changes to this setting with `sfboot` require a cold reboot to become effective.

Sfboot: Examples

- Show the current boot configuration for all adapters:

  `sfboot`

```
Solarflare boot configuration utility [v3.0.5]
Copyright Solarflare Communications 2006-2010, Level 5 Networks 2002-2005

sfxge0:
  Boot image                        Disabled
  MSI-X interrupt limit             32

sfxge1:
  Boot image                        Disabled
  MSI-X interrupt limit             32
```

- List all Solarflare adapters installed on the localhost:

  `sfboot --list`

```
Solarflare boot configuration utility [v3.0.5]
Copyright Solarflare Communications 2006-2010, Level 5 Networks 2002-2005
sfxge0 - 00-0F-53-01-38-40
sfxge1 - 00-0F-53-01-38-41
```


7.9 Upgrading Adapter Firmware with sfupdate

To Update Adapter Firmware

Reinstall the `sfutils` package, as described in FreeBSD Utilities Package on page 342.

Sfupdate: Command Usage

The general usage for sfupdate is as follows (as root):

`sfupdate [--adapter=sfxge<n>] [options]`

where:

- `sfgxe<n>` is the interface name of the Solarflare adapter you want to upgrade.
- `option` is one of the command options listed in Table 94.

The format for the options are:

`--<option>=<parameter>`

Running the command `sfupdate` with no additional parameters will show the current firmware version for all Solarflare adapters and whether the firmware within sfupdate is more up to date. To update the firmware for all Solarflare adapters run the command `sfupdate --write`

Solarflare recommend that you use sfupdate in the following way:

1. Run `sfupdate` to check that the firmware on all your adapters are up to date.
2. Run `sfupdate --write` to update the firmware on all adapters.

Sfupdate: Command Line Options

Table 94 lists the options for `sfupdate`.

Table 94: Sfupdate Options

Option	Description
`-h, --help`	Shows help for the available options and command line syntax.
`-v, --verbose`	Enable verbose output mode.
`-s, --silent`	Suppress all output except for errors. Useful for scripts.
`-V, --version`	Display version information and exit.
`-i, --adapter=sfxge<n>`	Specifies the target adapter when more than one adapter is installed in the machine. `sfxge<n>` = Adapter ifname or MAC address (as obtained with `--list`).

Table 94: Sfupdate Options

Option	Description
`--list`	Shows the adapter ID, adapter name and MAC address of each adapter installed in the machine.
`--write`	Re-writes the firmware from the images embedded in the `sfupdate` tool. To re-write using an external image, specify `--image=<filename>` in the command. `--write` fails if the embedded image is the same or a previous version. To force a write in this case, specify the option `--force`.
`--force`	Force update of all firmware, even if the installed firmware version is the same or more recent than the images embedded in the utility.
`--image=(filename)`	Update the firmware using the image contained in the specified file, rather than the image embedded in the utility. Use with the `--write` and, if needed, `--force` options.
`-y, --yes`	Prompts for user confirmation before re-writing the firmware.

Sfupdate: Examples

- Display firmware versions for all adapters:

  `sfupdate`

```
sfupdate: Solarflare Firmware Update Utility [v3.0.5.2164]
Copyright Solarflare Communications 2006-2010, Level 5 Networks 2002-2005
Network adapter driver version: v3.0.5.2163

sfxge0 - MAC: 00:0F:53:01:38:90
    Firmware   version: v3.0.5
    Boot ROM   version: v3.0.5.2163
    PHY        version: v2.0.2.5
    Controller version: v3.0.5.2161

The Boot ROM firmware is up to date
The PHY firmware is up to date
The image contains a more recent version of the Controller [v3.0.5.2163]
vs [v3.0.5.2161]
Use the -w|--write option to perform an update

sfxge1 - MAC: 00:0F:53:01:38:91
    Firmware   version: v3.0.5
    Boot ROM   version: v3.0.5.2163
    PHY        version: v2.0.2.5
    Controller version: v3.0.5.2161

The Boot ROM firmware is up to date
The PHY firmware is up to date
The image contains a more recent version of the Controller [v3.0.5.2163]
vs [v3.0.5.2161]
Use the -w|--write option to perform an update
```


7.10 Performance Tuning on FreeBSD

Introduction

The Solarflare family of network adapters are designed for high-performance network applications. The adapter driver is pre-configured with default performance settings that have been designed to give good performance across a broad class of applications. Occasionally, application performance can be improved by tuning these settings to best suit the application.

There are three metrics that should be considered when tuning an adapter:

- Throughput
- Latency
- CPU utilization

Different applications may be more or less affected by improvements in these three metrics. For example, transactional (request-response) network applications can be very sensitive to latency whereas bulk data transfer applications are likely to be more dependent on throughput.

The purpose of this section is to highlight adapter driver settings that affect the performance metrics described. This section covers the tuning of all Solarflare adapters.

Latency will be affected by the type of physical medium used: 10GBase-T, twinaxial (direct-attach), fiber or KX4. This is because the physical media interface chip (PHY) used on the adapter can introduce additional latency. Likewise, latency can also be affected by the type of SFP/SFP+/QSFP module fitted.

In addition, you may need to consider other issues influencing performance, such as application settings, server motherboard chipset, CPU speed, cache size, RAM size, additional software installed on the system, such as a firewall, and the specification and configuration of the LAN. Consideration of such issues is not within the scope of this guide.

Tuning settings

Port mode

The selected port mode for SFN7000 series adapters should correspond to the speed and number of connectors in use, after using any splitter cables. If a restricted set of connectors is configured, the driver can then transfer resources from the unused connectors to those configured, potentially improving performance.

Adapter MTU (Maximum Transmission Unit)

The default MTU of 1500 bytes ensures that the adapter is compatible with legacy 10/100Mbps Ethernet endpoints. However if a larger MTU is used, adapter throughput and CPU utilization can be improved. CPU utilization is improved, because it takes fewer packets to send and receive the same amount of data. Solarflare adapters support an MTU of up to 9216 bytes (this does not include the Ethernet preamble or frame-CRC).

Since the MTU should ideally be matched across all endpoints in the same LAN (VLAN), and since the LAN switch infrastructure must be able to forward such packets, the decision to deploy a larger than default MTU requires careful consideration. It is recommended that experimentation with MTU be done in a controlled test environment.

The MTU is changed dynamically using `ifconfig`, where `sfxge<n>` is the interface name and `<size>` is the MTU size in bytes:

```
# ifconfig sfxge<n> mtu <size>
```

Verification of the MTU setting may be performed by running `ifconfig` with no options and checking the MTU value associated with the interface. The change in MTU size can be made to persist across reboots by editing the `/etc/rc.conf` file and adding an `mtu` parameter to the single line of interface configuration data. For example:

```
ifconfig_sfxge<n>="inet <IPv4 address> netmask <netmask> mtu <size>"
```

Interrupt Moderation (Interrupt Coalescing)

Interrupt moderation reduces the number of interrupts generated by the adapter by coalescing multiple received packet events and/or transmit completion events together into a single interrupt.

The *interrupt moderation interval* sets the minimum time (in microseconds) between two consecutive interrupts. Coalescing occurs only during this interval:

- When the driver generates an interrupt, it starts timing the moderation interval.
- Any events that occur before the moderation interval expires are coalesced together into a single interrupt, that is raised only when the interval expires.

  A new moderation interval then starts, during which no interrupt is raised.
- An event that occurs after the moderation interval has expired gets its own dedicated interrupt, that is raised immediately.

  A new moderation interval then starts, during which no interrupt is raised.

Interrupt moderation settings are **critical for tuning adapter latency**:

- Increasing the interrupt moderation interval will:
  - generate less interrupts
  - reduce CPU utilization (because there are less interrupts to process)
  - increase latency
  - improve peak throughput.
- Decreasing the interrupt moderation interval will:
  - generate more interrupts
  - increase CPU utilization (because there are more interrupts to process)
  - decrease latency
  - reduce peak throughput.
- Turning off interrupt moderation will:
  - generate the most interrupts
  - give the highest CPU utilization
  - give the lowest latency
  - give the biggest reduction in peak throughput.

For many transaction request-response type network applications, the benefit of reduced latency to overall application performance can be considerable. Such benefits typically outweigh the cost of increased CPU utilization. It is recommended that:

- Interrupt moderation is disabled for applications that require best latency and jitter performance, such as market data handling.
- Interrupt moderation is enabled for high throughput single (or few) connection TCP streaming applications, such as iSCSI.

Interrupt moderation is changed dynamically using `sysctl`.

To set the interrupt moderation, where `sfxge<n>` is the interface name, and the `<interval>` is in microseconds (µs):

```
sysctl dev.sfxge.<n>.int_mod=<interval>
```

To turn off interrupt moderation, set an interval of zero (0):

```
sysctl dev.sfxge.<n>.int_mod=0
```

The change in interrupt moderation can be made to persist across reboots by editing the file `/etc/sysctl.conf`and adding `dev.sfxge.<n>.int_mod=<interval>` on a new line.

NOTE: The performance benefits of TCP Large Receive Offload are limited if interrupt moderation is disabled. See TCP Large Receive Offload (LRO) on page 360.

TCP/IP Checksum Offload

Checksum offload moves calculation and verification of IP Header, TCP and UDP packet checksums to the adapter. The driver has all checksum offload features enabled by default. Therefore, there is no opportunity to improve performance from the default.

Checksum offload is changed dynamically using `ifconfig`, with the following parameters:

- `rxcsum`, `txcsum`, `rxcsum6`, `txcsum6`

  Enable Rx and Tx checksum offload for IPv4 and IPv6

- `-rxcsum`, `-txcsum`, `-rxcsum6`, `-txcsum6`

  Disable Rx and Tx checksum offload for IPv4 and IPv6

To enable checksum offload, where `sfxge<n>` is the interface name:

```
# ifconfig sfxge<n> rxcsum txcsum rxcsum6 txcsum6
```

To disable checksum offload:

```
# ifconfig sfxge<n> -rxcsum -txcsum -rxcsum6 -txcsum6
```

Verification of the checksum offload setting may be performed by running `ifconfig` with no options and checking the checksum offload value associated with the interface. The change in checksum offload can be made to persist across reboots by editing the `/etc/rc.conf` file and adding the appropriate parameters to the single line of interface configuration data. For example:

```
ifconfig_sfxge<n>="inet <IPv4 address> netmask <netmask> rxcsum txcsum rxcsum6 txcsum6"
```

NOTE: Solarflare recommend you do not disable checksum offload.

TCP Segmentation Offload (TSO)

TCP Segmentation Offload (TSO) offloads the splitting of outgoing TCP data into packets to the adapter. TSO benefits applications using TCP. Applications using protocols other than TCP will not be affected by TSO.

The FreeBSD TCP/IP stack provides a large TCP segment to the driver, which splits the data into MSS size, each with adjusted sequence space and a hardware calculated checksum.

Enabling TSO will reduce CPU utilization on the transmit side of a TCP connection and improve peak throughput, if the CPU is fully utilized. Since TSO has no effect on latency, it can be enabled at all times. The driver has TSO enabled by default. Therefore, there is no opportunity to improve performance from the default.

TSO is changed dynamically using `ifconfig`.

To enable TSO, where `sfxge<n>` is the interface name:

```
# ifconfig sfxge<n> tso
```

To disable TSO:

```
# ifconfig sfxge<n> -tso
```


Verification of the TSO setting may be performed by running `ifconfig` with no options and checking the TSO value associated with the interface. The change in TSO can be made to persist across reboots by editing the `/etc/rc.conf` file and adding the appropriate parameter to the single line of interface configuration data. For example:

```
ifconfig_sfxge<n>="inet <IPv4 address> netmask <netmask> tso"
```

TCP and IP checksum offloads must be enabled for TSO to work.

NOTE: Solarflare recommend that you do not disable this setting.

TCP Large Receive Offload (LRO)

TCP Large Receive Offload (LRO) is a feature whereby the adapter coalesces multiple packets received on a TCP connection into a single larger packet before passing this onto the network stack for receive processing. This reduces CPU utilization and improves peak throughput when the CPU is fully utilized. The effectiveness of LRO is bounded by the interrupt moderation delay, and is limited if interrupt moderation is disabled (see Interrupt Moderation (Interrupt Coalescing) on page 357). Enabling LRO does not itself negatively impact latency.

The Solarflare network adapter driver enables LRO by default.

LRO is changed dynamically using `ifconfig`.

To enable LRO, where `sfxge<n>` is the interface name:

```
# ifconfig sfxge<n> lro
```

To disable LRO:

```
# ifconfig sfxge<n> -lro
```

Verification of the LRO setting may be performed by running `ifconfig` with no options and checking the LRO value associated with the interface. The change in LRO can be made to persist across reboots by editing the `/etc/rc.conf` file and adding the appropriate parameter to the single line of interface configuration data. For example:

```
ifconfig_sfxge<n>="inet <IPv4 address> netmask <netmask> lro"
```

NOTE: LRO should **NOT** be enabled when using the host to forward packets from one interface to another. For example, if the host is performing IP routing.

TCP Protocol Tuning

TCP Performance can also be improved by tuning kernel TCP settings. Settings include adjusting send and receive buffer sizes, connection backlog, congestion control, etc.

Initial buffering settings should provide good performance. However for certain applications, tuning buffer settings can significantly benefit throughput. To change buffer settings, adjust the `tcp_rmem` and `tcp_wmem` using the `sysctl` command:

- Receive buffering:

```
sysctl net.ipv4.tcp_rmem="<min> <default> <max>"
```

- Transmit buffering:

```
sysctl net.ipv4.tcp_wmem="<min> <default> <max>"
```

(`tcp_rmem` and `tcp_wmem` can also be adjusted for IPV6 and globally with the net.ipv6 and net.core variable prefixes respectively).

Typically it is sufficient to tune just the max buffer value. It defines the largest size the buffer can grow to. Suggested alternate values are max=500000 (1/2 Mbyte). Factors such as link latency, packet loss and CPU cache size all influence the affect of the max buffer size values. The minimum and default values can be left at their defaults minimum=4096 and default=87380.

See https://wiki.freebsd.org/NetworkPerformanceTuning for more details.

Receive Side Scaling (RSS)

Solarflare adapters support Receive Side Scaling (RSS). RSS enables packet receive-processing to scale with the number of available CPU cores. RSS requires a platform that supports MSI-X interrupts.

When RSS is enabled the controller uses multiple receive queues to deliver incoming packets. The receive queue selected for an incoming packet is chosen to ensure that packets within a TCP stream are all sent to the same receive queue – this ensures that packet-ordering within each stream is maintained. Each receive queue has its own dedicated MSI-X interrupt which ideally should be tied to a dedicated CPU core. This allows the receive side TCP processing to be distributed amongst the available CPU cores, providing a considerable performance advantage over a conventional adapter architecture in which all received packets for a given interface are processed by just one CPU core.

By default the driver enables RSS and configures one RSS Receive queue per CPU core. The number of RSS Receive queues is changed using kenv to modify the kernel environment variable `hw.sfxge.<n>.max_rss_channels`. The driver must be reloaded after the change using the `kldload` command.

To set `<m>` RSS Receive queues, where `sfxge<n>` is the interface name:

```
kenv hw.sfxge.<n>.max_rss_channels=<m>
kldload /boot/modules/sfxge.ko
```


Sometimes, it can be desirable to disable RSS when running single stream applications, since all interface processing may benefit from taking place on a single CPU. To do so, set a single RSS Receive queue:

```
kenv hw.sfxge.<n>.max_rss_channels=1
kldload /boot/modules/sfxge.ko
```

The change in RSS Receive queues can be made to persist across reboots by editing the file `/boot/loader.conf` and adding `hw.sfxge.<n>.max_rss_channels=<m>` on a new line.

If no MSI/MSI-X interrupts are available then the driver will fall-back to use a single legacy interrupt. RSS will be unavailable for that port.

NOTE: RSS also works for UDP packets. For UDP traffic the Solarflare adapter will select the Receive CPU based on IP source and destination addresses. Solarflare adapters support IPv4 and IPv6 RSS.

Other Considerations

PCI Express Lane Configurations

The PCI Express (PCIe) interface used to connect the adapter to the server can function at different speeds and widths. This is independent of the physical slot size used to connect the adapter. The possible widths are multiples x1, x2, x4, x8 and x16 lanes of (2.5Gbps for PCIe Gen 1, 5.0 Gbps for PCIe Gen 2 and 8.0Gbps for PCIe Gen 3) in each direction. *Solarflare adapters are designed for x8 or x16 lane operation*.

On some server motherboards, choice of PCIe slot is important. This is because some slots (including those that are physically x8 or x16 lanes) may only electrically support x4 lanes. In x4 lane slots, Solarflare PCIe adapters will continue to operate, but not at full speed. The Solarflare driver will warn if it detects that the adapter is plugged into a PCIe slot which electrically has fewer than x8 lanes.

SFN5xxx and SFN6xxx Solarflare adapters require a PCIe Gen 2 x8 slot for optimal operation. Solarflare SFN7xxx series adapters require a PCIe Gen 3 x8 or x16 slot for optimal performance. The Solarflare driver will warn if it detects that the adapter is placed in a sub-optimal slot.

In addition, the latency of communications between the host CPUs, system memory and the Solarflare PCIe adapter may be PCIe slot dependent. Some slots may be "closer" to the CPU, and therefore have lower latency and higher throughput. If possible, install the adapter in a slot which is local to the desired NUMA node

Please consult your server user guide for more information.

CPU Power Management

The `powerd` service controls the CPU clock speed dynamically according to current processing demand. For latency sensitive applications, where the application switches between having packets to process and having periods of idle time waiting to receive a packet, dynamic clock speed control may increase packet latency. Solarflare recommend disabling the `powerd` service if minimum latency is the main consideration.

To stop `powerd`, type:

```
/etc/rc.d/powerd stop
```

To disable `powerd` across reboots, ensure this setting is present in `/etc/rc.conf`:

```
powerd_enable="NO"
```

Memory bandwidth

Many chipsets use multiple channels to access main system memory. Maximum memory performance is only achieved when the chipset can make use of all channels simultaneously. This should be taken into account when selecting the number of memory modules (DIMMs) to populate in the server. For optimal memory bandwidth in the system, it is likely that:

- all DIMM slots should be populated
- all NUMA nodes should have memory installed.

Please consult the motherboard documentation for details.

Server Motherboard, Server BIOS, Chipset Drivers

Tuning or enabling other system capabilities may further enhance adapter performance. Readers should consult their server user guide. Possible opportunities include tuning PCIe memory controller (PCIe Latency Timer setting available in some BIOS versions).

Tuning Recommendations

The following tables provide recommendations for tuning settings for different applications.

Throughput - Table 95

Latency - Table 96

Forwarding - Table 97

Recommended Throughput Tuning

Table 95 shows recommended tuning settings for throughput:

Table 95: Throughput Tuning Settings

Tuning Parameter	How?
MTU Size	Configure to maximum supported by network: `ifconfig sfxge<n> mtu <size>`
Interrupt moderation	Leave at default (Enabled).
TCP/IP Checksum Offload	Leave at default (Enabled).
TCP Segmentation Offload	Leave at default (Enabled).
TCP Large Receive Offload	Leave at default (Enabled).
TCP Protocol Tuning	Leave at default
Receive Side Scaling (RSS)	Application dependent
Buffer Allocation Method	Leave at default. Some applications may benefit from specific setting.
PCI Express Lane Configuration	Ensure the adapter is in an x8 slot (2.0 or later), and that current speed (not the supported speed) reads back as “x8 and 5GT/s”, or “x8 and 8GT/s”, or “x8 and Unknown”.
CPU Power Management	Leave enabled
Memory bandwidth	Ensure memory utilizes all memory channels on system motherboard

Recommended Latency Tuning

Table 96 shows recommended tuning settings for latency:

Table 96: Latency Tuning Settings

Tuning Parameter	How?
MTU Size	Configure to maximum supported by network: `ifconfig sfxge<n> mtu <size>`
Interrupt moderation	Disable with: `sysctl dev.sfxge.<n>.int_mod=0`
TCP/IP Checksum Offload	Leave at default (Enabled).
TCP Segmentation Offload	Leave at default (Enabled).

Table 96: Latency Tuning Settings

Tuning Parameter	How?
TCP Large Receive Offload	Disable with: `ifconfig sfxge<n> -lro`
TCP Protocol Tuning	Leave at default, but changing does not impact latency
Receive Side Scaling	Application dependent
Buffer Allocation Method	Leave at default. Some applications may benefit from specific setting.
PCI Express Lane Configuration	Ensure the adapter is in an x8 slot (2.0 or later), and that current speed (not the supported speed) reads back as “x8 and 5GT/s”, or “x8 and 8GT/s”, or “x8 and Unknown”.
CPU Power Management	Disable with: `/etc/rc.d/powerd stop`
Memory bandwidth	Ensure memory utilizes all memory channels on system motherboard

Recommended Forwarding Tuning

Table 97 shows recommended tuning settings for forwarding:

Table 97: Forwarding Tuning Settings

Tuning Parameter	How?
MTU Size	Configure to maximum supported by network: `ifconfig sfxge<n> mtu <size>`
Interrupt moderation	Configure an explicit interrupt moderation interval with: `sysctl dev.sfxge.<n>.int_mod=150`
TCP/IP Checksum Offload	Leave at default (Enabled).
TCP Segmentation Offload	Leave at default (Enabled).
TCP Large Receive Offload	Disable with: `ifconfig sfxge<n> -lro`
TCP Protocol Tuning	Leave at default
Receive Side Scaling (RSS)	Leave at default

Table 97: Forwarding Tuning Settings

Tuning Parameter	How?
Buffer Allocation Method	Leave at default. Some applications may benefit from specific setting.
PCI Express Lane Configuration	Ensure the adapter is in an x8 slot (2.0 or later), and that current speed (not the supported speed) reads back as “x8 and 5GT/s”, or “x8 and 8GT/s”, or “x8 and Unknown”.
CPU Power Management	Leave enabled
Memory bandwidth	Ensure memory utilizes all memory channels on system motherboard

7.11 Module Parameters

Table 98 lists the available parameters in the Solarflare FreeBSD driver module:

- all parameters have a `hw.sfxge.` prefix
- for example, the full name of the parameter shown as `rx_ring` is `hw.sfxge.rx_ring`:

Table 98: Driver Module Parameters

Parameter	Description	Possible Value	Default Value
`rx_ring`	Size of Rx and Tx rings (maximum number of descriptors) per queue. Values used by the driver (default or specified when module is loaded) can be obtained using the same sysctl.	512, 1024, 2048, 4096	1024
`tx_ring`	Size of Rx and Tx rings (maximum number of descriptors) per queue. Values used by the driver (default or specified when module is loaded) can be obtained using the same sysctl.	512, 1024, 2048	1024
`lro.table_size`	Size of the LRO hash table. Must be a power of 2.	uint	128
`lro.chain_max`	Maximum length of chains in the LRO hash table.	uint	20
`lro.idle_ticks`	Time (in jiffies) after which an idle connection's LRO state is discarded.	uint	101

Table 98: Driver Module Parameters

Parameter	Description	Possible Value	Default Value
`lro.slow_start_packets`	Number of packets that must pass in-order before starting LRO.	uint	20000
`lro.loss_packets`	Number of packets that must pass in-order following loss before restarting LRO.	uint	20
`tx_dpl_get_max`	Maximum number of packets queued in the software *get-list* for a transmit queue. The get-list is used to get packets to be put onto the Tx ring. It should be big enough to avoid drops of locally generated TCP packets when many (1000+) streams are running in parallel. Accessing this list requires the transmit queue lock. If a packet is dropped because this limit has been exceeded, the sender gets an `ENOBUFS` error, and the `tx_get_overflow` counter grows.	uint	65536
`tx_dpl_get_non_tcp_max`	Maximum number of non-TCP packets queued in the software *get-list* for a transmit queue. This parameter can restrict utilizing the queue for non-TCP (e.g. UDP) packets, which can easily overflow any queue because there is no back-pressure. If a packet is dropped because this limit has been exceeded, the sender gets an `ENOBUFS` error, and the `tx_get_non_tcp_overflow` counter grows.	uint	1024
`tx_dpl_put_max`	Maximum number of packets queued in the software *put-list* for a transmit queue. The put-list is used to put packets temporarily when the transmit queue lock cannot be obtained. The packets are moved to the get-list as soon as the transmit queue lock is acquired and the queue is served. If a packet is dropped because this limit has been exceeded, the sender gets an `ENOBUFS` error, and the `tx_put_overflow` counter grows.	uint	1024

Table 98: Driver Module Parameters

Parameter	Description	Possible Value	Default Value
`tso_fw_assisted`	Whether to assist TSO using the firmware. Applicable to SFN7*xxx* adapters only.	0|1	1
`<n>.max_rss_channels`	The number of RSS Receive queues for interface `sfxge<n>`. See Receive Side Scaling (RSS) on page 361. The actual number may be lower due to availability of MSI-X interrupts. There is a maximum of 32 MSI-X interrupts across all network devices. If no value is set (the default), the number is limited only by the number of CPUs and MSI-X interrupts.	unit	—

7.12 Kernel and Network Adapter Statistics

The Linux command `sysctl` will display an extensive range of statistics originated from the MAC on the Solarflare network adapter. To display statistics use the following command:

```
sysctl dev.sfxge.<n>.stats
```

where `sfgxe<n>` is the interface name.

Tables below list the complete output from the `sysctl dev.sfxge.<n>.stats` command. See:

- Table 99 on page 368
- Table 100 on page 371
- Table 101 on page 372.

Per port statistics (Table 101 on page 372) are from the physical adapter port. Other statistics are from the specified PCIe function.

NOTE: `sysctl dev.sfxge.<n>.stats` output depends on the features supported by the adapter type.

Table 99: Event queue statistics

Field	Description
ev_all	Total number of events.
ev_rx	Number of packets received by driver.
ev_rx_ok	Number of received packets not discarded.
ev_rx_recovery	Not supported.

Table 99: Event queue statistics

Field	Description
ev_rx_frm_trunc	Number of packets truncated because an internal FIFO is full. As a packet is received it is fed by the MAC into a 128K FIFO. If for any reason the PCI interface cannot keep pace and is unable to empty the FIFO at a sufficient rate, the MAC will be unable to feed more of the packet to the FIFO. In this event the MAC will truncate the packet marking it as such, and discard the remainder. The driver on seeing a 'partial' packet which has been truncated will discard it.
ev_rx_tobe_disc	Number of packets marked by the adapter to be discarded because of one of the following: • mismatched unicast address and unicast promiscuous mode is not enabled • packet is a pause frame • packet has length discrepancy • internal FIFO overflow condition • length < 60 bytes.
ev_rx_pause_frm_err	Number of pause packets received.
ev_rx_buf_owner_id_err	Event caused by internal driver error.
ev_rx_ipv4_hdr_chksum_err	Number of packets received with IP header checksum error.
ev_rx_tcp_udp_chksum_err	Number of packets received with TCP/UDP checksum error.
ev_rx_eth_crc_err	Number of packets received whose CRC did not match the internally generated CRC value.
ev_rx_ip_frag_err	Number of IP fragments received (note this is not an error).
ev_rx_mcast_pkt	Number of IP multicast packets received.
ev_rx_mcast_hash_match	Number of IP multicast packets received which have matched the IP multicast match filter.
ev_rx_tcp_ipv4	Number of TCP/IPv4 packets received.
ev_rx_tcp_ipv6	Number of TCP/IPv6 packets received.

Table 99: Event queue statistics

Field	Description
ev_rx_udp_ipv4	Number of UDP/IPv4 packets received.
ev_rx_udp_ipv6	Number of UDP/IPv6 packets received.
ev_rx_other_ipv4	Number of IPv4 packets received which are not TCP or UDP.
ev_rx_other_ipv6	Number of IPv6 packets received which are not TCP or UDP.
ev_rx_non_ip	Number of packets received which are not IP.
ev_rx_overrun	Number of received packets dropped by receiver because of FIFO overrun.
ev_tx	Number of transmitted packets.
ev_tx_wq_ff_full	Number of transmitted packets dropped because of FIFO overrun.
ev_tx_pkt_err	Number of transmitted packets dropped because of driver error.
ev_tx_pkt_too_big	Number of transmitted packets dropped because of driver error.
ev_tx_unexpected	Number of transmitted packets dropped because of driver error.
ev_global	Internal driver event.
ev_global_phy	Internal driver event.
ev_global_mnt	Internal driver event.
ev_global_rx_recovery	Internal driver event.
ev_driver	Internal driver event.
ev_driver_srm_upd_done	Internal driver event.
ev_driver_tx_descq_fls_done	Internal driver event.
ev_driver_rx_descq_fls_done	Internal driver event.
ev_driver_rx_descq_fls_failed	Internal driver event.
ev_driver_rx_dsc_error	Internal driver event.
ev_driver_tx_dsc_error	Internal driver event.

Table 99: Event queue statistics

Field	Description
ev_drv_gen	Internal driver event.
ev_mcdi_response	Internal driver event.

Table 100: Driver statistics

Field	Description
lro_merges	Number of packets absorbed by LRO.
lro_bursts	Number of bursts spotted by LRO.
lro_slow_start	Number of packets not merged because connection may be in slow-start.
lro_misorder	Number of out-of-order packets seen in tracked streams.
lro_too_many	Incremented when the driver is trying to track too many streams.
lro_new_stream	Number of distinct streams the driver has tracked.
lro_drop_idle	Number of streams discarded because they went idle.
lro_drop_closed	Number of streams that have seen a FIN or RST.
tso_bursts	Number of times TSO transmit invoked by the kernel.
tso_packets	Number of packets sent via the TSO transmit path.
tso_long_headers	Number of packets with headers too long for standard blocks.
tx_collapses	Number of packets with too many fragments collapsed.
tx_drops	Number of packets dropped by the driver because of: • transmit queue in inappropriate state • memory allocation or DMA mapping failures required to handle packet with long header by TSO • mbuf collapse failure.

Table 100: Driver statistics

Field	Description
tx_get_overflow	Number of packets early dropped by the driver because of software transmit queue overflow (see `hw.sfxge.tx_dpl_get_max` and `hw.sfxge.tx_dpl_put_max` in Table 98 on page 366).
tx_put_overflow	Number of packets early dropped by the driver because of software transmit queue overflow (see `hw.sfxge.tx_dpl_get_max` and `hw.sfxge.tx_dpl_put_max` in Table 98 on page 366).
tx_get_non_tcp_overflow	Number of non-TCP packets early dropped by the driver because of software transmit queue limit for non-TCP packets (see `hw.sfxge.tx_dpl_get_non_tcp_max` in Table 98 on page 366).
tx_netdown_drops	Number of packets early dropped by the driver because of link is down.
tso_pdrop_too_many	Number of TSO packets partially dropped by the driver because TSO generates too many segments (most likely because of tiny MSS).
tso_pdrop_no_rsrc	Number of TSO packets partially dropped by the driver because the packet header is too big and requires per-segment memory allocation and DMA mapping which failed.

Table 101: Port statistics

Field	Description
rx_octets	Number of bytes received. Not include collided bytes.
rx_pkts	Number of packets received.
rx_unicst_pkts	Number of unicast packets received.
rx_multicst_pkts	Number of multicast packets received.
rx_brdcst_pkts	Number of broadcasted packets received.
rx_pause_pkts	Number of pause frames received with valid pause op_code.

Table 101: Port statistics

Field	Description
rx_le_64_pkts	Number of packets received where the length is less than or equal to 64 bytes.
rx_65_to_127_pkts	Number of packets received where the length is between 65 and 127 bytes.
rx_128_to_255_pkts	Number of packets received where the length is between 128 and 255 bytes.
rx_256_to_511_pkts	Number of packets received where the length is between 256 and 511 bytes.
rx_512_to_1023_pkts	Number of packets received where the length is between 512 and 1023 bytes.
rx_1024_to_15xx_pkts	Number of packets received where the length is between 1024 and 1518 bytes (1522 with VLAN tag).
rx_ge_15xx_pkts	Number of packets received where the length is between 1518 bytes (1522 with VLAN tag) and 9000 bytes.
rx_errors	Number of packets received with errors.
rx_fcs_errors	Number of packets received with FCS errors.
rx_drop_events	Number of packets dropped by receiver.
rx_false_carrier_errors	Count of the instances of false carrier detected. False carrier is activity on the receive channel that does not result in a packet receive attempt being made.
rx_symbol_errors	Port error condition.
rx_align_errors	Port error condition.
rx_internal_errors	Port error condition.
rx_jabber_pkts	Port error condition.
rx_lane0_char_err	Port error condition.
rx_lane1_char_err	Port error condition.
rx_lane2_char_err	Port error condition.
rx_lane3_char_err	Port error condition.
rx_lane0_disp_err	Port error condition.

Table 101: Port statistics

Field	Description
rx_lane1_disp_err	Port error condition.
rx_lane2_disp_err	Port error condition.
rx_lane3_disp_err	Port error condition.
rx_match_fault	Number of packets received which did not match any filter.
rx_nodesc_drop_cnt	Number of packets dropped by the network adapter because of a lack of RX descriptors in the RX queue. Packets can be dropped by the NIC when there are insufficient RX descriptors in the RX queue to allocate to the packet. This problem occurs if the receive rate is very high and the network adapter receive cycle process has insufficient time between processing to refill the queue with new descriptors. A number of different steps can be tried to resolve this issue: • Disable the irqbalance daemon in the OS • Distribute the traffic load across the available CPU/cores by setting rss_cpus=cores. Refer to Receive Side Scaling section • Increase receive queue size using ethtool.
tx_octets	Number of bytes transmitted.
tx_pkts	Number of packets transmitted.
tx_unicst_pkts	Number of unicast packets transmitted. Includes flow control packets.
tx_multicst_pkts	Number of multicast packets transmitted.
tx_brdcst_pkts	Number of broadcast packets transmitted.
tx_pause_pkts	Number of pause frames transmitted with valid pause op_code.
tx_le_64_pkts	Number of frames transmitted where the length is less than or equal to 64 bytes.
tx_65_to_127_pkts	Number of frames transmitted where the length is between 65 and 127 bytes

Table 101: Port statistics

Field	Description
tx_128_to_255_pkts	Number of frames transmitted where the length is between 128 and 255 bytes
tx_256_to_511_pkts	Number of frames transmitted where the length is between 256 and 511 bytes
tx_512_to_1023_pkts	Number of frames transmitted where length is between 512 and 1023 bytes
tx_1024_to_15xx_pkts	Number of frames transmitted where the length is between 1024 and 1518 bytes (1522 with VLAN tag).
tx_ge_15xx_pkts	Number of frames transmitted where length is between 1518 bytes (1522 with VLAN tag) and 9000 bytes.
tx_errors	Port error condition.
tx_sgl_col_pkts	Port error condition.
tx_mult_col_pkts	Port error condition.
tx_ex_col_pkts	Port error condition.
tx_late_col_pkts	Port error condition.
tx_def_pkts	Port error condition.
tx_ex_def_pkts	Port error condition.

Netstat statistics

The Linux command `netstat` also displays some of these statistics. They are periodically updated from the port and driver statistics. See Table 102:

Table 102: Netstat statistics

Field	Value
Ipkts	rx_pkts
Ierrs	rx_errors
Idrop	0
Ibytes	rx_octets
Opkts	tx_pkts

Table 102: Netstat statistics

Field	Value
Oerrs	tx_errors + tx_drops + get_overflow + get_non_tcp_overflow + put_overflow + netdown_drops + tso_pdrop_too_many + tso_pdrop_no_rsrc
Obytes	tx_octets
Coll	tx_sgl_col_pkts + tx_mult_col_pkts + tx_ex_col_pkts + tx_late_col_pkts

8 SR-IOV Virtualization Using KVM

8.1 Introduction

This chapter describes SR-IOV and virtualization using Linux KVM and the Solarflare SFN7000 series adapters.

SR-IOV enabled on Solarflare adapters provides accelerated cut-through performance and is fully compatible with hypervisor based services and management tools. The advanced design of the Solarflare SFN7000 series adapter incorporates a number of features to support SR-IOV. These features can be summarized as follows:

- PCIe Virtual Functions (VF).

  A PCIe physical function, PF, can support a configurable number of PCIe virtual functions. In total 240 VFs can be allocated between the PFs. The adapter can also support a total of 2048 MSI-X interrupts.

- Layer 2 Switching Capability.

  A layer 2 switch configured in firmware supports the transport of network packets between PCI physical functions (PF), Virtual functions (VF) and the external network. This allows received packets to be replicated across multiple PFs/VFs and allows packets transmitted from one PF to be received on another PF or VF.

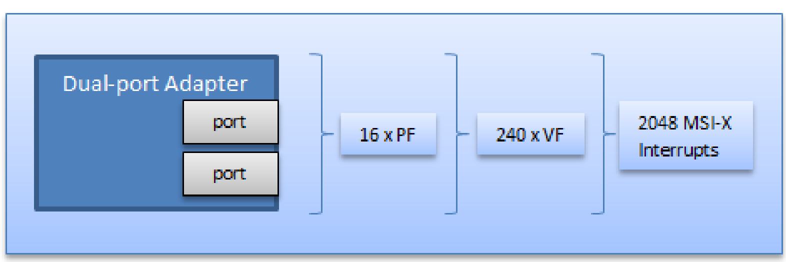


Figure 50: Per Adapter - Configuration Options

Supported Platforms

Host

- Red Hat Enterprise Linux 6.5 - 7.0 KVM

Guest VM

- Red Hat Enterprise Linux 5.x, 6.x and 7.x

Acceleration of guest Virtual Machines (VM) running other (non-Linux) operating systems are not currently supported, however other schemes, for example, a KVM direct bridged configuration using the Windows virtio-net driver could be used.

Driver/Firmware

Features described in the chapter require the following (minimum) Solarflare driver and firmware versions.

```
# ethtool -i eth<N>
driver: sfc
version: 4.4.1.1017
firmware-version: 4.4.2.1011 rx0 tx0
```

The adapter must be using the *full-feature* firmware variant which can be selected using the `sfboot` utility and confirmed with **`rx0 tx0`** appearing after the version number in the output from ethtool as shown above.

The firmware update utility (`sfupdate`) and bootROM configuration tool (`sfboot`) are available in the Solarflare Linux Utilities package (SF-107601-LS issue 28 or later).

Platform support - SR-IOV

BIOS

To use SR-IOV modes, SR-IOV must be enabled in the platform BIOS where the actual BIOS setting can differ between machines, but may be identified as SR-IOV, IOMMU or VT-d and VT-x on an Intel platform.

The following links identify Linux Red Hat documentation for SR-IOV BIOS settings.

https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Virtualization_Deployment_and_Administration_Guide/index.html

https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Virtualization_Administration_Guide/sect-Virtualization-Troubleshooting-Enabling_Intel_VT_and_AMD_V_virtualization_hardware_extensions_in_BIOS.html

There may be other BIOS options which should be enabled to support SR-IOV, for example on DELL servers the following BIOS option must also be enabled:

```
Integrated Devices, SR-IOV Global Enable
```


Users are advised to consult the server vendor BIOS options documentation.

Kernel Configuration

On an Intel platform, the IOMMU must be explicitly enabled by appending `intel_iommu=on` to the kernel line in the `/boot/grub/grub.conf` file. The equivalent setting on an AMD system is `amd_iommu=on`.

Solarflare recommends that users also enable the `pci=realloc` kernel parameter in the `/boot/grub/grub.conf` file. This allows the kernel to reassign addresses to PCIe apertures (i.e. bridges, ports) in the system when the BIOS does not allow enough PCI apertures for the maximum number of supported VFs.

KVM - Interrupt Re-Mapping

To use PCIe VF passthrough, the server must support interrupt re-mapping. If the target server does not support interrupt re-mapping it is necessary to set the following option in a user created file e.g. `kvm_iommu_map_guest.conf` in the `/etc/modprobe.d` directory:

```
[RHEL 6] options kvm allow_unsafe_assigned_interrupts=1
```

```
[RHEL 7] options vfio_iommu_type1 allow_unsafe_assigned_interrupts=1
```

Alternative Routing-ID Interpretation (ARI)

The ARI extension to the PCI Express Base Specification extends the capacity of a PCIe endpoint by increasing the number of accessible functions (PF+VF) from 8, up to 256. Without ARI support - which is a feature of the server hardware and BIOS, a server hosting a virtualized environment will be limited to 8 functions. The Solarflare SFN7000 series adapter can expose up to 16 PFs and 240 VFs per adapter.

Users should consult the appropriate server vendor documentation to ensure that the host server supports ARI.

Supported Adapters

All Solarflare SFN7000 series adapters fully support SR-IOV. Features described in this chapter are not supported by Solarflare SFN5000 or SFN6000 series adapters which support a limited SR-IOV implementation.

The `sfboot` utility allows the user to configure:

- The number of PFs exposed to host and/or Virtual Machine (VM).
- The number VFs exposed to host and/or Virtual Machine (VM).
- The number of MSI-X interrupts assigned to each PF or VF.

The Solarflare implementation uses a single driver (sfc.ko) that binds to both PFs and VFs.

sfboot - Configuration Options

Adapter configuration options are set using the `sfboot` utility *v4.5.0 or later* from the Solarflare Linux Utilities package (SF-107601-LS issue 28 or later). The firmware variant must be set to full-feature / Virtualization.

```
# sfboot firmware-variant=full-feature
```

To check the current adapter configuration run the `sfboot` command:

```
# sfboot
Solarflare boot configuration utility [v4.5.0]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-2005

eth5:
  Boot image                            Option ROM only
    Link speed                          Negotiated automatically
    Link-up delay time                  5 seconds
    Banner delay time                   2 seconds
    Boot skip delay time                5 seconds
    Boot type                           Disabled
  Physical Functions per port           1
  MSI-X interrupt limit                 32
  Number of Virtual Functions           2
  VF MSI-X interrupt limit              8
  Firmware variant                      full feature / virtualization
  Insecure filters                      Disabled
  MAC spoofing                          Disabled
  VLAN tags                             None
  Switch mode                           SRIOV
```

For some configuration option changes using sfboot, the server must be power cycled (power off/power on) before the changes are effective. sfboot will display a warning when this is required.

Table 103 identifies `sfboot` SR-IOV configurable options.

Table 103: sfboot - SR-IOV options

Option	Default Value	Description
pf-count=<n>	1	Number of PCIe PFs per physical port. MAC address assignments may change, after next reboot, following changes with this option.
pf-vlans	None	A comma separated list of VLAN tags for each PF. sfboot pf-vlans=0,100,110,120 The first tag is assigned to the first PF, thereafter tags are assigned to PFs in (lowest) MAC address order.

Table 103: sfboot - SR-IOV options

Option	Default Value	Description
mac-spoofing	disabled	If enabled, non-privileged functions may create unicast filters for MAC addresses that are not associated with themselves. This should be used when using bonded interfaces where a bond slave inherits the bond master hardware address.
msix-limit=<n>	32	Number of MSI-X interrupts assigned to each PF. The adapter supports a maximum 2048 interrupts. The specified value for a PF must be a power of 2.
switch-mode=<mode>	default	Specifies the mode of operation that the port will be used in: `default` - single PF created, zero VFs created. `sriov` - SR-IOV enabled, single PF created, VFs configured with `vf-count`. `partitioning` - PFs configured with `pf-count`, VFs configured with `vf-count`. See NIC Partitioning on page 58 for details. `partitioning-with-sriov` - SR-IOV enabled, PFs configured with `pf-count`, VFs configured with `vf-count`. See NIC Partitioning on page 58 for details. `pfiov` - PFIOV enabled, PFs configured with `pf-count`, VFs not supported. Layer 2 switching between PFs.
vf-count=<n>	240	Number of virtual functions per PF.

Table 103: sfboot - SR-IOV options

Option	Default Value	Description
vf-msix-limit=<n>	8	Number of MSI-X interrupts per VF. The adapter supports a maximum 2048 interrupts. The specified value for a PF must be a power of 2.
insecure_filters= <enabled|disabled>	disabled	When enabled, a function (PF or VF) can insert filters not qualified by its own permanent MAC address.

8.2 SR-IOV

In the simplest of SR-IOV supported configurations each physical port is exposed as a single PF (adapter default) and up to 240 VFs.

The Solarflare net driver (sfc.ko) will detect that PF/VFs are present from the sfboot configuration and automatically configure the virtual adapters and virtual ports as required.

Adapter firmware will also configure the firmware switching functions allowing packets to pass between PF and VFs or from VF to VF.

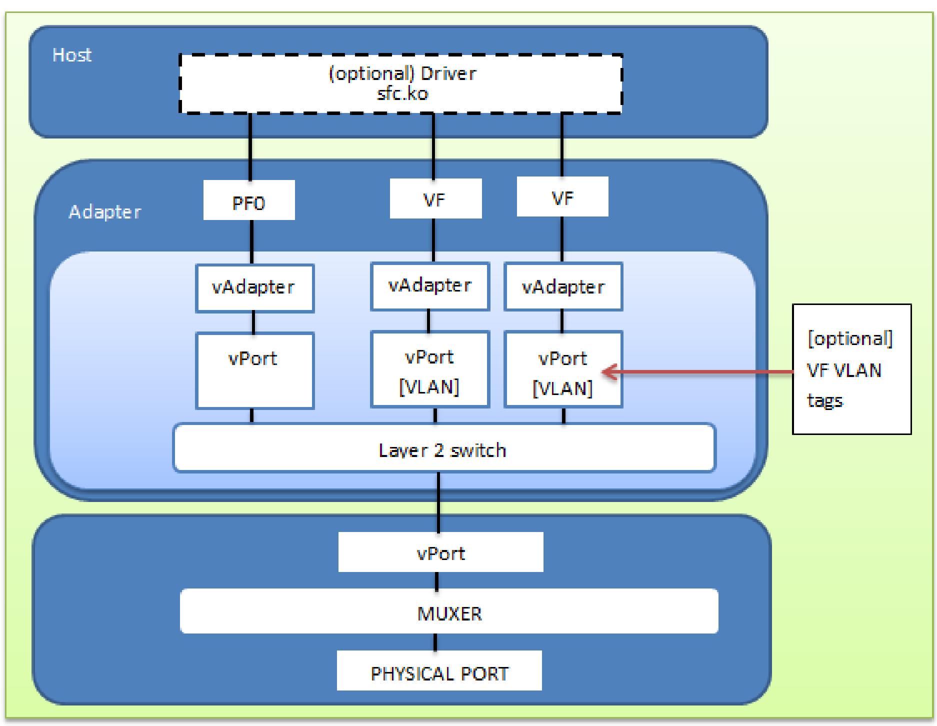


Figure 51: SR-IOV - Single PF, Multiple VFs

- With no VLAN configuration, the PFs and VFs are in the same Ethernet layer 2 broadcast domain i.e. a packet broadcast from the PF would be received by all VFs. VLAN tags can optionally be assigned to VFs using standard libvirt commands.
- The L2 switch supports replication of received/transmitted broadcast packets to all functions.
- The L2 switch will replicate received/transmitted multicast packets to all functions that have subscribed.
- The MUXER function is a firmware enabled layer2 switching function for transmit and receive traffic.

In the example above there are no virtual machines (VM) created. Network interfaces for the PF and each VF will appear in the host. An sfc NIC driver loaded in the host will identify the PF and each VF as individual network interfaces.

SR-IOV Configuration

Ensure SR-IOV and the IOMMU are enabled on the host server kernel command line - Refer to Platform support - SR-IOV on page 378.

1 The example configures 1 PF per port (default), 2 VFs per PF):

```
sfboot switch-mode=sriov pf-count=1 vf-count=2
Solarflare boot configuration utility [v4.5.0]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-
2005

eth8:
  Boot image                                    Option ROM only
    Link speed                                  Negotiated automatically
    Link-up delay time                          5 seconds
    Banner delay time                           2 seconds
    Boot skip delay time                        5 seconds
    Boot type                                   Disabled
  Physical Functions per port                   1
  MSI-X interrupt limit                         32
  Number of Virtual Functions                   2
  VF MSI-X interrupt limit                      8
  Firmware variant                              full feature / virtualization
  Insecure filters                              Disabled
  MAC spoofing                                  Disabled
  VLAN tags                                     None
  Switch mode                                   SRIOV
```

2 Create VFs - see Enabling Virtual Functions on page 399.

3 The server should be cold rebooted following changes using `sfboot`. Following the reboot, The PF and VFs will be visible in the host using the `ifconfig` command and `lspci` (the output below is from a dual-port adapter. VFs are shown in bold text):

```
# lspci -d1924:
03:00.0 Ethernet controller: Solarflare Communications SFC9120 (rev 01)
03:00.1 Ethernet controller: Solarflare Communications SFC9120 (rev 01)
```



```
03:00.2 Ethernet controller: Solarflare Communications Device 1903 (rev 01)
03:00.3 Ethernet controller: Solarflare Communications Device 1903 (rev 01)
03:00.4 Ethernet controller: Solarflare Communications Device 1903 (rev 01)
03:00.5 Ethernet controller: Solarflare Communications Device 1903 (rev 01)
```

4 To identify which physical port a given network interface is using:

```
# cat /sys/class/net/eth<N>/device/physical_port
```

5 To identify which PF a given VF is associated with use the following command (in this example there are 4 VFs assigned to PF eth4):

```
# ip link show

19: eth4: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN qlen 1000
    link/ether 00:0f:53:21:00:61 brd ff:ff:ff:ff:ff:ff
    vf 0 MAC 76:c1:36:0a:be:2b
    vf 1 MAC 1e:b8:a8:ea:c7:fb
    vf 2 MAC 52:6e:32:3d:50:85
    vf 3 MAC b6:ad:a0:56:39:94
```

MAC addresses beginning `00:0f:53` are Solarflare designated hardware addresses. MAC addresses assigned to VFs in the above example output have been randomly generated by the host. MAC addresses visible to the host will be replaced by libvirt-generated MAC addresses in a VM.

8.3 KVM Network Architectures

This section identifies SR-IOV and the Linux KVM virtualization infrastructure configurations to consume adapter port Physical Functions (PF) and Virtual Functions (VF).

- KVM libvirt Bridged on page 384
- KVM Direct Bridged on page 388
- KVM Libvirt Direct Passthrough on page 391
- KVM Libvirt Network Hostdev on page 394
- General Configuration on page 399
- Enabling Virtual Functions on page 399

When migration is not a consideration, Solarflare recommends the network-hostdev configuration for highest throughput and lowest latency performance

KVM libvirt Bridged

The traditional method of configuring networking in KVM virtualized environments uses the para-virtualized (PV) driver, `virtio-net`, in the virtual machine and the standard Linux bridge in the host.

The bridge emulates a layer 2 learning switch to replicate multicast and broadcast packets in software and supports the transport of network traffic between VMs and the physical port.

This configuration uses standard Linux tools for configuration and needs only a virtualized environment and guest operating system.

Performance (latency/throughput) will not be as good as a network-hostdev configuration because network traffic must pass via the host kernel.

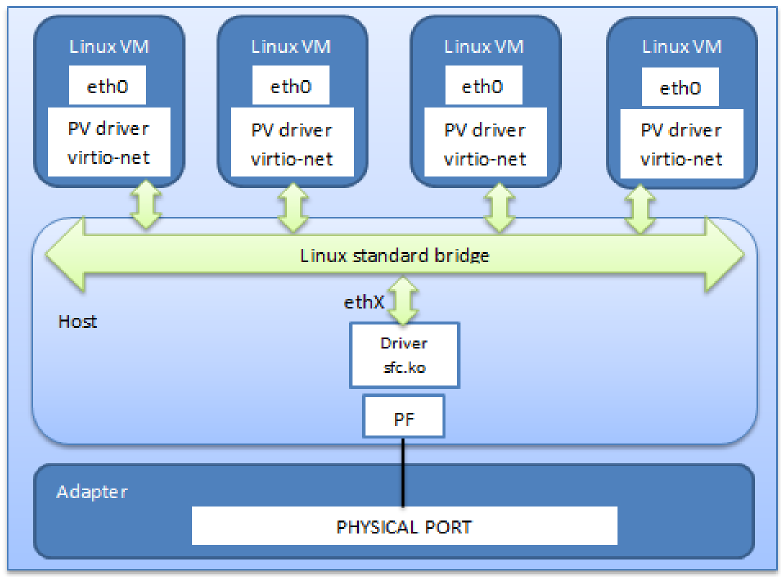


Figure 52: KVM - libvirt bridged

KVM libvirt bridged - Configuration

1 Ensure the Solarflare adapter driver (sfc.ko) is installed on the host.

2 In the host, configure the PF:

```
# sfboot switch-mode=default pf-count=1
```

The `sfboot` settings shown above are the default (shipping state) settings for the SFN7000 series adapter. A cold reboot of the server is only required when changes are made using `sfboot`.

3 Create virtual machines:

VMs can be created from the standard Linux `virt-manager` GUI interface or the equivalent `virsh` command line tool. As root, run the command `virt-manager` from a terminal to start the GUI interface. A VM can also be created from an existing VM XML file.

The following procedure assumes the VM is created. The example procedure will create a bridge 'br1' and network 'host-network' to connect the VM to the Solarflare adapter via the bridge.

4 Define a bridge in `/etc/sysconfig/network-scripts/ifcfg-br1`

```
DEVICE=br1
TYPE=Bridge
BOOTPROTO=none
ONBOOT=yes
DELAY=0
NM_CONTROLLED=no
```


5 Associate the bridge with the required Solarflare PF (HWADDDR) in a config file in `/etc/sysconfig/network-scripts/ifcfg-eth4` (this example uses eth4):

```
DEVICE=eth4
TYPE=Ethernet
HWADDR=00:0F:53:21:00:60
BOOTPROTO=none
ONBOOT=yes
BRIDGE=br1
```

6 Bring up the bridge:

```
# service network restart
```

7 The bridge will be visible in the host using the `ifconfig` command:

```
# ifconfig -a
br1       Link encap:Ethernet  HWaddr 00:0F:53:21:00:60
          inet6 addr: fe80::20f:53ff:fe21:60/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:170 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:55760 (54.4 KiB)  TX bytes:468 (468.0 b)
```

8 Define a network in an XML file i.e. `host-network.xml`:

```
<network>
  <name>host-network</name>
  <forward mode='bridge'/>
  <bridge name="br1"/>
</network>
```

9 Define and start the network using `virsh net-<option>` commands:

```
# virsh net-define host-network.xml
Network host-network defined from host-network.xml

# virsh net-start host-network
Network host-network started

# virsh net-autostart host-network
Network host-network marked as autostarted

# virsh net-list --all
Name                 State      Autostart     Persistent
--------------------------------------------------
default              active     yes           yes
host-network         active     yes           yes
```

10 On the host machine, edit the VM XML file:

```
# virsh edit <vmname>
```

11 Add the network component to the VM XML file:

```
<interface type='network'>
  <source network='host-network'/>
  <model type='virtio'/>
</interface>
```

12 Restart the VM after editing the XML file.

```
# virsh start <vmname>
```


13 The bridged interface is visible in the VM when viewed from the GUI Virtual Machine Manager:

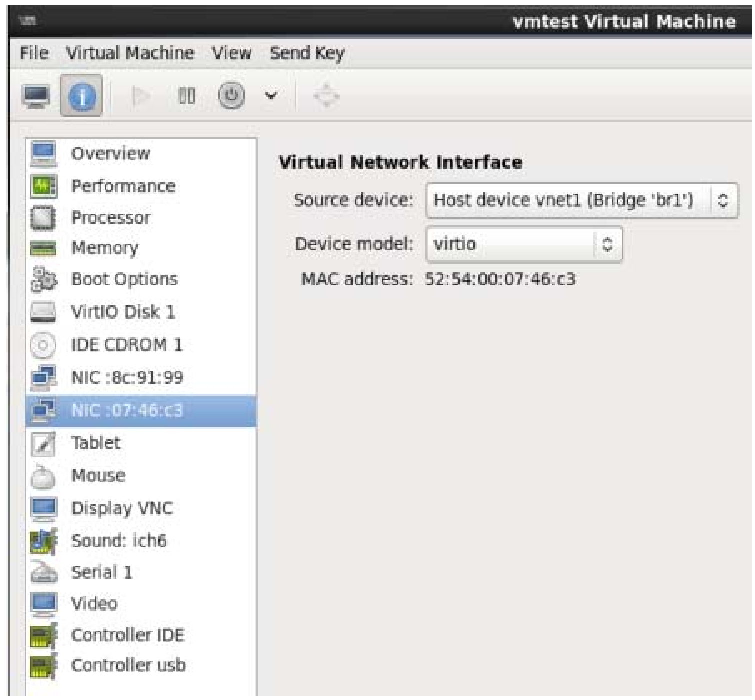


Figure 53: Virtual Machine Manager - Showing the network/bridged interface

XML Description

The following extract is from the VM XML file after the configuration procedure has been applied (line numbers have been added for ease of description):

```
1. <interface type='bridge'>
2.   <mac address='52:54:00:96:0a:8a'/>
3.   <source bridge='br1'/>
4.   <model type='virtio'/>
5.   <address type='pci' domain='0x0000' bus='0x00' slot='0x07'  function='0x0'/>
6. </interface>
```

1 Interface type must be specified by the user as ‘bridge’.

2 The MAC address. If not specified by the user this will be automatically assigned a random MAC address by libvirt.

3 The source bridge as created in configuration step 4 above.

4 Model type must be specified by the user as ‘virtio’.

5 The PF PCIe address (as known by the guest) will be added automatically by libvirt.

For further information about the direct bridged configuration and XML formats, refer to the following links: http://libvirt.org/formatdomain.html#elementsNICSBridge

KVM Direct Bridged

In this configuration multiple macvtap interfaces are bound over the same PF. For each VM created, libvirt will automatically instantiate a macvtap driver instance and the macvtap interfaces will be visible on the host.

Where the KVM libvirt bridged configuration uses the standard Linux bridge, a direct bridged configuration bypasses this providing an internal bridging function and increasing performance.

When using macvtap there is no link state propagation to the guest which is unable to identify if a physical link is up or down.

Macvtap does not currently forward multicast joins from the guests to the underlying network driver with the result that all multicast traffic received by the physical port is forwarded to all guests. Due to this limitation this configuration is not recommended for deployments that use a non-trivial amount of multicast traffic.

Guest migration is fully supported as there is no physical hardware state in the VM guests. A guest can be migrated to a host using a different VF or a host without an SR-IOV capable adapter.

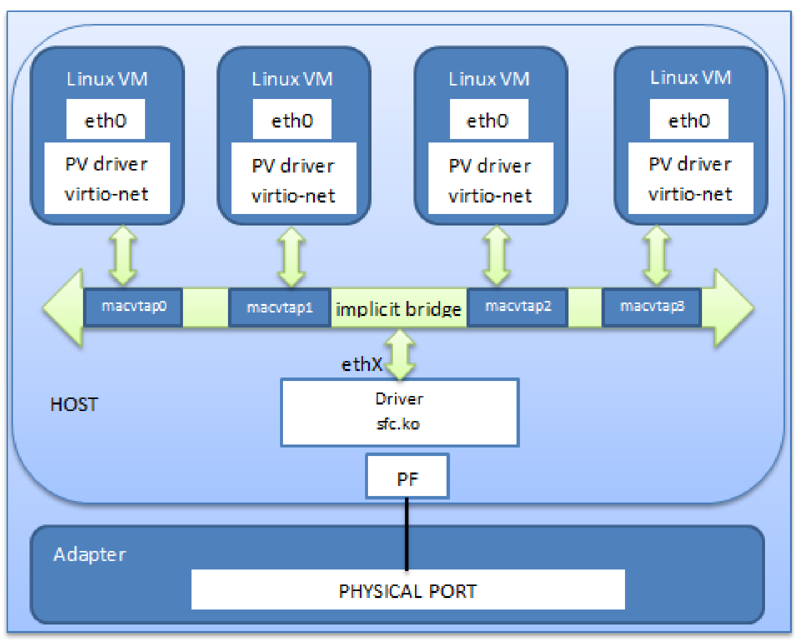


Figure 54: KVM - direct bridged

KVM direct Bridged - Configuration

1 Ensure the Solarflare adapter driver (sfc.ko) is installed on the host.

2 In the host, configure the PF.

```
# sfboot switch-mode=default pf-count=1
```

The `sfboot` settings shown above are the default (shipping state) settings for the SFN7000 series adapter. A cold reboot of the server is only required when changes are made using `sfboot`.

3 Create virtual machines:

VMs can be created from the standard Linux `virt-manager` GUI interface or the equivalent `virsh` command line tool. As root, run the command `virt-manager` from a terminal to start the GUI interface. A VM can also be created from an existing VM XML file.

The following procedure assumes the VM is created. The example procedure will create an interface configuration file and connect the VM directly to the Solarflare adapter.

4 Create a configuration file for the required Solarflare PF (`HWADDR`) in a config file in `/etc/sysconfig/network-scripts/ifcfg-eth4` (this example uses eth4):

```
DEVICE=eth4
TYPE=Ethernet
HWADDR=00:0F:53:21:00:60
BOOTPROTO=none
ONBOOT=yes
```

5 Bring up the interface:

```
# service network restart
```

6 On the host machine, edit the VM XML file:

```
# virsh edit <vmname>
```

7 Add the interface component to the VM XML file:

```
<interface type='direct'>
   <source dev='eth4' mode='bridge'/>
   <model type=’virtio’/>
</interface>
```

8 Restart the VM after editing the XML file.

```
# virsh start <vmname>
```

9 The bridged interface is visible when viewed from the GUI Virtual Machine Manager:

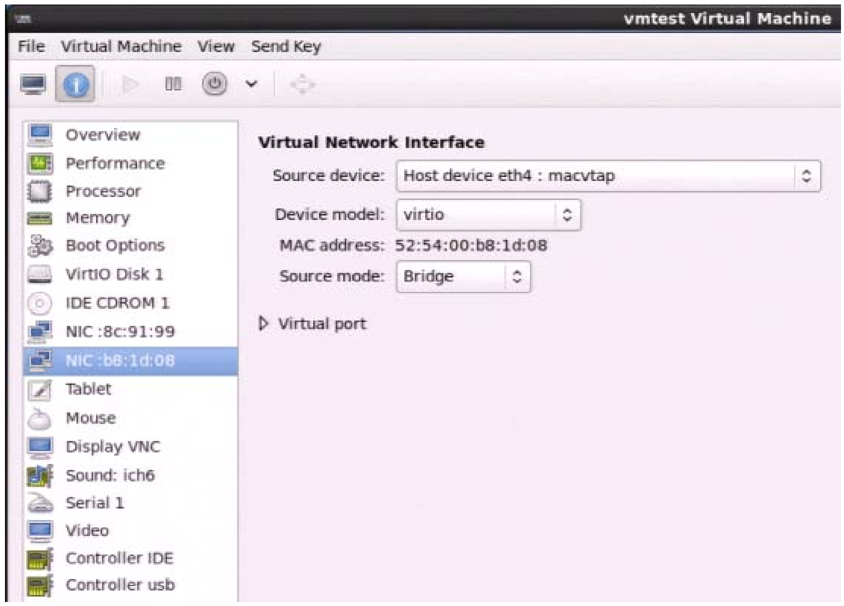


Figure 55: Virtual Machine Manager - Showing the direct bridged interface

XML Description

The following extract is from the VM XML file after the configuration procedure has been applied (line numbers have been added for ease of description):

```
1.  <interface type='direct'>
2.   <mac address='52:54:00:db:ab:ca'/>
3.   <source dev='eth4' mode='bridge'/>
4.   <model type='virtio'/>
5.   <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0'/>
6.  </interface>
```

1 Interface type must be specified by the user as 'direct'.

2 The MAC address. If not specified by the user this will be automatically assigned a random MAC address by libvirt.

3 The source dev is the interface identifier from the host - added by the user. The user should also specify the mode which must be 'bridge'.

4 If not specified by the user, the model type will be automatically assigned by libvirt when the guest is started. Use virtio for best performance.

5 The PF PCIe address (as known by the guest) will be added automatically by libvirt

For further information about the direct bridged configuration and XML formats, refer to the following link:

http://libvirt.org/formatdomain.html#elementsNICSBridge

KVM Libvirt Direct Passthrough

Using a libvirt direct-passthrough configuration, VFs are used in the host OS to provide network acceleration for guest VMs. The guest continues to use a paravirtualized driver and is unaware this is backed with a VF from the network adapter.

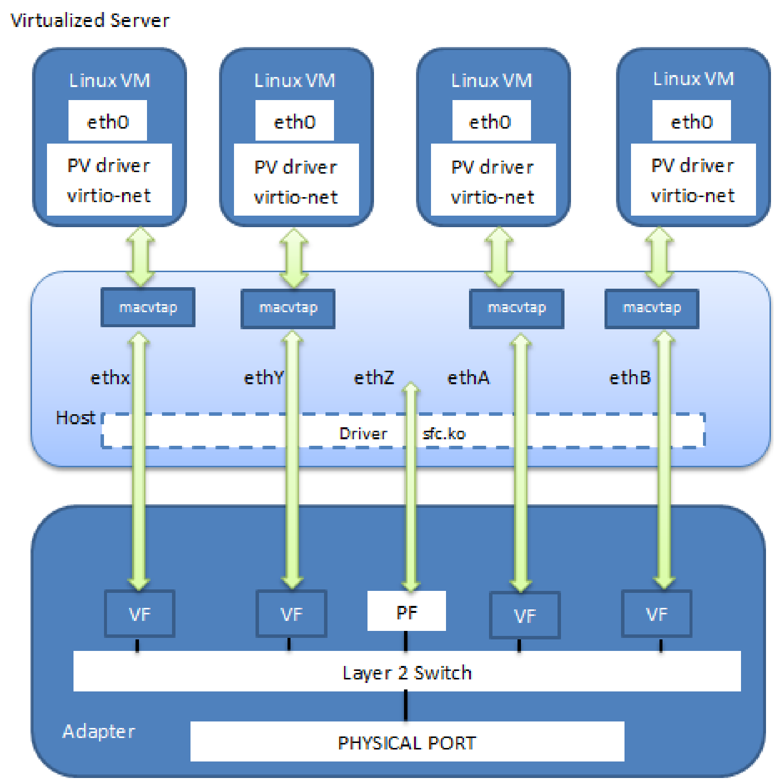


Figure 56: SR-IOV VFs used in the host OS

- The Solarflare net driver is bound over the top of each VF.
- Each macvtap interface is implicitly created by libvirt over a single VF network interface and is not visible to the host OS.
- Each macvtap instance builds over a different network interface - so there is no implicit macvtap bridge.
- Macvtap does not currently forward multicast joins from the guests to the underlying network driver with the result that all multicast traffic received by the physical port is forwarded to all guests. Due to this limitation this configuration is not recommended for deployments that use a non-trivial amount of multicast traffic.
- Guest migration is fully supported as there is no physical hardware state in the VM guests. A guest can be reconfigured to a host using a different VF or a host without an SR-IOV capable adapter.
- The MAC address from the VF is passed through to the para-virtualized driver.

- Because there is no VF present in a VM, Onload and other Solarflare applications such as SolarCapture cannot be used in the VM.

KVM Libvirt Direct Passthrough - Configuration

1 Ensure the Solarflare adapter driver (sfc.ko) is installed on the host.

2 In the host, configure the switch-mode, PF and VFs:

```
# sfboot switch-mode=sriov pf-count=1 vf-count=4
```

A cold reboot of the server is required when changes are made using `sfboot`.

3 Create VFs in the host (example uses PF eth4):

```
echo 2 > /sys/class/net/eth4/device/sriov_numvfs
cat /sys/class/net/eth4/device/sriov_totalvfs
```

For Linux versions earlier than RHEL6.5 see Enabling Virtual Functions on page 399.

4 PFs and VFs will be visible using the lspci command (VFs in **bold**):

```
# lspci -D -d1924:
0000:03:00.0 Ethernet controller: Solarflare Communications SFC9120
0000:03:00.1 Ethernet controller: Solarflare Communications SFC9120
0000:03:00.2 Ethernet controller: Solarflare Communications Device 1903
0000:03:00.3 Ethernet controller: Solarflare Communications Device 1903
0000:03:00.4 Ethernet controller: Solarflare Communications Device 1903
0000:03:00.5 Ethernet controller: Solarflare Communications Device 1903
```

VFs will also be listed using the `ifconfig` command (abbreviated output below, from a dual port adapter, shows 2 x PF and 4 x VF. (pf-count=1 vf-count=2). VFs are shown in **bold**).

```
eth4      Link encap:Ethernet  HWaddr 00:0F:53:21:00:60
eth5      Link encap:Ethernet  HWaddr 00:0F:53:21:00:61
eth6      Link encap:Ethernet  HWaddr AE:82:AB:C9:67:49
eth7      Link encap:Ethernet  HWaddr 86:B4:C8:9E:27:D6
eth8      Link encap:Ethernet  HWaddr 72:0B:C7:21:E1:59
eth9      Link encap:Ethernet  HWaddr D2:B7:68:54:35:A5
```

5 Create virtual machines:

VMs can be created from the standard Linux `virt-manager` GUI interface or the equivalent `virsh` command line tool. As root, run the command `virt-manager` from a terminal to start the GUI interface. A VM can also be created from an existing VM XML file.

The following procedure assumes the VM is created. The example procedure will create an interface configuration file for each VF to be passed through to the VM.

6 For each VF to be passed through to a VM, create a configuration file in the `/etc/sysconfig/network-scripts` directory i.e. `ifcfg-eth6`:

```
DEVICE=eth6
TYPE=Ethernet
HWADDR=AE:82:AB:C9:67:49
BOOTPROTO=none
ONBOOT=yes
```


The above example is the file `ifcfg-eth6` and identifies the MAC address assigned to the VF. One file is required for each VF.

7 On the host machine, edit the VM XML file:

```
# virsh edit <vmname>
```

8 Add the interface component to the VM XML file e.g:

```
<interface type='direct'>
  <source dev='eth6' mode='passthrough'/>
  <model type='virtio'/>
</interface>
```

One interface type component is required for each VF.

9 Restart the VM after editing the XML file.

```
# virsh start <vmname>
```

The passed through VF interface is visible when viewed from the GUI Virtual Machine Manager

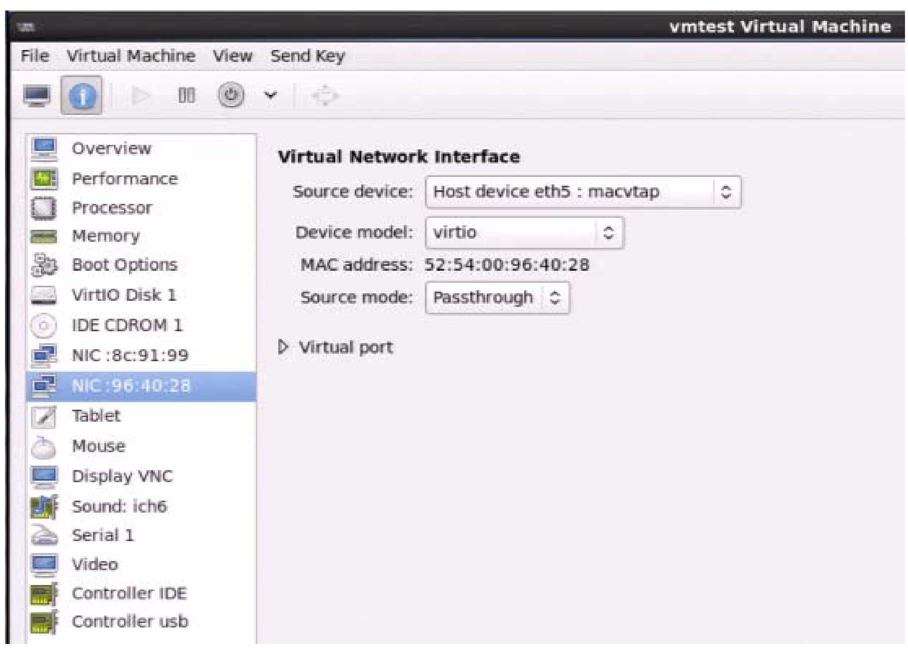


Figure 57: Virtual Machine Manager - Showing the passthrough interface

XML Description

The following (example) extract is from the VM XML file after a VF has been passed through to the guest using the procedure above (line numbers have been added for ease of description):

```
1.  <interface type='direct'>
2.    <mac address='52:54:00:96:40:28'/>
3.    <source dev='eth6' mode='passthrough'/>
4.    <model type='virtio'/>
5.    <address type='pci' domain='0x0000' bus='0x00' slot='0x07'   function='0x0'/>
6.  </interface>
```


1 A description of how the VF interface is managed - added by the user.

2 The MAC address. If not specified by the user this will be automatically assigned a random MAC address by the guest OS. The user can specify a MAC address when editing the XML file.

3 The source dev is the VF interface identifier - added by the user. The user should also specify the mode which must be 'passthrough'.

4 If not specified by the user, the model type will be automatically assigned by libvirt when the guest is started.

5 The VF PCIe address (as known by the guest) will be added automatically by libvirt.

For further information about the direct passthrough configuration and XML formats, refer to the following link: http://libvirt.org/formatdomain.html#elementsNICSDirect

KVM Libvirt Network Hostdev

Network Hostdev exposes VFs directly into guest VMs allowing the data path to fully bypass the host OS and therefore provides maximum acceleration for network traffic.

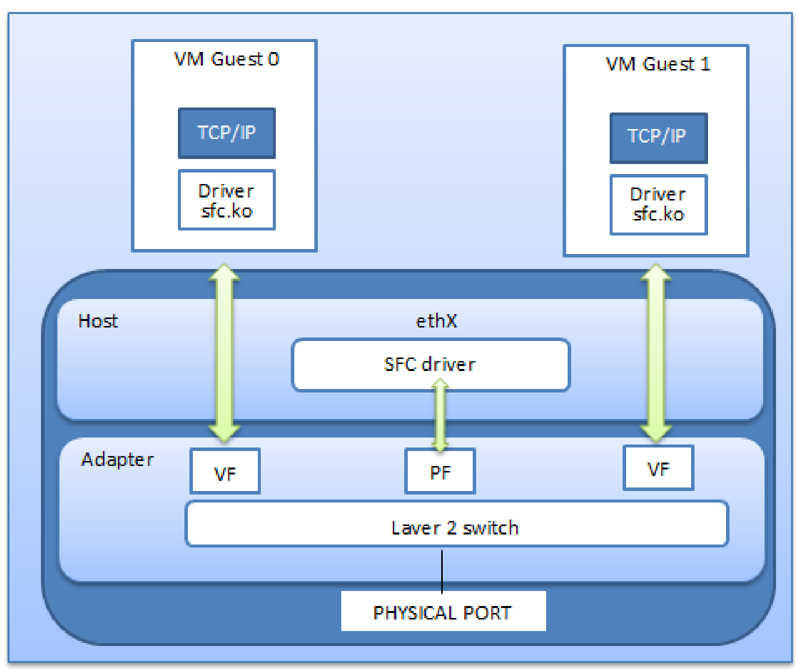


Figure 58: SR-IOV VFs passed to guests

- The hostdev configuration delivers the highest throughput and lowest latency performance. Because the guest is directly linked to the virtual function therefore directly connected to the underlying hardware.

- Migration is not supported in this configuration because the VM has knowledge of the network adapter hardware (VF) present in the server.
- The VF is visible in the guest. This allows applications using the VF interface to be accelerated using OpenOnload or to use other Solarflare applications such as SolarCapture.
- The Solarflare net driver (sfc.ko) needs to be installed in the guest.

KVM Libvirt network hostdev - Configuration

1 Create the VM from the Linux `virt-manager` GUI interface or the `virsh` command line tool.

2 Install Solarflare network driver (sfc.ko) in the guest and host.

3 Create the required number of VFs:

```
# sfboot switch-mode=sriov vf-count=4
```

A cold reboot of the server is required for this to be effective.

4 For the selected PF - configure the required number of VFs e.g:

```
# echo 4 > /sys/class/net/eth8/device/sriov_numvfs
```

5 VFs will now be visible in the host - use `ifconfig` and the `lspci` command to identify the Ethernet interfaces and PCIe addresses (VFs shown below in **bold** text):

```
# lspci -D -d1924:
0000:03:00.0 Ethernet controller: Solarflare Communications SFC9120 (rev 01)
0000:03:00.1 Ethernet controller: Solarflare Communications SFC9120 (rev 01)
0000:03:00.2 Ethernet controller: Solarflare Communications Device 1903 (rev 01)
0000:03:00.3 Ethernet controller: Solarflare Communications Device 1903 (rev 01)
0000:03:00.4 Ethernet controller: Solarflare Communications Device 1903 (rev 01)
0000:03:00.5 Ethernet controller: Solarflare Communications Device 1903 (rev 01)
```

6 Using the PCIe address, unbind the VFs to be passed through to the guest from the host sfc driver e.g.:

```
# echo 0000:03:00.5 > /sys/bus/pci/devices/0000\:03\:00.5/driver/unbind
```

7 Check that the required VF interface is no longer visible in the host using `ifconfig`.

8 On the host, stop the virtual machine:

```
# virsh shutdown <vmname>
```

9 On the host, edit the virtual machine XML file:

```
# virsh edit <vmname>
```

10 For each VF that is to be passed to the guest, add the following `<interface type>` section to the file identifying the VF PCIe address (use `lscpi` to identify PCIe address):

```
<interface type='hostdev' managed='yes'>
  <source>
   <address type='pci' domain='0x0000' bus='0x03' slot='0x00'   function='0x5'/>
  </source>
</interface>
```


11 Restart the virtual machine in the host and VF interfaces will be visible in the guest:

```
# virsh start <vmname>
```

The following (example) extract is from the VM XML file after a VF has been passed through to the guest using the procedure above (line numbers have been added for ease of description):

```
1. <interface type='hostdev' managed='yes'>
2.   <mac address='52:54:00:d1:ec:85'/>
     <source>
3.     <address type='pci' domain='0x0000' bus='0x03' slot='0x00' function='0x5'/>
     </source>
4.   <alias name='hostdev0'/>
5.   <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0'/>
 </interface>
```

XML Description

1 A description of how the VF interface is managed - added by user.

When `managed=yes`, the VF is detached from the host before being passed to the guest and the VF will be automatically reattached to the host after the guest exits.

If `managed=no`, the user must call `virNodeDeviceDetach` (or use the command `virsh nodedev-detach`) before starting the guest or hot-plugging the device and call `virNodeDeviceReAttach` (or use command `virsh nodedev-reattach`) after hot-unplug or after stopping the guest.

2 The VF MAC address. If not specified by the user this will be automatically assigned a random MAC address by libvirt The user can specify a MAC address when editing the XML file.

3 The VF PCIe address, this is the address of the VF interface as it is identified in the host. This should be entered by the user when editing the XML file.

4 If not specified by the user the alias name will be automatically assigned by libvirt The user can supply an alias when editing the XML file.

5 The VF PCIe address (as known by the guest) will be added automatically by libvirt.

For further information about the hostdev configuration and XML formats, refer to the following link:

http://libvirt.org/formatdomain.html#elementsNICSHostdev

8.4 PF-IOV

Physical Function I/O Virtualization allows PFs to be passed to a VM. Although this configuration is not widely used, it is included here for completeness. This mode provides no advantage over “Network Hostdev” and therefore Solarflare recommends that customers deploy “Network hostdev instead of PF-IOV. PF-IOV does not use SR-IOV and does not require SR-IOV hardware support.

Each physical port is partitioned into a number of PFs with each PF passed to a different Virtual Machine (VM). Each VM supports a TCP/IP stack and Solarflare adapter driver (sfc.ko).

This mode allows switching between PFs via the Layer 2 switch function configured in firmware.

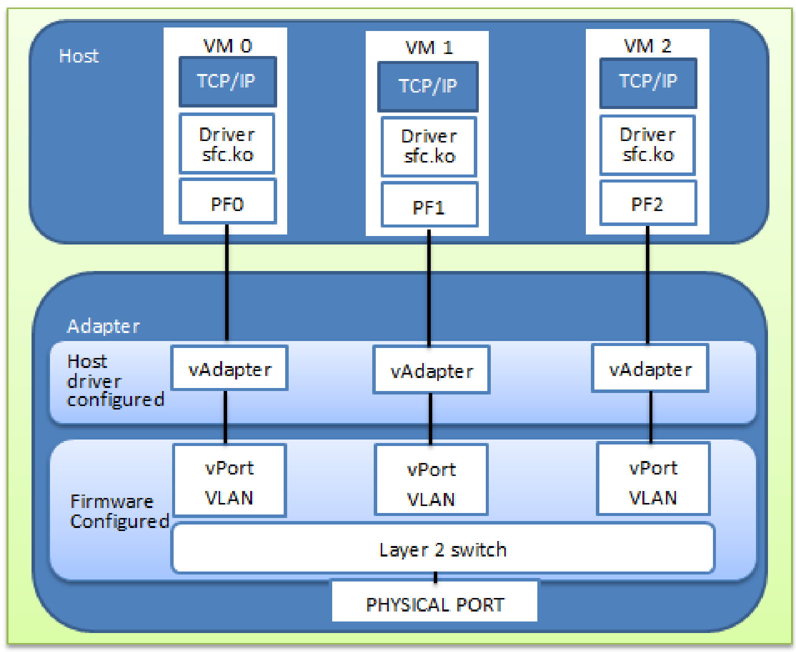


Figure 59: PFIOV

- Up to 16 PFs and 16 MAC addresses are supported *per adapter.*
- With no VLAN configuration, all PFs are in the same Ethernet layer 2 broadcast domain i.e. a packet broadcast from any one PF would be received by all other PFs.
- PF VLAN tags can optionally be assigned when creating PFs using the sfboot utility.
- The layer 2 switch supports replication of received/transmitted broadcast packets to all PFs and to the external network.
- The layer 2 switch supports replication of received/transmitted multicast packets to all subscribers.
- VFs are not supported in this mode.

PF-IOV Configuration

The `sfboot` utility from the Solarflare Linux Utilities package (SF-107601-LS) is used to partition physical interfaces to the required number of PFs.

- Up to 16 PFs and 16 MAC addresses are supported per adapter.
- The PF setting applies to all physical ports. Ports cannot be configured individually.
- vf-count must be zero.

1 To partition all ports (example configures 4 PFs per port):

```
# sfboot switch-mode=pfiov pf-count=4

Solarflare boot configuration utility [v4.3.1]
Copyright Solarflare Communications 2006-2014, Level 5 Networks 2002-
2005

eth5:
  Boot image                            Option ROM only
    Link speed                          Negotiated automatically
    Link-up delay time                  5 seconds
    Banner delay time                   2 seconds
    Boot skip delay time                5 seconds
    Boot type                           Disabled
  Physical Functions per port           4
  MSI-X interrupt limit                 32
  Number of Virtual Functions           0
  VF MSI-X interrupt limit              8
  Firmware variant                      full feature / virtualization
  Insecure filters                      Disabled
  VLAN tags                             None
  Switch mode                           PFIOV
```

2 A reboot of the server is required for the changes to be effective.

3 Following reboot the PFs will be visible using the `ifconfig` or `ip` commands - each PF will have a unique MAC address. The `lspci` command will also identify the PFs:

```
# lspci -d 1924:

07:00.0 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.1 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.2 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.3 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.4 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.5 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.6 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
07:00.7 Ethernet controller: Solarflare Communications Device 0903 (rev 01)
```


8.5 General Configuration

Enabling Physical Functions

Use the `sfboot` utility from the Solarflare Linux Utilities package to create PFs. Up to 16 PF and 16 MAC addresses are supported *per adapter.*

```
sfboot pf-count=<N>
```

PF VLAN tags can also be assigned using `sfboot`.

```
sfboot pf-count=4, pf-vlan=100,110,200,210
```

The first VLAN tag is assigned to the first function, thereafter the tags are applied to PFs in MAC address order.

Enabling Virtual Functions

On RHEL6.5 and later versions, VF creation is controlled through `sysfs`. Use the following commands (example) to create and view created VFs.

```
echo 2 > /sys/class/net/eth8/device/sriov_numvfs
cat /sys/class/net/eth8/device/sriov_totalvfs
```

On kernels not having this control via `sysfs` the Solarflare net driver module option `max_vfs` can be used to enable VFs. The `max_vfs` value applies to all adapters and can be set to a single integer i.e. all adapter physical functions will have the same number of VFs, or can be set to a comma separated list to have different numbers of VFs per PF.

The driver module parameter should be enabled in a user-created file (e.g. `sfc.conf`) in the `/etc/modprobe.d` directory and the sfc driver must be reloaded following changes.

```
options sfc max_vfs=4
options sfc max_vfs=2,4,8
```

When specified as a comma separated list, the first VF count is assigned to the PF with the lowest index i.e. the lowest MAC address, then the PF with the next highest MAC address etc. If the sfc driver option is used to create VFs, reload the driver:

```
modprobe -r sfc
modprobe sfc
```

VLAN tags can be dynamically assigned to VFs using `libvirt` commands, or using the ip command:

```
ip link vf NUM [mac LLADDR] [vlan VLANID]
```

To ensure VLAN tags persist after reboot, these can be configured in the VM XML file.

Using OpenOnload in a Virtual Machine

Onload users should refer to the Onload User Guide (SF-104474-CD) for further information about using Onload in a KVM.

When Onload and the sfc net driver have been installed in the guest, the sfc driver module option `num_vis` is used to allocate the required number of virtual interfaces. One VI is needed for each Onload stack using a VF.

Driver module options should be enabled in a user created file (e.g. sfc.conf) in the `/etc/modprobe.d` directory.

options sfc num_vis=<num>

Reload the driver after setting/changing this value:

```
# onload_tool reload
```

8.6 Feature Summary

Table 104: Feature Summary

	Default	**SRIOV**	**Partitioning**	**Partitioning + SRIOV**	**PFIOV**
Number of PFs (per adapter)	num ports	num ports	≥num ports ≤16	≥num ports ≤16	≥num ports ≤16
All PFs (per port) must be on unique VLANs	N/A	N/A	Yes	Yes	No
Num VFs (per adapter)	0	>0, ≤240	0	>0, ≤240	0
Mode suitable for PF PCIe passthrough	No	No	No	No	Yes
Mode suitable for VF PCIe passthrough	No	Yes	No	Yes	No
sfboot settings	switch-mode =default	switch-mode =sriov	switch-mode =partitioning	switch-mode =partitioning -with-sriov	switch-mode =pfiov
	pf-count=1	pf-count=1	pf-count>1	pf-count>1	pf-count>1
	vf-count=0	vf-count>0	vf-count=0	vf-count>0	vf-count=0
L2 switching between PF and associated VFs	N/A	Yes	N/A	Yes	N/A
L2 switching between PFs on the same physical port	N/A	N/A	No	No	Yes

8.7 Limitations

Users are advised to refer to the Solarflare net driver release notes for details of all limitations.

Per Port Configuration

For initial releases, all PFs on a physical port have the same expansion ROM configuration where PXE/iSCSI settings are stored. This means that all PFs will PXE boot or none will attempt to PXE boot. Users should ensure that a DHCP server responds to the first MAC address.

The PF (pf-count) configuration is a global setting and applies to all physical ports on an adapter. It is not currently possible to configure ports individually.

PTP

PTP can only run on the primary physical function of each physical port and is not supported on VF interfaces.

9 SR-IOV Virtualization Using ESXi

This chapter includes procedures for installation and configuration of Solarflare adapters for SR-IOV and Virtualization deployment using VMware® ESXi. For details of installation and configuration on VMware® platforms refer to Solarflare Adapters on VMware on page 264.

9.1 Introduction

This chapter describes the Solarflare SFN7000 series adapter support for SR-IOV and DirectPath I/O using the VMware ESXi hypervisor.

SR-IOV enabled on Solarflare adapters provides accelerated cut-through performance and is compatible with hypervisor based services and management tools. The advanced design of the Solarflare SFN7000 series adapter incorporates a number of specific features when deploying the adapter into virtualized environments.

- PCIe Physical Functions (PF)

  By partitioning the NIC, each physical network port can be exposed to the host as up to 16 PCIe Physical Functions (PF) with each having a unique interface name and unique MAC address.

- PCIe Virtual Functions (VF)

  A PCIe physical function, PF, can support a configurable number of PCIe virtual functions. In total 240 VFs can be allocated between the PFs. The adapter can also support a total of 2048 MSI-X interrupts.

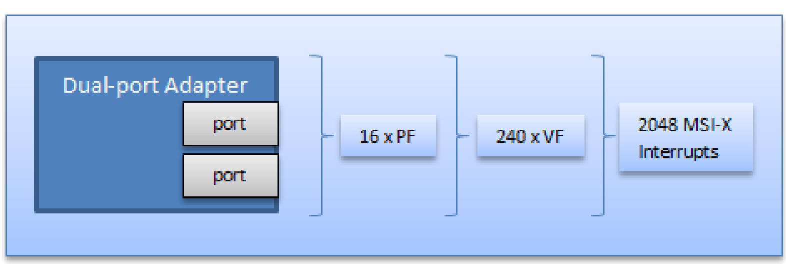


Figure 60: Per Adapter - Configuration Options

Features Supported

On ESXi Solarflare adapters support the following deployments:

Table 105: ESXi Virtualization Features

Feature	Guest OS
VF Passthrough	Linux 6.5 to 7.x
PF Passthrough (DirectPath I/O)	Linux 6.5 to 7.x Windows Server 2012 R2

Recommended Reading

The instructions in this chapter follow the procedures for VF and PF passthrough as documented in the VMware Networking user Guide for ESXi 5.5.

Platform Compatibility

SR-IOV and DirectPath I/O are not supported on all server platforms and users are advised to check server compatibility.

DirectPath I/O - PF Passthrough does not require platform SR-IOV support.

- Check for SR-IOV support in the VMware compatibility webpage:

  http://www.vmware.com/resources/compatibility/search.php

- Ensure the BIOS has all SR-IOV/Virtualization options enabled.
- On a server with SR-IOV correctly configured, identify if Virtual Functions (VF) can be exposed to the host OS. Refer to sfboot options below for the procedure to configure VFs on the Solarflare adapter.

BIOS

To use SR-IOV modes, SR-IOV must be enabled in the platform BIOS where the actual BIOS setting can differ between machines, but may be identified as SR-IOV, IOMMU or VT-d and VT-x on an Intel platform.

There may be other BIOS options which should be enabled to support SR-IOV, for example on DELL servers the following BIOS option must also be enabled:

`Integrated Devices, SR-IOV Global Enable`

Users are advised to consult the server vendor BIOS options documentation.

Supported Platform OS

Host

- VMware ESXi 5.5 and 6.0
- Solarflare v4.7 (or later) net drivers

Guest VM

- Red Hat Enterprise Linux 6.5 to 7.x
- Windows Server 2012 R2
- Solarflare v4.5 (or later) net drivers

Acceleration of Virtual Machines (VM) running guest operating systems not listed above are not currently supported.

Supported Adapters

All Solarflare SFN7000 series adapters fully support SR-IOV. Features described in this chapter are not supported by Solarflare SFN5000 or SFN6000 series adapters.

Solarflare Driver/Firmware

Features described in the chapter require the following (minimum) Solarflare driver and firmware versions.

```
# ethtool -i vmnic<N>
driver: sfc
version: 4.7.0.1031
firmware-version: 4.7.0.1020 rx0 tx0
```

The adapter must be using the *full-feature* firmware variant which can be selected using the `sfboot` utility and confirmed with **`rx0 tx0`** appearing after the version number in the output from ethtool as shown above.

The firmware update utility (`sfupdate`) and bootROM configuration tool (`sfboot`) are available in the Solarflare Linux Utilities package (SF-107601-LS issue 36 or later).

9.2 Configuration Procedure - SR-IOV

Use the following procedure to configure the adapter and server for SR-IOV.

- Install the Solarflare Driver on the ESXi host on page 405
- Install Solarflare Utilities on the ESXi host on page 406
- Install Solarflare Drivers in the Guest on page 405
- Configure VFs on the Host/Adapter on page 408
- Virtual Machine on page 409
- vSwitch and Port Group Configuration on page 411
- VF Passthrough on page 415

9.3 Configuration Procedure - DirectPath I/O

Use the following procedure to configure the adapter and server for PF passthrough.

- Install the Solarflare Driver on the ESXi host on page 405
- Install Solarflare Utilities on the ESXi host on page 406
- Install Solarflare Drivers in the Guest on page 405
- Partition the Adapter on page 421
- Virtual Machine on page 409
- Make PF Passthrough Devices available to the Guest on page 422
- Assign PF Passthrough Devices to the VM on page 423
- Listing Devices in a Windows Guest on page 424

9.4 Install Solarflare Drivers in the Guest

For both VF and PF passthrough configurations, the Solarflare adapter driver must be installed in the virtual machine guest OS.

Drivers are available from the Solarflare download site for Linux and Windows guests: https://support.solarflare.com/.

Driver installation procedures on a Guest are the same as installation for a host.

9.5 Install the Solarflare Driver on the ESXi host

Solarflare VMware ESXi drivers are available from: https://support.solarflare.com/.

Refer to Installing Solarflare Drivers and Utilities on VMware on page 266 for instructions to install VIB driver packages through the CLI.

9.6 Install Solarflare Utilities on the ESXi host

Solarflare utilities - including sfboot, sfupdate and sfkey are distributed in the Solarflare Linux Utilities package (SF-107601-LS issue 36 or later) from:

https://support.solarflare.com/.

Refer to Solarflare Utilities Package on page 270 for instructions to install the utilities on the ESXi host server.

NOTE: The Solarflare driver must be installed before using sfboot or any of the utilities.

sfboot - Configuration Options

The `sfboot` utility allows the user to configure:

- The number of PFs exposed per port to host and/or Virtual Machine (VM).
- The number VFs exposed per port to host and/or Virtual Machine (VM).
- The number of MSI-X interrupts assigned to each PF or VF.
- Firmware Variant and switch mode.

To check the current adapter configuration run the `sfboot` command:

```
# sfboot

Solarflare boot configuration utility [v4.7.0]
Copyright Solarflare Communications 2006-2015, Level 5 Networks 2002-2005

vmnic6:
  Boot image                          Disabled
  Physical Functions on this port     1
  PF MSI-X interrupt limit            32
  Virtual Functions on each PF        4
  VF MSI-X interrupt limit            16
  Port mode                           2x10G
  Firmware variant                    Full feature / virtualization
  Insecure filters                    Enabled
  MAC spoofing                        Disabled
  VLAN tags                           None
  Switch mode                         SR-IOV
  RX descriptor cache size            32
  TX descriptor cache size            16
  Total number of VIs                 2048
  Rate limits                         None
  Event merge timeout                 8740 nanoseconds
```

An alternative bootable ISO image of the Solarflare Utilities is available from the Solarflare download site under **Downloads > Linux > Misc**.

Firmware Variant

The firmware variant must be set to full-feature / virtualization.

```
# sfboot --adapter=vmnic6 firmware-variant=full-feature
```

SR-IOV (VF Passthrough) sfboot Settings

The following example creates 4 VFs for each physical port.

```
# sfboot switch-mode=sriov pf-count=1 vf-count=4
```

When used without the --adapter option, the command applies to all adapters

DirectPath I/O (PF Passthrough) sfboot Settings

The following example partitions the NIC so that each physical port is exposed as 4 PCIe PFs.

```
# sfboot switch-mode=partitioning pf-count=4 vf-count=0
```

For some configuration option changes using sfboot, the server must be power cycled (power off/power on) before the changes are effective. sfboot will display a warning when this is required.

9.7 Configure VFs on the Host/Adapter

The following host procedure is used to expose VFs from the Solarflare adapter.

1 Set the sfc driver module parameter for the required number of VFs:

```
esxcli system module parameters set -m sfc -p max_vfs=4
esxcli system module parameters list -m sfc
```

2 Use sfboot to create VFs on the adapter:

```
sfboot switch-mode=sriov vf-count=4
```

The server must be restarted (power off/power on) for these changes to take effect.

3 Following restart - list VFs exposed in the host:

```
# lspci | grep Solarflare
0000:04:00.0 Network controller: Solarflare SFC9120 [vmnic6]
0000:04:00.1 Network controller: Solarflare SFC9120 [vmnic7]

0000:04:00.2 Network controller: Solarflare  [PF_0.4.0_VF_0]
0000:04:00.3 Network controller: Solarflare  [PF_0.4.0_VF_1]
0000:04:00.4 Network controller: Solarflare  [PF_0.4.0_VF_2]
0000:04:00.5 Network controller: Solarflare  [PF_0.4.0_VF_3]

0000:04:00.6 Network controller: Solarflare  [PF_0.4.1_VF_0]
0000:04:00.7 Network controller: Solarflare  [PF_0.4.1_VF_1]
0000:04:01.0 Network controller: Solarflare  [PF_0.4.1_VF_2]
0000:04:01.1 Network controller: Solarflare  [PF_0.4.1_VF_3]
```

The example above is a dual-port adapter. Each physical port is exposed as 1 PF and 4 VFs (PFs are shown in bold text).

9.8 Virtual Machine

The procedures in the Chapter assume the VM has already been created. Users should consult the VMware documentation to create the VM. The recommended method is to use the VMware vSphere Web Client:

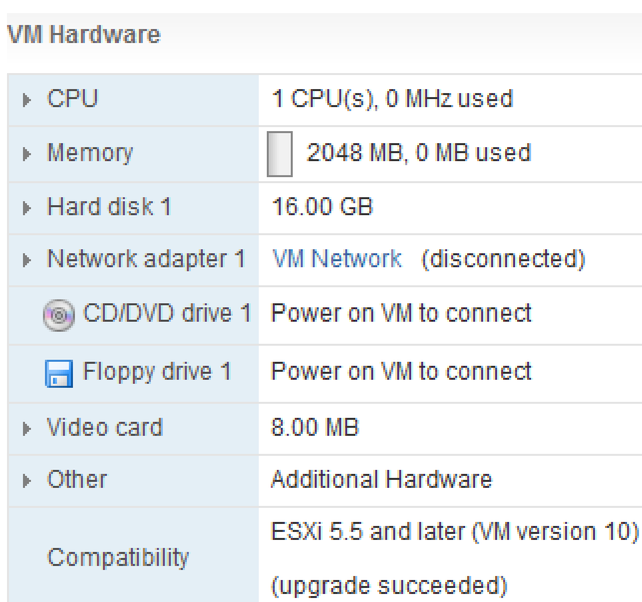
The VM must be compatible with version 10 (or later).

VM Compatibility

The VM must be compatible with ESXi 5.5 (or later). When the VM is not compatible, the following procedure via the vSphere Web Client will upgrade compatibility:

- Locate the VM from the listed hosts in the Web Client.
- Right click the VM > **Edit Settings**
- Under the **Virtual Hardware** tab > **Upgrade**
- Check the “Schedule VM Compatibility Upgrade” check box
- Select **ESXi 5.5 and later** from the drop down list
- Click **OK** to close the dialog
- Shutdown and restart the guest

After the VM has been shutdown and restarted, the compatibility will be displayed under the Settings tab:



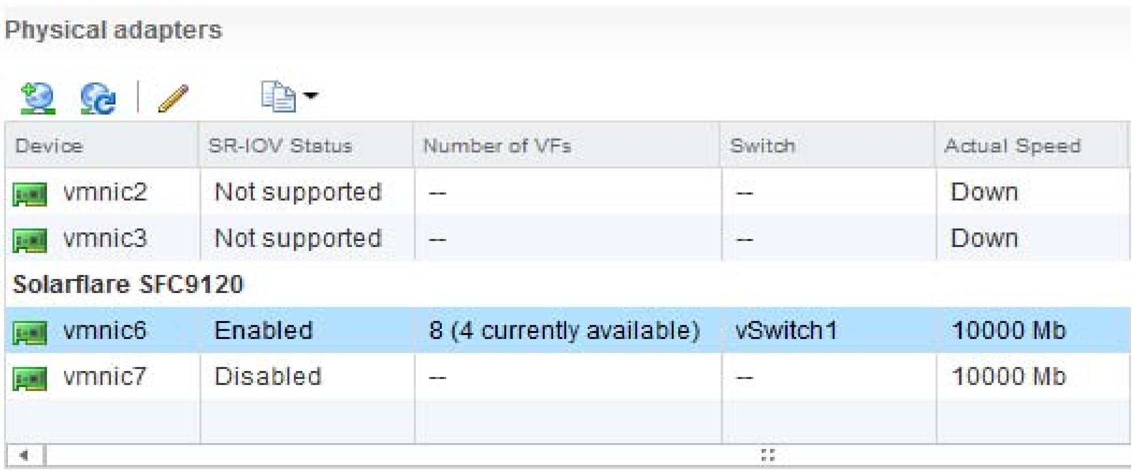
VM Hardware

CPU	1 CPU(s), 0 MHz used
Memory	2048 MB, 0 MB used
Hard disk 1	16.00 GB
Network adapter 1	VM Network (disconnected)
CD/DVD drive 1	Power on VM to connect
Floppy drive 1	Power on VM to connect
Video card	8.00 MB
Other	Additional Hardware
Compatibility	ESXi 5.5 and later (VM version 10) (upgrade succeeded)

9.9 List Adapters - Web Client

To list available adapters.

Navigate to the **Host** > **Networking** > **Physical Adapters**



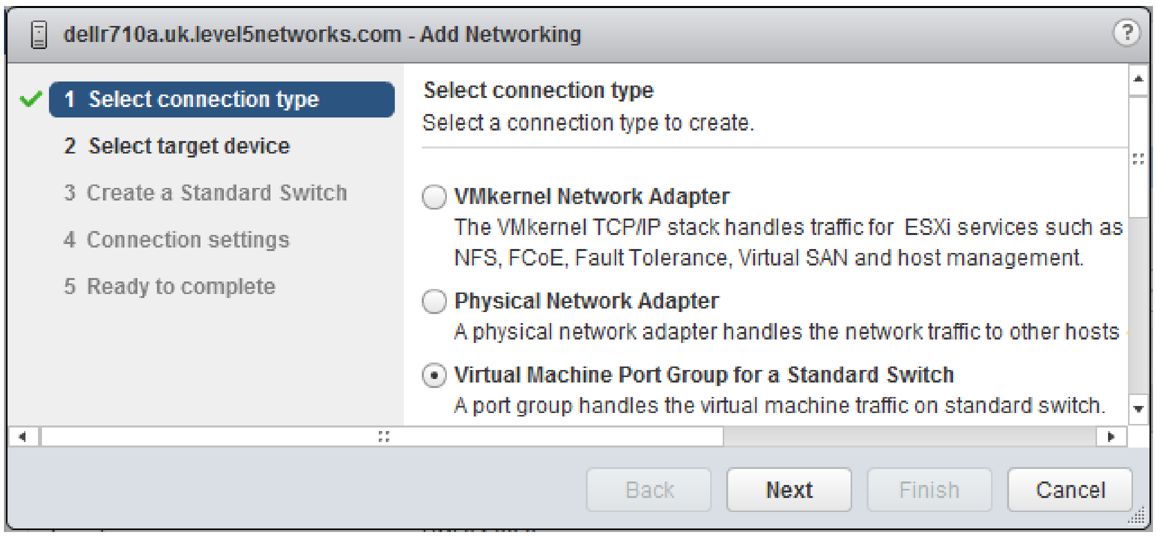
Selecting an adapter and clicking the edit (pencil) icon allows adapter settings to be edited.

9.10 vSwitch and Port Group Configuration

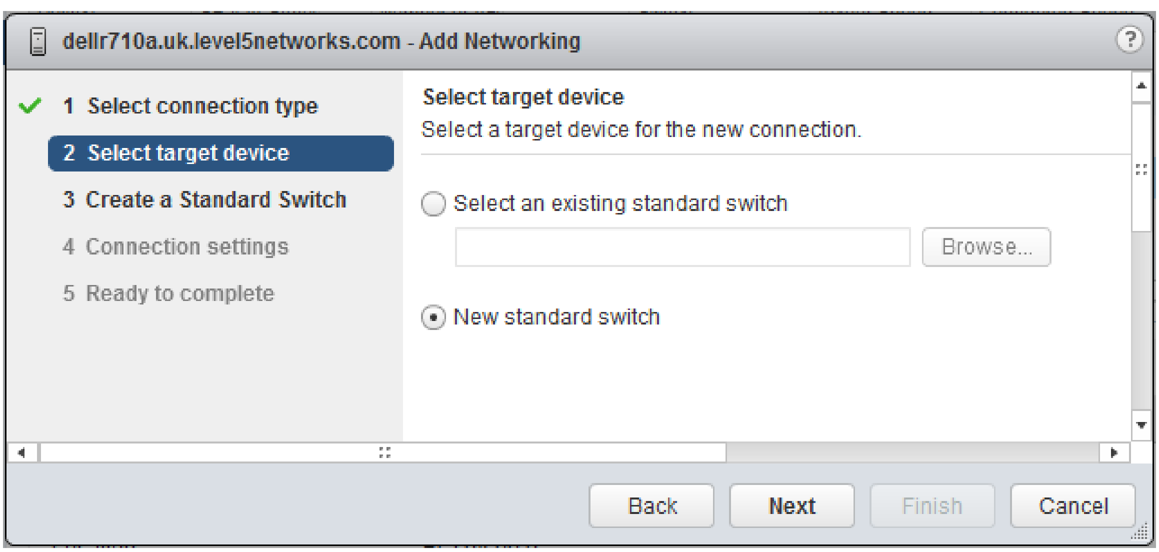
Using the vSphere Web Client, navigate to the host.

Right click and select **All vCenter Actions** > **Add Networking** to display the **Add Networking** wizard.

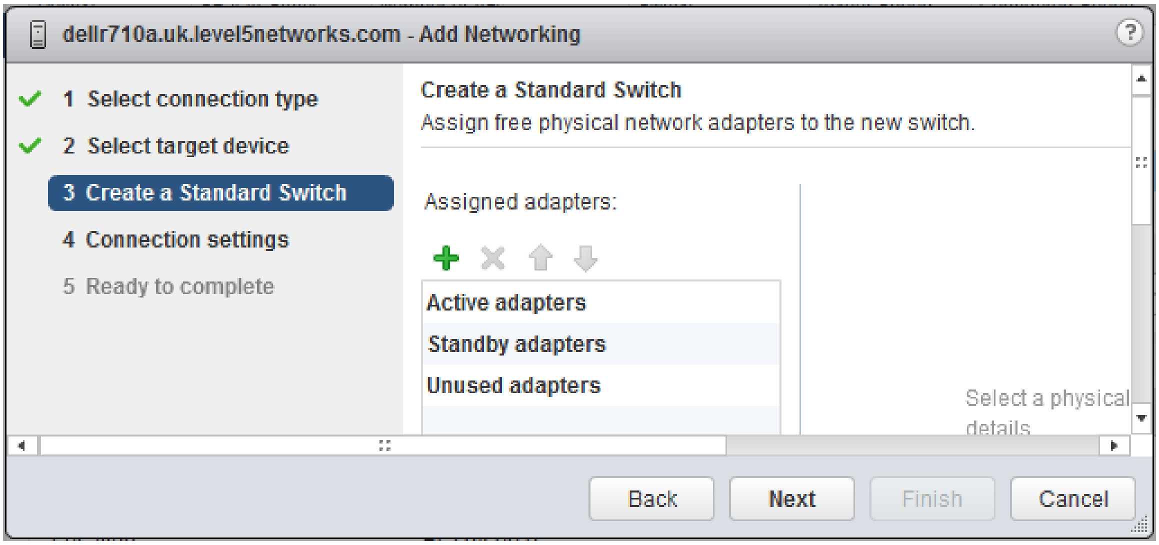
Select **Virtual Machine Port Group for a Standard Switch**, click **Next**.



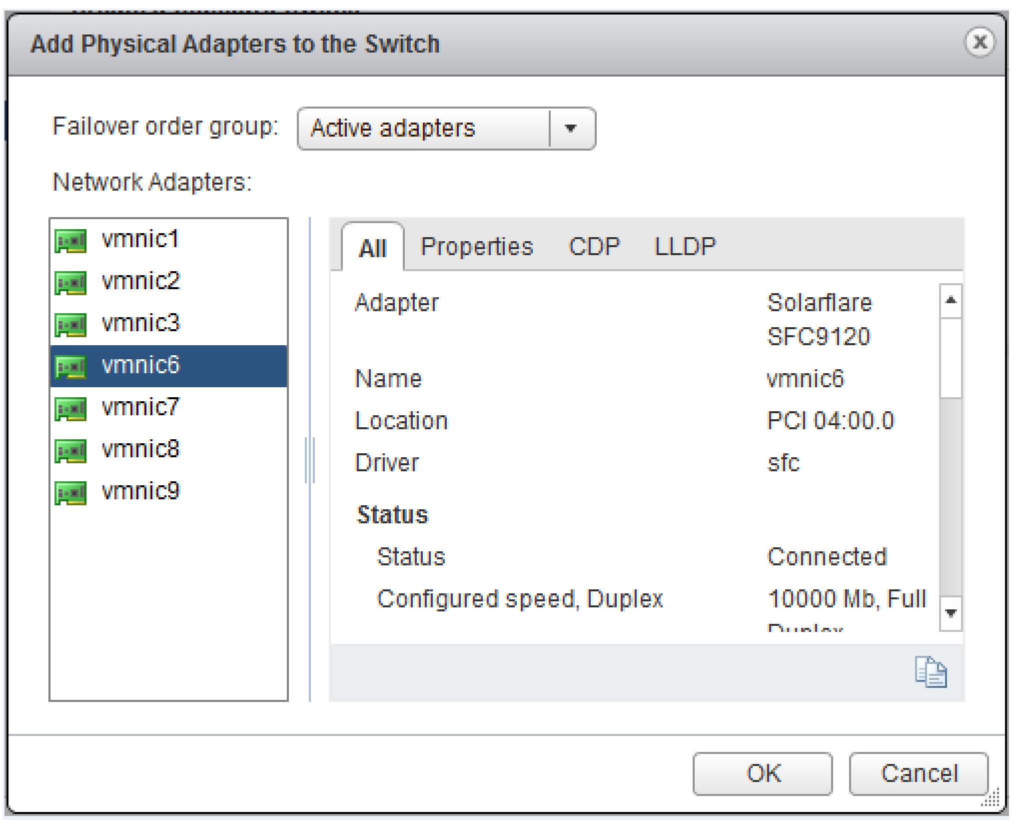
In **Select target device** - select an existing standard switch or create a new switch, click **Next**.



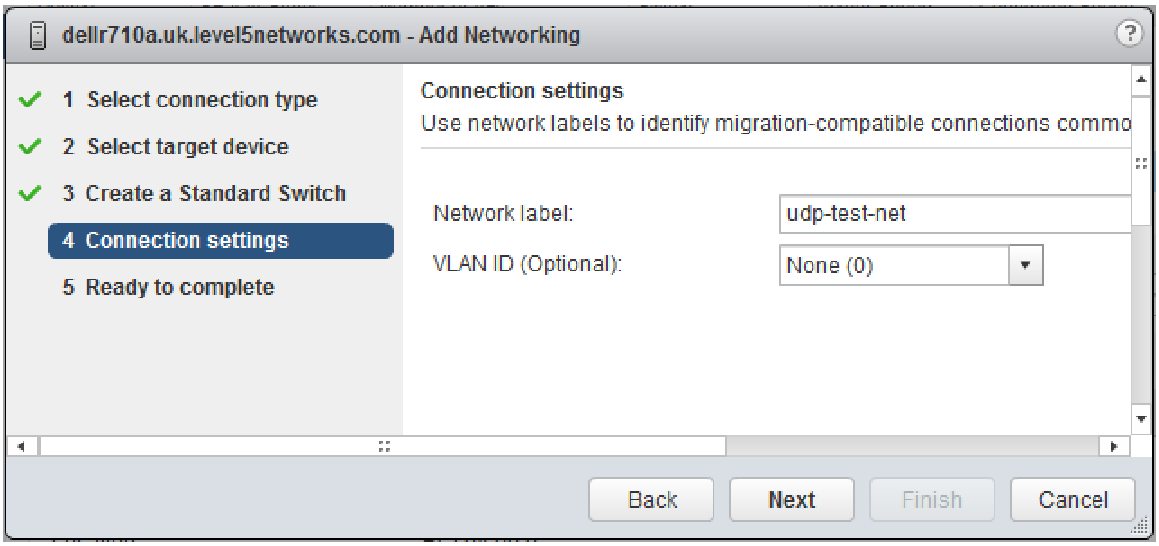
When creating a new standard switch, click the **plus** icon under Assigned adapters.

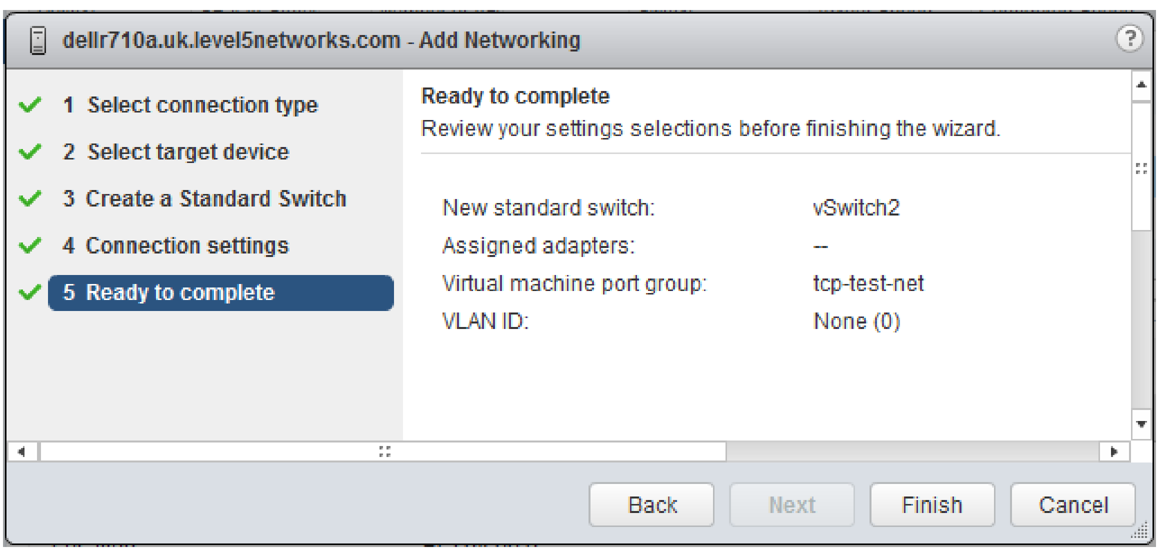


Select the required physical adapter(s) which, as uplinks, will connect the vswitch with a network.



Label the portgroup and assign a network label and VLAN ID if required. VFs will later be assigned to the same portgroup and will be able to send/receive traffic through the uplink adapter(s).

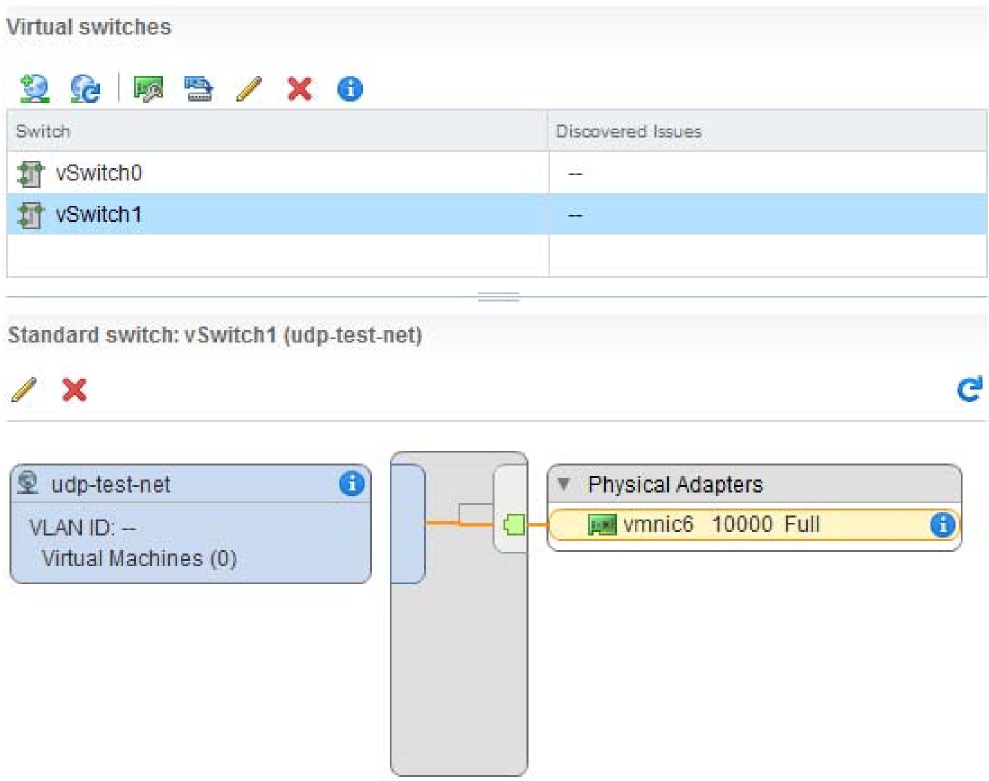




Review settings and click **Finish** to complete.

The vSwitch and associated uplink adapter(s) topology can be viewed as follows:

Select the host > **Manage** tab > **Networking** > **Virtual switches**



9.11 VF Passthrough

The following procedure uses the VMware vSphere Web Client to configure SR-IOV VF passthrough.

The procedure is documented in the VMware Networking user Guide for ESXi 5.5.

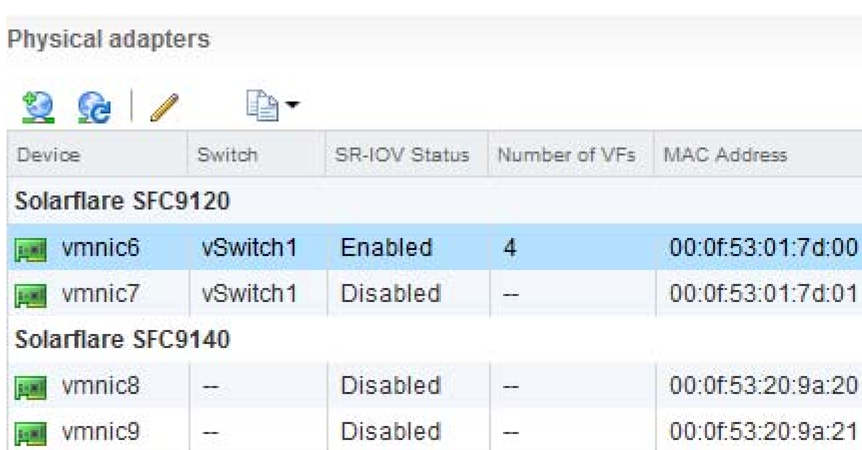
Assumptions

The procedure assumes the following tasks are complete:

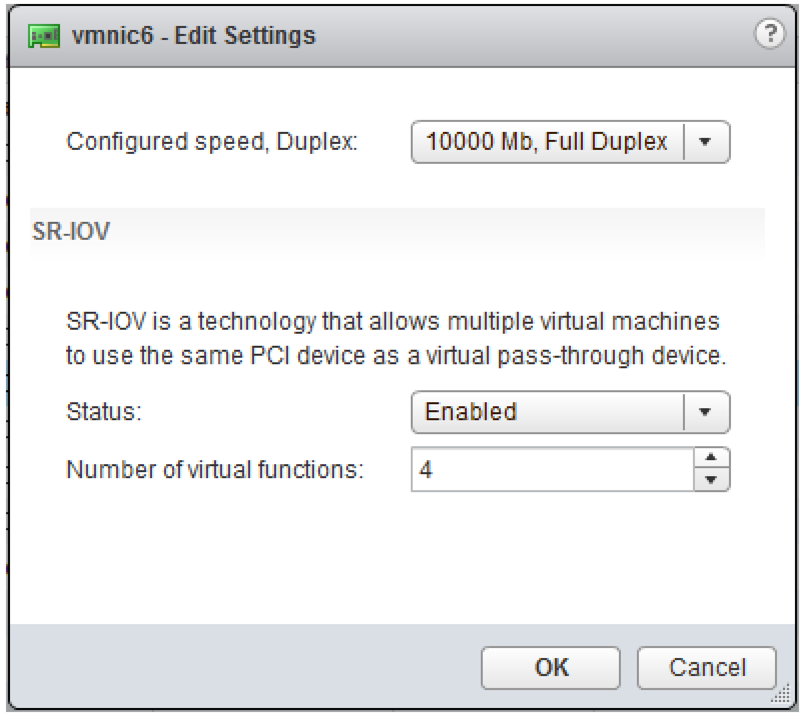
- A VM has been created and the guest OS installed.
- A Solarflare SFN7000 series adapter is physically installed on the host.
- The Solarflare VMware net driver package is installed on the host.
- A Solarflare net driver is installed in the VM.
- The Solarflare adapter exposes VFs to the host OS.
- On the host a vswitch has been created and the PF(s) from the Solarflare adapter are selected as uplinks.

Enable SR-IOV on the Host Adapter

- In the vSphere Web Client, navigate to the host.
- Select the **Manage** tab > **Networking** > **Physical Adapters** to list all available host adapters.
- Select the required Solarflare adapter, then select the pencil **(edit)** icon.



- From the adapter **Edit Settings** dialog enable SRIOV and specify the number of VFs which can be used by the VM.



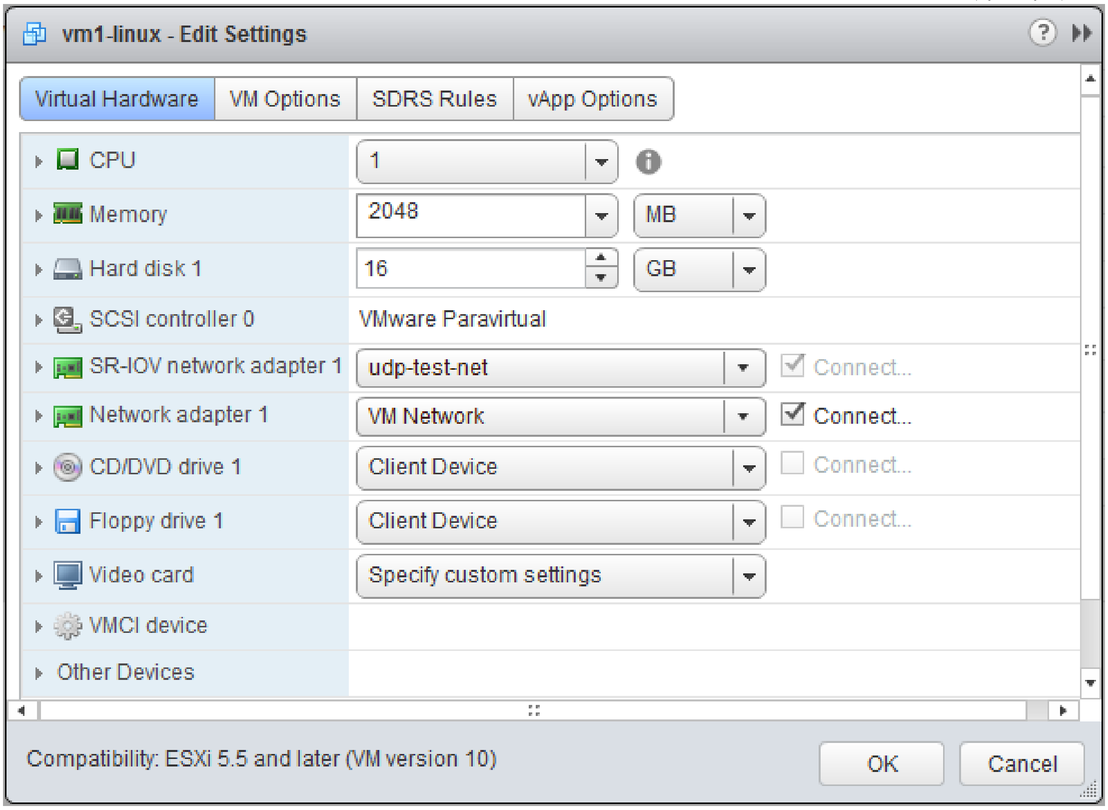
NOTE: The number of virtual functions should not exceed the value set by the max-vfs Solarflare driver option.

- **The host must be restarted following adapter settings changes**. Select and right clock the host in the Web Client window for reboot options.

Assign a VF as a SR-IOV Passthrough adapter to the VM

- In the vSphere Web Client, navigate to the VM.
- Power **OFF** the VM.
- Select the **Manage** tab > **Settings** > **VM Hardware**, then click the **Edit** button to display the VM - **Edit Settings** window.

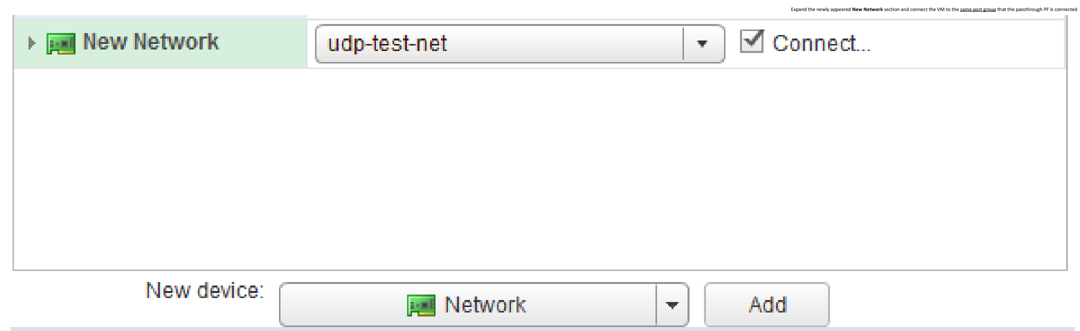
Alternatively navigate to the VM and right click to open the Edit window.



Select the **Manage** tab > **Setting** > **VM Hardware**, the click the **Edit** button to display the VM - **Edit Settings** window.

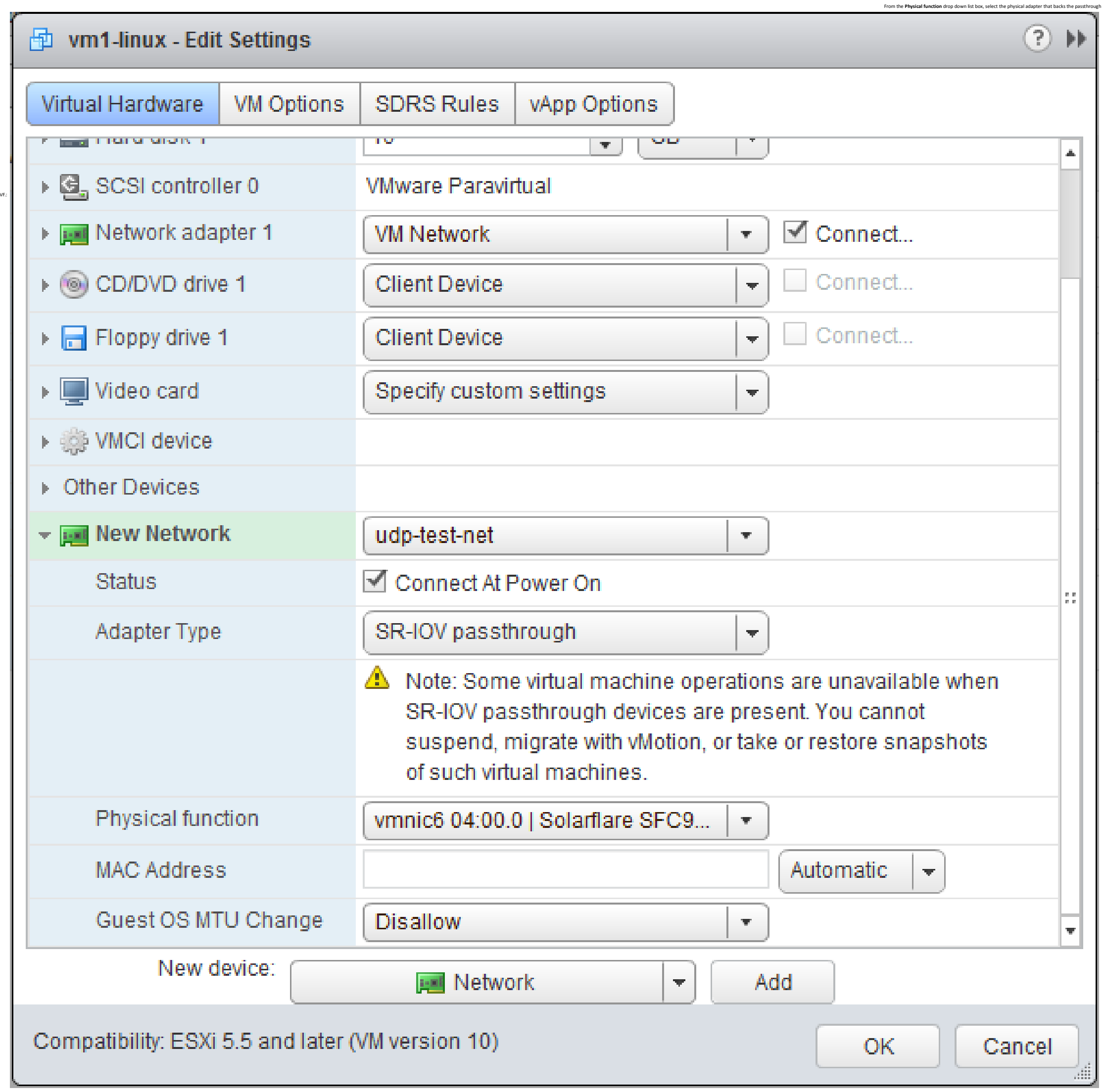
- To add a VF, select **Network** from the **New device** drop down list and click the **Add** button.

Expand the newly appeared **New Network** section and connect the VM to the same port group that the passthrough PF is connected



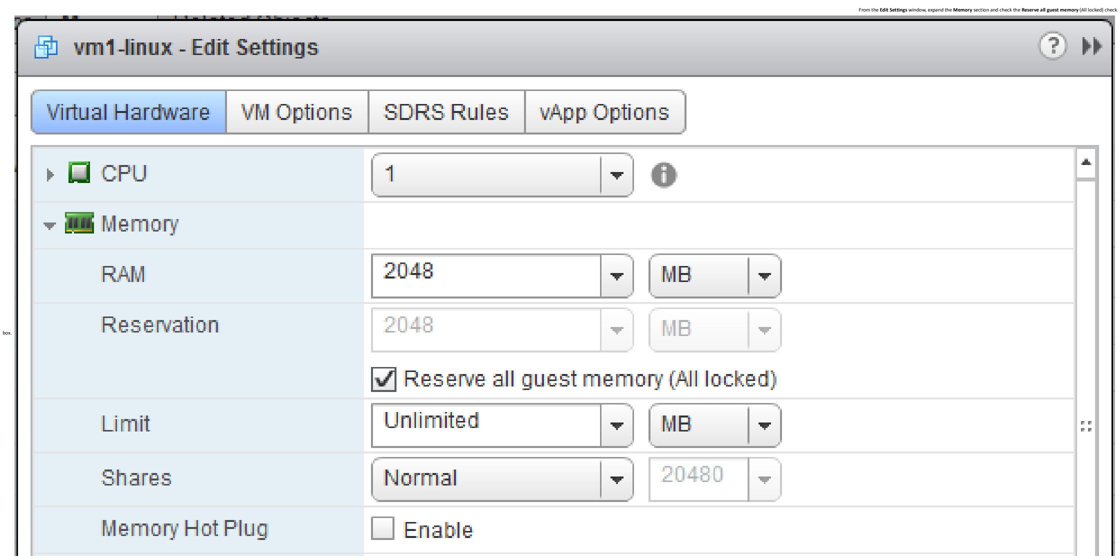
- Expand the **New Network** section:
- connect the VM to a port group identifying the network the VF is to connect to. If there is no uplink associated with the portgroup, VFs attached to the same portgroup can sent between themselves, but not to any network.

- select Adapter Type as **SR-IOV passthrough**.
- select the physical adapter associated with the port group and which will back the passthrough virtual machine adapter. ESXi will select an unused VF associated with this adapter.



From the Edit Settings window, expand the **Memory** section and check the R**eserve all guest memory** (All locked) check box.

Reserving all memory is required for both SR-IOV and DirectPath configurations because the adapter PF/VF must be able to DMA to the guest memory which needs to be present in physical RAM.



- From the **Edit Settings** window, click the **OK** button to close.

- With the VM selected, click the **Manage** tab and the **VM Hardware** option to view hardware configuration. The Solarflare adapter PF and VF are listed in this window.

VM Hardware

CPU	1 CPU(s), 0 MHz used
Memory	2048 MB, 0 MB used
Hard disk 1	16.00 GB
SR-IOV network adapter 1	
MAC Address	00:50:56:8b:5c:c0
DirectPath I/O	Not supported
Network	VM Network 2 (connected)
Allow Guest MTU Change	Disallow
Physical Function	vmnic6 04:00.0 | Solarflare SFC9120
Virtual Function	04:00.2 | Solarflare <class> Ethernet controller
Network adapter 1	VM Network (connected)
CD/DVD drive 1	Disconnected
Floppy drive 1	Disconnected
Video card	8.00 MB
Other	Additional Hardware
Compatibility	ESXi 5.5 and later (VM version 10) (upgrade succeeded)

Listing Passthrough Devices in a Linux Guest

When the VM has been restarted the passed through VF devices are visible in the Linux guest using both lspci and the ifconfig commands.

```
localhost ~]# lspci | grep Solarflare
0 Ethernet controller: Solarflare Communications Device
0 Ethernet controller: Solarflare Communications Device
localhost ~]# _
```

9.12 DirectPath I/O

DirectPath I/O allows a VM access to PF on platforms having an IOMMU. Platform support for SR-IOV is not required.

The Solarflare SFN7000 series adapter can be partitioned into multiple PCIe PFs, supporting up to 16 PCIe physical functions.

For details of NIC Partitioning see NIC Partitioning on page 58.

Partition the Adapter

The Solarflare NIC can be partitioned to expose up to 16 PFs using the sfboot command from the ESXi host command line interface:

```
# sfboot --adapter=vmnic6 vf-count=0 pf-count=4 switch-mode=partitioning
```

The server must be cold-power cycled. When the server restarts, PFs will be visible in the host:

```
# lspci -vvv | grep Solarflare

0000:07:00.0 Ethernet controller Network controller: Solarflare SFC9140
[vmnic8]
0000:07:00.1 Ethernet controller Network controller: Solarflare SFC9140
[vmnic9]
0000:07:00.2 Ethernet controller Network controller: Solarflare SFC9140
[vmnic4]
0000:07:00.3 Ethernet controller Network controller: Solarflare SFC9140
[vmnic5]
0000:07:00.4 Ethernet controller Network controller: Solarflare SFC9140
[vmnic10]
0000:07:00.5 Ethernet controller Network controller: Solarflare SFC9140
[vmnic11]
0000:07:00.6 Ethernet controller Network controller: Solarflare SFC9140
[vmnic12]
0000:07:00.7 Ethernet controller Network controller: Solarflare SFC9140
[vmnic13]
```

In the above example a dual-port adapter is partitioned to expose 4 PFs per physical port.

Make PF Passthrough Devices available to the Guest

This procedure uses the vSphere Web Client and follows the procedure from the vmware documentation for DirectPath I/O:

VMware Networking user Guide for ESXi 5.5

- Navigate to the host. Select the **Manage** tab > **Settings** option.
- Under the **Hardware** section, select **PCI Devices**.

DirectPath I/O PCI Devices Available to VMs

ID	Status	Vendor Name
04:00.2	Available	Solarflare
00:09.0	Not Configurable	Intel Corporation
07:00.3	Available	Solarflare
07:00.7	Available	Solarflare
07:00.0	Unavailable	Solarflare
07:00.2	Available	Solarflare
07:00.5	Available	Solarflare
07:00.6	Available	Solarflare
07:00.1	Unavailable	Solarflare
07:00.4	Available	Solarflare

- Right click any device listed and select **Edit** from the pop-up menu. Or click the edit (pencil) icon.
- From the **All PCI Devices** window, tick the check-box of the required PF devices:

All PCI Devices

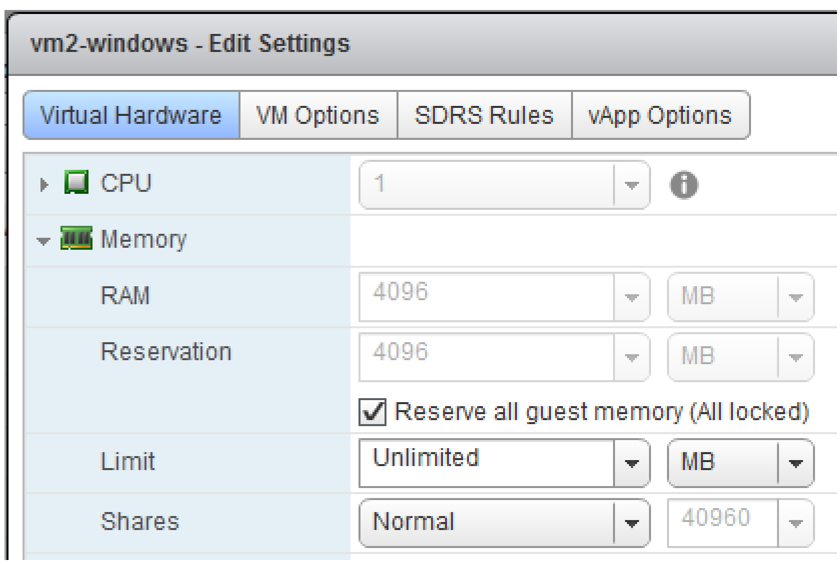
ID	Status	1 ▲ Vendor Name	Device Name
[x] 07:00.5	Available	Solarflare	SFC9140
[x] 07:00.7	Available	Solarflare	SFC9140
[x] 07:00.6	Available	Solarflare	SFC9140
[x] 07:00.2	Available	Solarflare	SFC9140
[x] 07:00.3	Available	Solarflare	SFC9140
[x] 07:00.4	Available	Solarflare	SFC9140
[ ] 07:00.0	Unavailable	Solarflare	SFC9140
[ ] 07:00.1	Unavailable	Solarflare	SFC9140

CAUTION: The Primary PF from each physical port cannot be passed through to a guest. PF0, having configuration privileges, is used by the sfc driver in the hypervisor and should not be passed to a VM.

Note When selecting new devices these will be marked as “Available (pending)”. The host must be restarted to effect the changes.

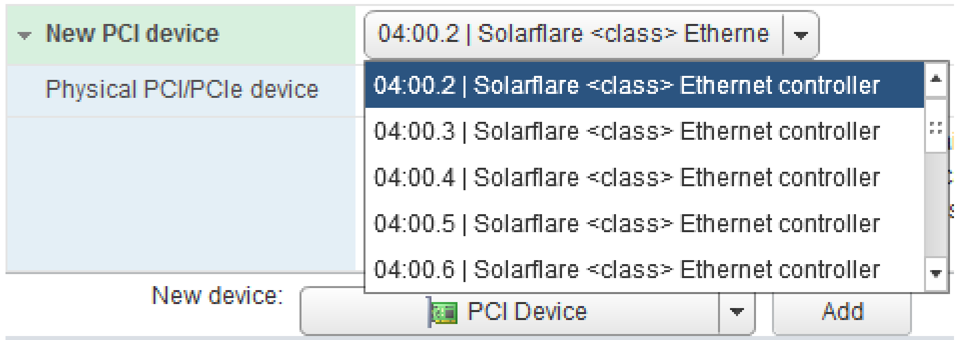
Assign PF Passthrough Devices to the VM

- Locate and select the VM in the vSphere Web Client.
- Power **OFF** the VM.
- Select the **Manage** tab, select **Settings** then **VM Hardware**, then click the **Edit** button to open the VM Edit Settings window.
- Select the **Virtual Hardware** tab and then the **Memory** option:



- For the **Memory** option, set the **Limit** to “Unlimited”.

- Next, select **PCI Device** from the **New device** drop-down list, then click the **Add** button.

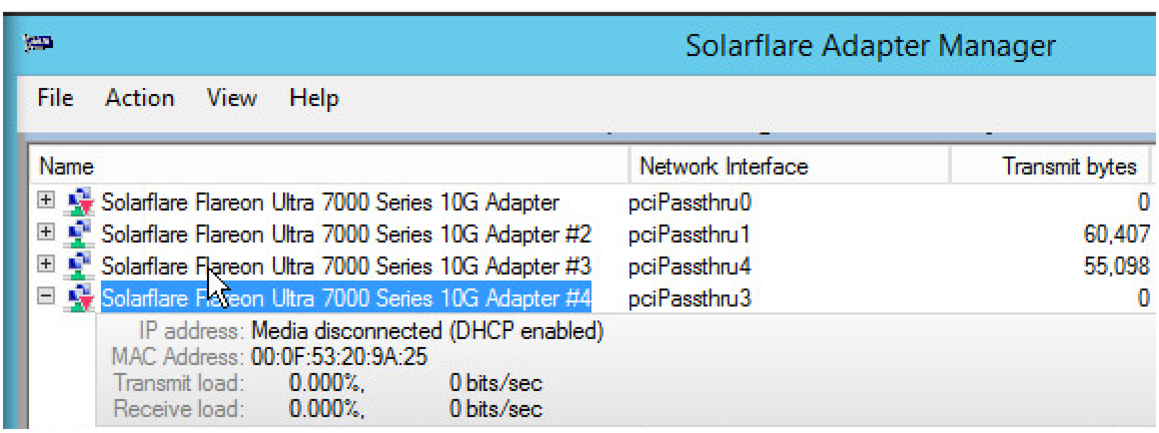


- Select the required Solarflare PF device to be passed through.
- Repeat - selecting **New Device** then **Add** for each PF to be passed through.
- Click the **OK** button when done.
- Power ON the VM.

Listing Devices in a Windows Guest

Start the Solarflare Adapter Manager GUI by selecting **Start** > select **arrow icon** > from the desktop and select the **Configure adapters on this PC**.

This will start **SAM**, after a refresh, Selected PFs will be visible as Solarflare adapters.



10 Solarflare Adapters on Mac OS X

This chapter covers the following topics on the Mac OS X® platform:

10.1 System Requirements

- Refer to Software Driver Support on page 13 for supported Mac OS X Distributions.
- Solarflare Mac OS X drivers are supported for all Solarflare SFN5xxx AND SFN6xxx series adapters.
- Driver package SF-107120-LS supports OS X 10.8 and earlier versions.
- Driver package SF-111621-LS supports OSX 10.9 and later versions.

10.2 Supported Hardware Platforms

The following Apple hardware platforms are supported:

- Mac Pro
- Mac Pro Server
- X-Serve (supported but not routinely tested by Solarflare)

10.3 Mac OS X Platform Feature Set

The following table lists the features supported by Solarflare adapters on Mac OS X distributions.

Table 106: Mac OS X Feature Set

Large Receive Offload	TCP receive frame coalescing to reduce CPU utilization and improve TCP throughput
TCP Segmentation Offload	TCP transmit segmentation to reduce CPU utilization and improve TCP throughput
RMON	Statistics counters
Checksum offloads	IPv4, TCP and UDP
MSI Interrupts	
MTU	Standard 1500 byte and jumbo 9000 byte MTU

10.4 Thunderbolt

The Solarflare adapter driver provides basic support for Thunderbolt. When a network adapter is connected to a Thunderbolt-capable system e.g. via a Thunderbolt-to-PCIe chassis, the interfaces can be configured in the usual way.

Due to limitations in the Thunderbolt connection performance may be worse compared to using the Solarflare adapter in a PCIe slot.

Full support for Thunderbolt, including plugging and unplugging the Thunderbolt cable is planned for a future release.

10.5 Driver Install

Uninstall Previous Driver

An installed Solarflare network adapter driver *must be uninstalled* before upgrading to a new driver release.

1 Open System Preferences > Network.

2 Disable the service for all ports of the driver:

- choose an active driver service in the list
- click on the gear icon and choose 'Make Service Inactive'

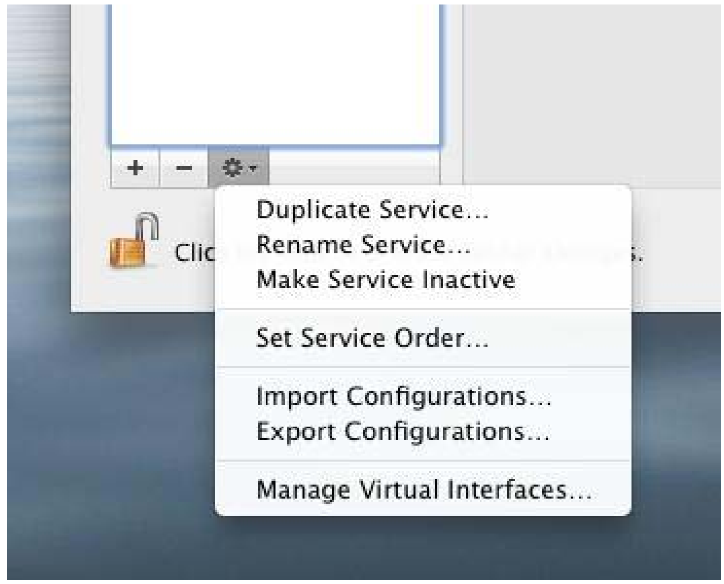


Figure 61: Disable Driver Services

3 Repeat above steps for all ports of the driver.

4 Double-click SF-107120-LS.dmg in Finder to mount the disk image. Invoke the Solarflare driver uninstall script in Terminal as root (replacing <version> with the version number of the install package that is being used)

```
/Volumes/Solarflare10GbE-<version>/uninstall.sh
```

Download and Install the Mac OS X Driver

1 Download SF-107120-LS.dmg into a convenient working directory.

2 Double click SF-107120-LS.dmg in Finder to mount the disk image.

3 Run the Solarflare10GbE.pkg install package and follow the install instructions.

Figure 62: Install Solarflare Driver Window

10.6 Interface Configuration

With the adapter driver installed, the network interface can be configured using the network interface settings menu:

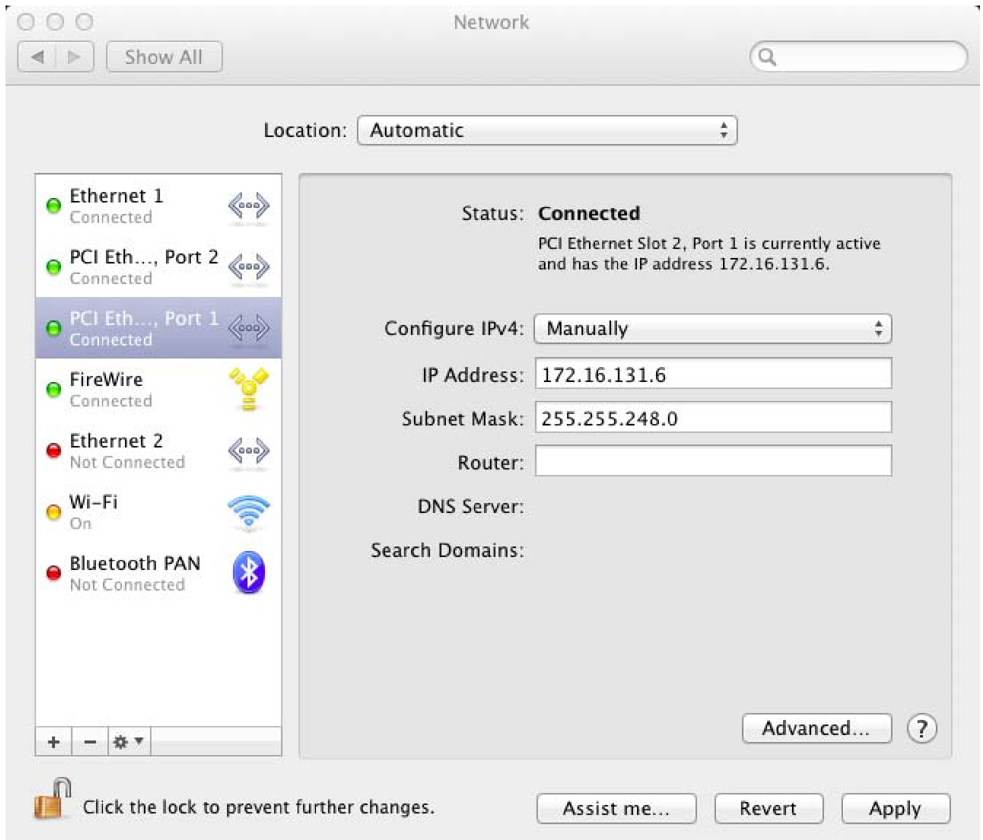


Figure 63: Solarflare Adapter Interface Configuration

10.7 Tuning

System Tuning

For many applications (including file serving) tuning the Mac OS X network stack for 10G operation can improve network performance. Therefore, for such applications it is possible to tune the Mac OS X kernel and network stack by applying the following settings in the `/etc/sysctl.conf` file. Settings added to `/etc/sysctl.conf` are effective following a machine reboot.

```
kern.ipc.maxsockbuf=4194304
net.inet.tcp.sendspace=2097152
net.inet.tcp.recvspace=2097152
net.inet.tcp.delayed_ack=2
```

Settings can also be updated using the following method - but these are non-persistent and will return to default values following a reboot:

```
sudo sysctl -w <name>=<value>
```

Optional Driver Tuning

The driver's default configuration has been chosen to provide optimal performance over a wide range of applications. It is recommended to only change the driver settings if advised to do so by Solarflare support.

10.8 Driver Properties via sysctl

Driver properties are also made visible via the `sysctl` program. Changes made via `sysctl` calls are applied immediately, and are not persistent (i.e. the changes are lost when the driver is unloaded or after a reboot). To make persistent changes to `sysctl` values, edit the file `/etc/sysctl.conf`.

Changes made via `sysctl` apply to a single driver interface, using the BSD name of the network interface. The BSD name of a network interface is shown by the ifconfig command line tool, and in the Network Utility application. For Ethernet interfaces, the BSD name starts with en followed by a number.

Table 107 identifies currently supported driver `sysctl` values.

Table 107: Mac OS X sysctl driver values

sysctl name	R/W	Value	Description
`net.sfxge.version`	RO		Driver version string
`net.sfxge.<enX>.mac`	RO		MAC address
`net.sfxge.<enX>.moderation`	RW	0	Disable interrupt moderation
		> 0	interrupt moderation (microseconds)
`net.sfxge.<enX>.rx_ring_size`	RW	512 1024 2048 4096	Hardware receive ring entries
`net.sfxge.<enX>.tx_ring_size`	RW	512 1024 2048 4096	Hardware transmit ring entries
`net.sfxge.<enX>.ipv4lro`	RW	0	IPv4 LRO disabled
		1	IPv4 LRO enabled

Table 107: Mac OS X sysctl driver values

sysctl name	R/W	Value	Description
`net.sfxge.<enX>.ipv6lro`	RW	0	IPv6 LRO disabled
		1	IPv6 LRO enabled
`net.sfxge.<enX>.ipv4tso`	RO	0	IPv4 TSO disabled
		1	IPv4 TSO enabled
`net.sfxge.<enX>.ipv6tso`	RO	0	IPv6 TSO disabled
		1	IPv6 TSO enabled

10.9 Firmware Update

The Solarflare driver package for Apple Mac OS X also includes the firmware update utility program sfupdate.

When the driver package is installed the sfupdate binary is installed into `/Library/Application Support/Solarflare10GbE` directory and a symbolic link placed in `/usr/local/bin/sfupdate`.

When upgrading or installing the network adapter driver it is recommended to upgrade the adapter firmware.

sfupdate: Command Usage

The general usage for sfupdate is as follows (as root):

```
sfupdate [--adapter=enX] [options]
```

where:

enX is the interface name of the Solarflare adapter to be upgraded.

`option` is one of the command options listed in Sfupdate Options on page 432.

The format for the options are <option>=<parameter>

Running the command `sfupdate` with no additional parameters will display the current firmware version for all Solarflare adapters and identifies whether the firmware within sfupdate is more up to date.

sfupdate: All Solarflare adapters

1 Run `sfupdate` to check that the firmware on all adapters is up to date.

2 Run `sfupdate --write` to update the firmware on all adapters.

sfupdate: Command Line Options

Table 108 lists the options for `sfupdate`.

Table 108: Sfupdate Options

Option	Description
`-h, --help`	Display help for the available options and command line syntax.
`-i, --adapter=enX`	Specifies the target adapter when more than one adapter is installed in the localhost. enX = Adapter ifname or MAC address (as obtained with `--list`).
`--list`	Shows the adapter ID, adapter name and MAC address of each adapter installed in the localhost.
`--write`	Re-writes the firmware from the images embedded in the `sfupdate` tool. To re-write using an external image, specify `--image=<filename>` in the command. `--write` fails if the embedded image is the same or a previous version. To force a write in this case, specify `--force` in the command.
`--force`	Force the update of all firmware, even if the installed firmware version is the same as, or more recent then, the firmware embedded in `sfupdate`.
`--image=(filename)`	Update the firmware using the binary image from the given file rather than from those embedded in the utility.
`-y, --yes`	Prompts for user confirmation before writing the firmware to the adapter.
`-v, --verbose`	Verbose mode.
`-s, --silent`	Suppress output while the utility is running; useful when the utility is used in a script.
`-V, --version`	Display version information and exit.

10.10 Performance

The following section is an overview of benchmark tests results measured by Solarflare to provide an indication of expected performance with current drivers.

Performance tests were conducted on Mac OS X 10.7.2 on a pair of Mac Pro servers configured back-to-back. The Mac OS X network stack was tuned for 10G operation as described in Tuning on page 429.

Reference System Specification

- MacPro5,1, 3GB memory (all channels populated)
- Processor: Single Quad-Core Intel Xeon @ 2.8 GHz L2 Cache (per core): 256 KB, L3 Cache: 8 MB

Throughput (Netperf TCP_STREAM)

Results using Netperf IPv4 TCP_STREAM at 1500 MTU:

Table 109: Throughput Results

Message size	No. of streams	Bandwidth
64Kbyte	1	9.26 Gb/s
64Kbyte	1 bidirectional	17.8 Gb/s

Latency (Netperf TCP_RR)

Latency measured using Netperf IPv4 TCP_RR will depend on the interrupt moderation settings and the type of SFN5xxx adapter used (10GBaseT cards have higher latency). Latency as measured on SFN5122F at the standard 1500 MTU is as follows:

- Interrupt moderation at 40µs: **45.7 µs** RTT/2
- Interrupt moderation disabled: **18.2 µs** RTT/2

File System Benchmarks (AJA System Test)

The AJA System Test benchmark provides some indication of likely network file system performance for video applications.

System Setup:

- 2.25GB ramdisk on file-system ‘target’ server. The test consisted of writing and then reading a 1.0GB file to and from this ramdisk
- SFN5122F SFP+ back-to-back configuration

- To configure a ramdisk for the test (of size 4500000 x 512k sectors):

```
$ sudo diskutil eraseVolume HFS+ "ramdisk" \
  `hdiutil attach -nomount ram://4500000`
```

Table 110: File System Benchmark Test Results

Protocol		1500 MTU		9000 MTU Jumbo	
AFP	Frame size	Read MB/s	Write MB/s	Read MB/s	Write MB/s
	720 X 468, 8 bit	439.5	547.6	433.0	574.4
	1920 x 1080, 10 bit	509.3	728.3	502.3	770.1
	4096 x 2160, 10 bit-RGB	521.3	807.0	516.0	849.5
SMB	1920 X 1080, 10 bit	312.7	255.7	370.0	291.3

11 Solarflare Boot ROM Agent

Solarflare adapters support PXE and iSCSI booting, enabling diskless systems to boot from a remote target operating system. Solarflare adapters comply with PXE 2.1.

Solarflare adapters are shipped with boot ROM support 'exposed', that is the Boot ROM Agent runs during the machine bootup stage allowing the user to enter the setup screens (via Ctrl+B) and enable PXE support when this is required. The Boot ROM Agent can also be invoked using the Solarflare supplied sfboot utility - For instructions on the sfboot method refer to the sfboot commands in the relevant OS section of this user guide. PXE boot is supported on all Solarflare adapters.

Using the sfboot utility the boot-image options identifies which boot images are exposed on the adapter during boot time. The 'all' or 'optionrom' options allows the user to select either PXE or iSCSI boot using the boot-type option. The 'uefi' option controls booting from UEFI software. *If network booting is not required, startup time can be decreased when the boot-image option is 'disabled' so that the CTRL-B option is not exposed during system startup.*

Some Solarflare distributors are able to ship Solarflare adapters with PXE boot enabled. Customers should contact their distributor for further information.

PXE and iSCSI network boot is not supported for Solarflare adapters on IBM System p servers.

- Configuring the Solarflare Boot ROM Agent on page 436
- PXE Support on page 436
- Multiple PF - PXE Boot on page 456
- iSCSI Boot on page 439
- Configuring the iSCSI Target on page 439
- Configuring the Boot ROM on page 439
- DHCP Server Setup on page 445
- Installing an Operating System to an iSCSI target on page 447
- Default Adapter Settings on page 455
- Multiple PF - PXE Boot on page 456

11.1 Configuring the Solarflare Boot ROM Agent

Updating Firmware

Before configuring the Boot ROM Agent, Solarflare recommend that servers are running the latest adapter firmware which can be updated as follows:

- From a Windows environment you can use the supplied Command Line Tool sfupdate.exe. See Sfupdate: Firmware Update Tool on page 197 for more details.
- From a Linux environment, you can update the firmware via sfupdate. See Upgrading Adapter Firmware with sfupdate on page 86.
- From a VMware environment, you can update the firmware via sfupdate. See Upgrading Adapter Firmware with sfupdate on page 283.

NOTE: The Solarflare firmware supports both PXE and iSCSI.

Configuring the Boot ROM Agent

The Boot ROM Agent can be configured in the following ways:

- On server startup, press Ctrl+B when prompted during the boot sequence.
- From a Windows Environment, via SAM. See Using SAM for Boot ROM Configuration on page 170. Alternatively you can use the supplied Command Line Tool sfboot. See Sfboot: Boot ROM Configuration Tool on page 183.
- From a Linux environment, via sfboot. See Configuring the Boot ROM with sfboot on page 72.
- From a VMware environment, via sfboot. See Solarflare Utilities Package on page 270.

11.2 PXE Support

Solarflare Boot ROM agent supports the PXE 2.1 specification. PXE requires DHCP and TFTP Servers, the configuration of these servers depends on the deployment service used.

Linux

For Red Hat Enterprise and SUSE Linux Enterprise Server, please consult your Linux documentation.

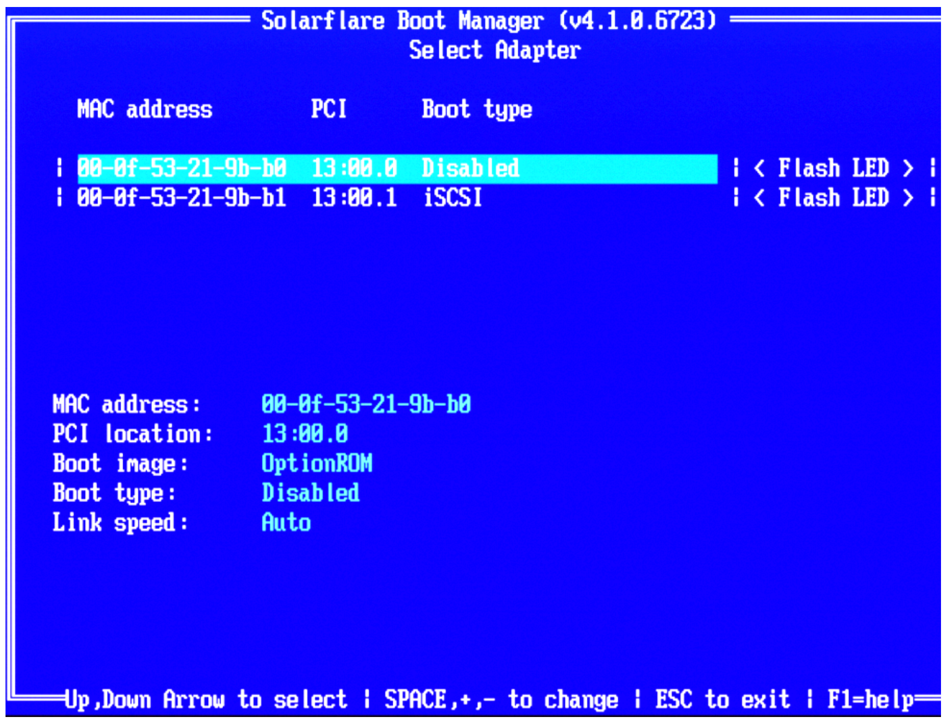
See Unattended Installation - Red Hat Enterprise Linux on page 52 and Unattended Installation - SUSE Linux Enterprise Server on page 54 for more details of unattended installation on Linux

Configuring the Boot ROM Agent for PXE

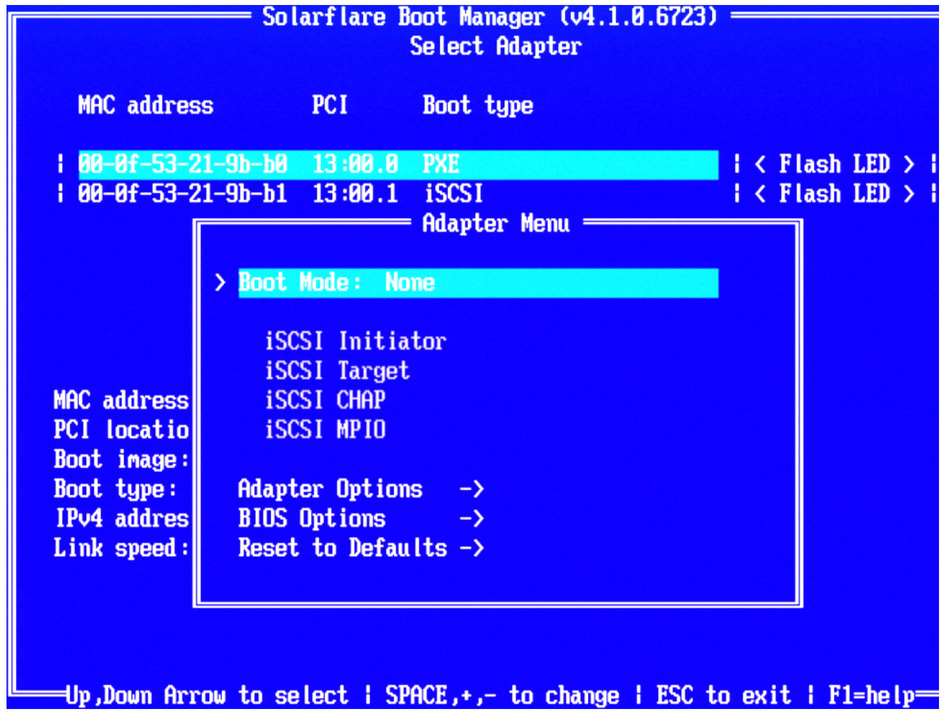
This section describes configuring the adapter via the Ctrl+B option during server startup. For alternative methods of configuring PXE see above.

NOTE: If the BIOS supports console redirection, and you enable it, then Solarflare recommends that you enable ANSI terminal emulation on both the BIOS and your terminal. Some BIOSs are known to not render the Solarflare Boot Manager properly when using vt100 terminal emulation.

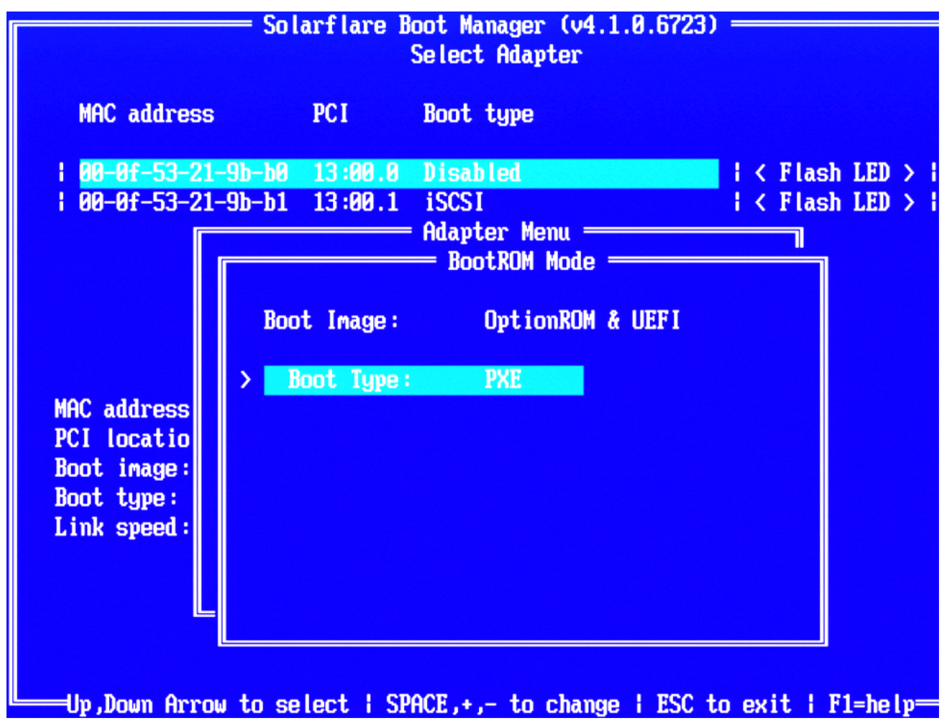
1 On starting or re-starting the server, press **Ctrl+B** when prompted. The Solarflare Boot Configuration Utility is displayed.



2 Use the arrow keys to highlight the adapter you want to boot via PXE and press Enter. The **Adapter Menu** is displayed.



3 From the **Boot Mode** option, press the arrow keys to change the **Boot Image** and/or the **Boot Type**.



4 From the **Boot Type**, press **Space** until PXE is selected.

5 Solarflare recommend leaving the **Adapter Options** and **BIOS Options** at their default values. For details on the default values for the various adapter settings, see Table 112 on page 455.

11.3 iSCSI Boot

Introduction

Solarflare adapters support diskless booting to a target operating system over Internet Small Computer System Interface (iSCSI). iSCSI is a fast, efficient method of implementing storage area network solutions.

The Boot ROM in the Solarflare adapter contains an iSCSI initiator allowing the booting of an operating system directly from an iSCSI target.

NOTE: Adapter teaming and VLANs are not supported in Windows for iSCSi remote boot enabled Solarflare adapters. To configure load balancing and failover support on iSCSI remote boot enabled adapters, you can use Microsoft MultiPath I/O (MPIO), which is supported on all Solarflare adapters.

11.4 Configuring the iSCSI Target

To the server (iSCSI initiator), the iSCSI target represents the hard disk from where the operating system is booted from. To enable connections from the server, you will need to allocate and configure a logical unit number (LUN) on an iSCSI target. The server (iSCSI initiator) will see the LUN as a logical iSCSI device and will attempt to establish a connection with it. You may need to enter details of the Solarflare adapter ID (MAC address) and other details to validate the connection.

Refer to the iSCSI target documentation for details on how to configure your target.

11.5 Configuring the Boot ROM

The server (iSCSI initiator) needs to contain at least one Solarflare network adapter. To enable the adapter for iSCSI booting, you will need to configure the Boot ROM with the correct initiator, target and authentication details. This can also be configured via the sfboot command line tool on all platforms, and through SAM on Windows.

For Windows, see Sfboot: Boot ROM Configuration Tool on page 183

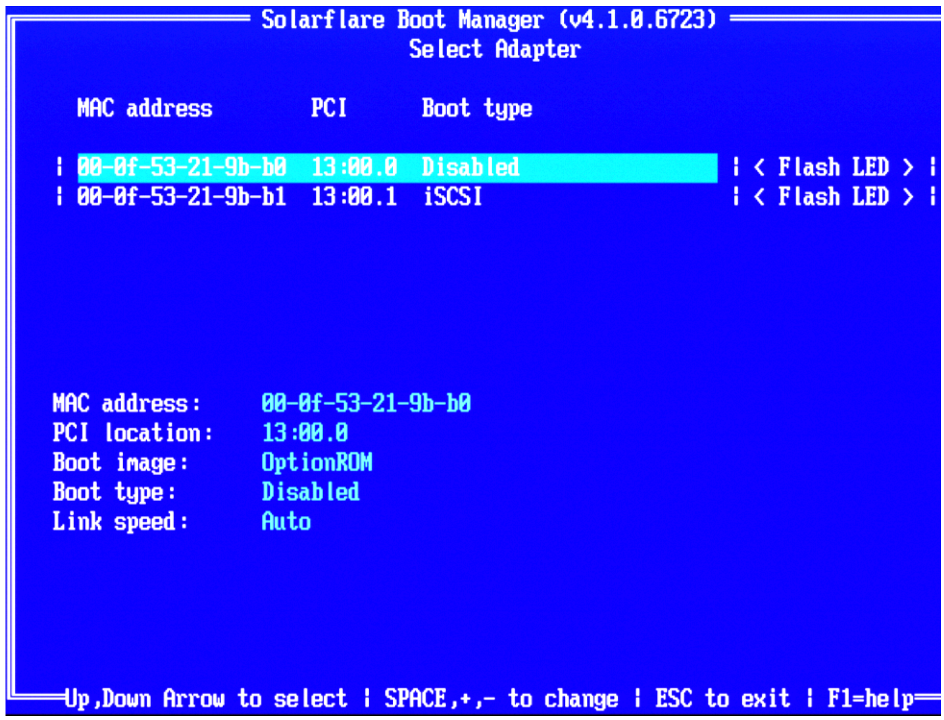
For Linux, see Configuring the Boot ROM with sfboot on page 72

For VMware, see Solarflare Utilities Package on page 270

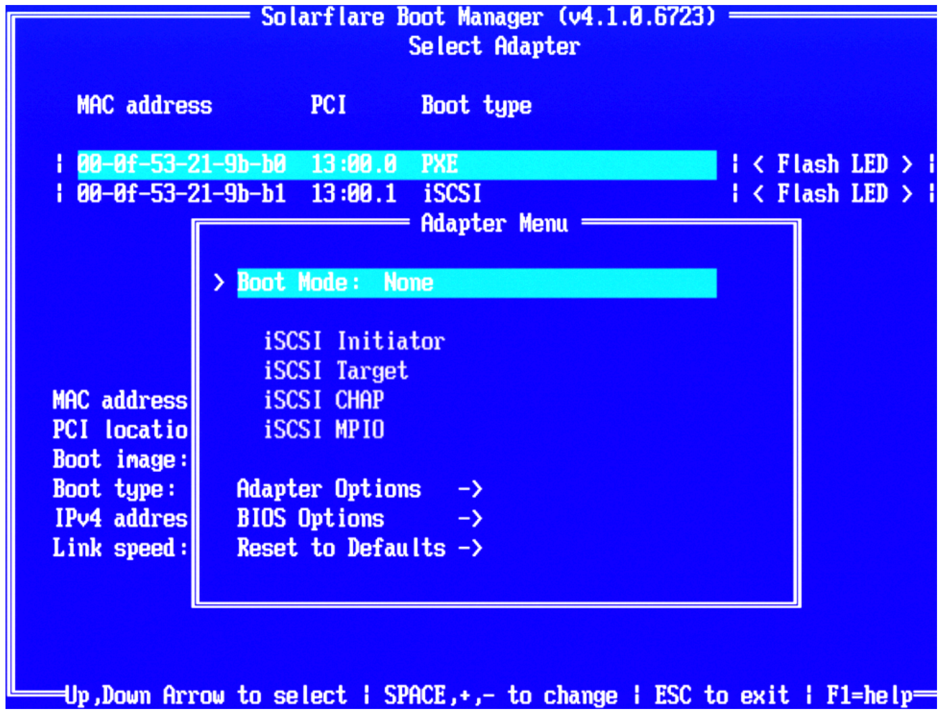
For SAM, see Using SAM for Boot ROM Configuration on page 170

1 Start or re-start the iSCSI initiator server and when prompted, press **Ctrl+B**. The Solarflare Boot Configuration Utility will display.

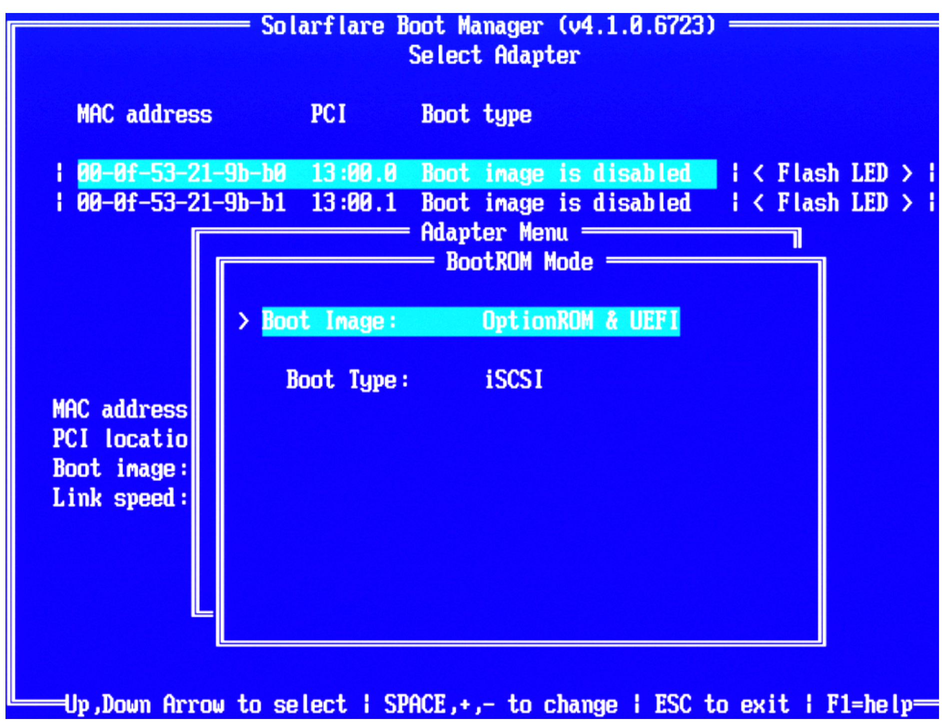
NOTE: If the BIOS supports console redirection, and you enable it, then Solarflare recommends that you enable ANSI terminal emulation on both the BIOS and your terminal. Some BIOSs are known to not render the Solarflare Boot Manager properly when using vt100 terminal emulation.



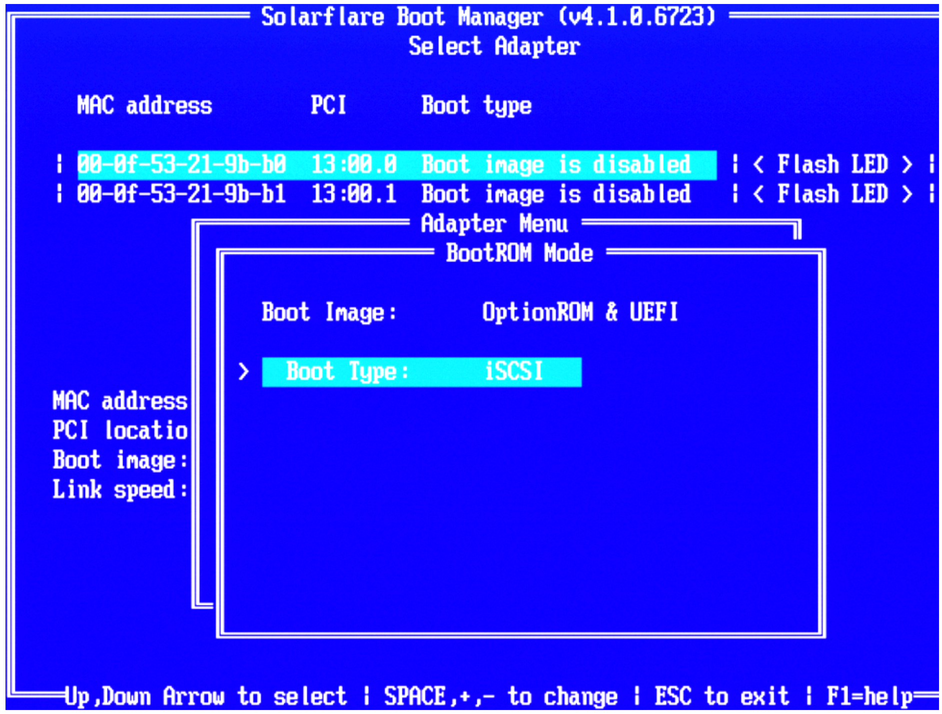
2 Highlight the adapter to configure and Press **Enter**. The **Adapter Menu** is displayed.



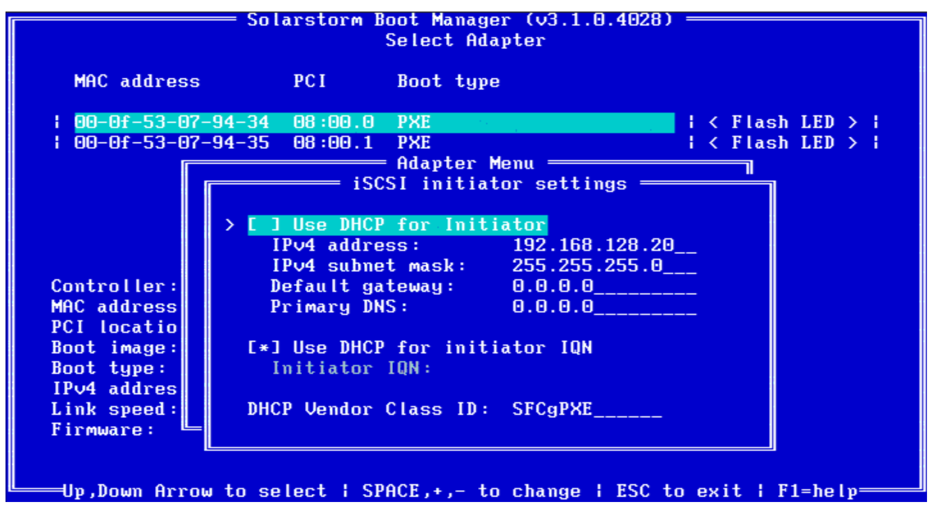
3 From the **BootROM Mode** option, press the space bar to change the **Boot Image** and or the **Boot Type**.



From the **Boot Type**, press **Space** until **iSCSI** is selected.



Press Enter. The iSCSI Initiator options are displayed.



Use DHCP for Initiator is selected as default. This instructs the adapter to use a DHCP server to obtain the relevant details to configure the Solarflare Boot ROM iSCSI initiator. See DHCP Server Setup on page 445. If you are not using DHCP, press enter and add the following details:

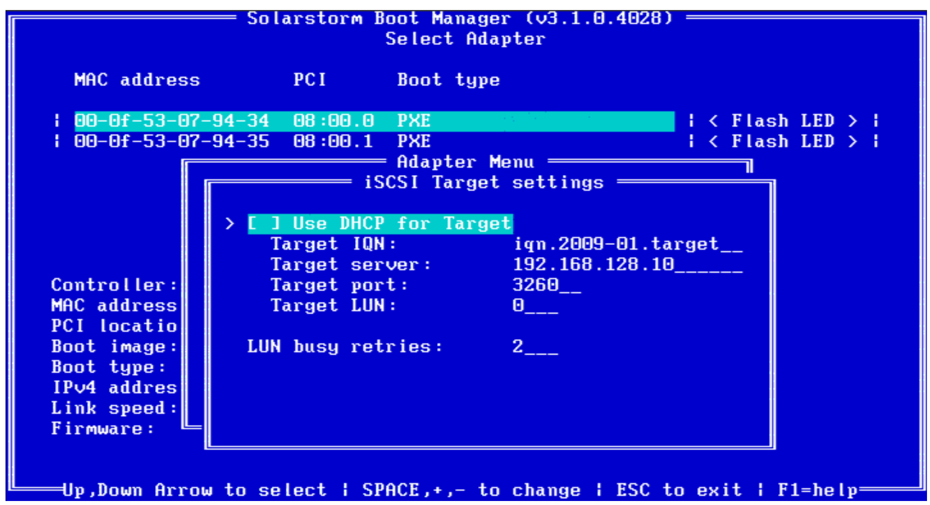
- **IP address**: IP address of the Solarflare adapter to use at boot time.
- **Netmask**: IP address subnet mask.
- **Gateway**: Network gateway address. A gateway address may be required if the iSCSI target is on a different subnet from the initiator.
- **Primary DNS**: Address of a primary DNS server.

Use DHCP initiator IQN is selected as default. This instructs the adapter to obtain the iSCSI initator IQN from the DHCP server via option 43.203 or if this is not available to construct an iSCSI initator IQN from option 12. See DHCP Server Setup on page 445 for more details

- **Initiator IQN**: The iSCSI initiator IQN of the Solarflare adapter if you are not using DHCP to obtain the iSCSI initiator IQN.
- **DHCP Vendor Class ID:** If you are using DHCP to obtain the iSCSI initiator IQN, the adapter will use DHCP option 43 to try and obtain this information from the DHCP server. DHCP option 43 is described as “vendor specific information” and requires that the vendor id (DHCP option 60) configured at the DHCP server matches the vendor id configured in the Boot ROM. See DHCP Option 60, Vendor ID on page 446 for more details. Solarflare strongly recommend leaving this setting as “SFCgPXE”.

Press **Esc** to return to the Adapter Menu.

4 Highlight **iSCSI Target** and press **Enter**.



By default, the adapter uses DHCP to obtain details about the iSCSI target. See DHCP Server Setup on page 445 for details of how to enter this information into your DHCP server. If you are not using DHCP, press Enter and enter the following details:

Target IQN: Name of the iSCSI Target. The format of this is usually IQN or EUI: refer to your iSCSI Target documentation for details of how to configure this setting.

Target Server: IP address or DNS name of the target server.

TCP port: The TCP Connection port number to connect to on the iSCSI target (required). Default: 3260.

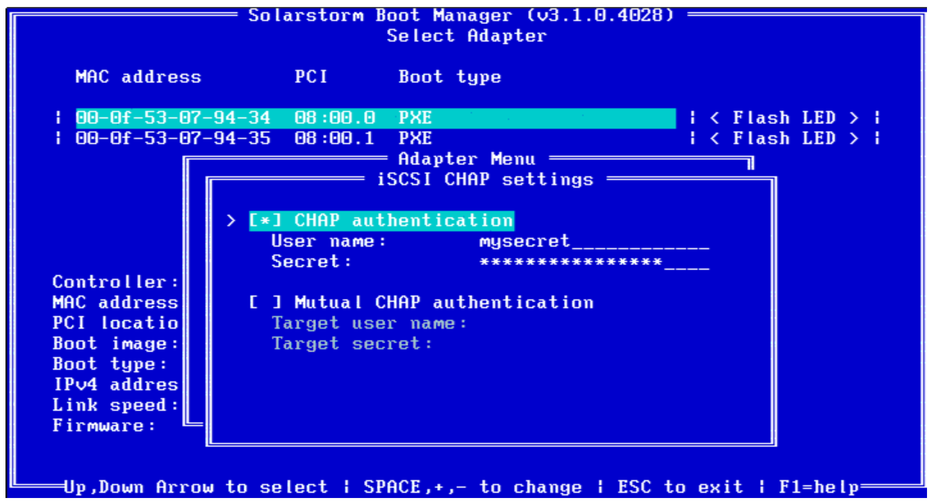
Boot LUN: Logical unit number (LUN) of the iSCSI Target (required). Default: 0. Values: 0-255.

The following settings can also be configured:

LUN busy retry count: Number of times the initiator will attempt to connect to the iSCSI target. Default: 2. Range: 0-255.

Press **Esc** to return to the Adapter Menu.

5 If CHAP authentication is required, highlight **iSCSI CHAP** and press **Enter**.



Enter **User Name** and **Secret** information.

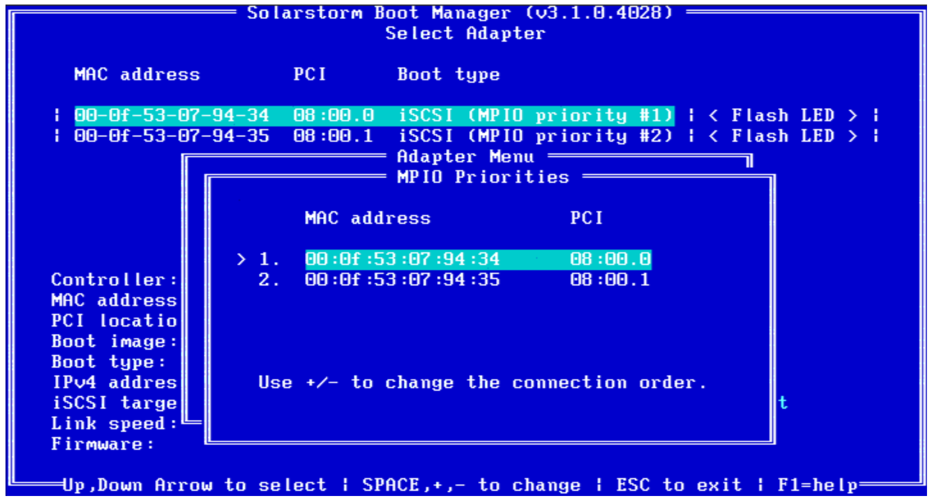
If Mutual CHAP is required as well as CHAP, highlight this option and press **Enter**.

Enter **Target user name** and **Target secret** information.

Press **Esc** to return to the Adapter Menu.

6 MPIO can be configured to provide alternative paths to the iSCSI target to increase the resilience to network outages. The MPIO priority defines the order the configured adapters are used to attempt to connect to the iSCSI target.

You can use the MPIO option to configure the MPIO rank for all adapters. Ensure all adapters to be used for MPIO are correctly configured for **iSCSI** boot. Highlight **iSCSI MPIO** and press **Enter**.



Note that you can set the MPIO rank for all Solarflare adapters from the configuration menus of any of the available adapters.

Press Esc to return to the Adapter menu.

7 When you have finished, select **Save and exit**.

11.6 DHCP Server Setup

If your network has a DHCP server, the adapter Boot ROM can be configured so the adapter is able to dynamically retrieve iSCSI initiator and target configurations from it on startup.

DHCP Option 17, Root Path

The root path option can be used to describe the location of the iSCSI target. This information is used when configuring the boot ROM (see step 4 on page 443).

The iSCSI root path option configuration strings uses the following format:

”iscsi:“<server name or IP address>”:“<protocol>”:“<port>:<LUN>”:“<targetname>

- **Server name**: FQDN or IP address of the iSCSI target.
- **Protocol**: Network protocol used by iSCSI. Default is TCP (6).
- **Port**: Port number for iSCSI. Default is 3260.
- **LUN**: LUN ID configured on the ISCSI target. Default is zero.
- **Target name**: iSCSI target name to uniquely identify the iSCSI target in IQN format. Example:

  `iqn.2009-01.com.solarflare.`

DHCP Option 12, Host Name

If the adapter is configured to obtain its iSCSI initiator IQN via DHCP and option 43.203 is not configured on your DHCP server, then the adapter will use the DHCP host name option to construct an iSCSI initiator IQN.

DHCP Option 3, Router List

If the iSCSI initiator and iSCSI target are on different subnets, configure option 3 with the default gateway or router IP address.

DHCP Option 43, Vendor Specific Information

Option 43 provides sub-options that can be used to specify the iSCSI initiator IQN and the iSCSI target IQN.

- Option 43.201 provides an alternative to option 17 to describe the location of the iSCSI target. The format for the iSCSI target IQN is the same as described for DHCP option 17
- Option 43.203 provides a method of completely defining the iSCSI initiator IQN via DHCP.

Table 111: DHCP Option 43 Sub-Options

Sub-Option	Description
201	First iSCSI target information in the standard root path format “iscsi:”<servername>”:”<protocol>”:”<port>”:”<LUN>”:”<targetname>
202	Secondary target IQN. This is **Not** supported.
203	iSCSI initiator IQN

NOTE: If using Option 43, you will also need to configure Option 60.

DHCP Option 60, Vendor ID

When using DHCP option 43 you must also configure option 60 (Vendor id). DHCP option 43 is described as “vendor specific information” and requires that the vendor id (DHCP option 60) configured at the DHCP server matches the vendor id configured in the Boot ROM. By default the Boot ROM uses the vendor id `SFCgPXE`.

11.7 Installing an Operating System to an iSCSI target

Introduction

This section contains information on setting up the following operating systems for iSCSI booting:

- Installing Windows Server 2008 R2 on page 447
- Installing SUSE Linux Enterprise Server on page 448
- Installing Red Hat Enterprise Linux on page 452

Installing Windows Server 2008 R2

To install Windows Server 2008 R2 (with or without a local drive present):

Prerequisites

- Configure the iSCSI target and Solarflare adapter Boot ROM, as described in Configuring the iSCSI Target on page 439 and Configuring the Boot ROM on page 439.
- Copy the correct Solarflare driver files to a floppy disk or USB flash drive. Refer to

Steps to Install

1. Insert the Windows Server 2008 R2 DVD and restart the server. The Windows Server setup program will start.
2. Click **Load Driver** and browse to Solarflare drivers folder on the floppy or USB driver. Load the Solarflare VBD driver (if needed locate the INF file `netSFB*.inf`).
3. Click **Load Driver** a second time and browse to Solarflare drivers folder on the floppy or USB driver. Load the Solarflare NDIS driver (if needed locate the INF file `netSFN*.inf`).
4. After loading the drivers, click **Refresh** to refresh the list of available partitions.
5. Select the target partition that is located on the iSCSI target and continue installing Windows on the target.
6. Remove the Solarflare drivers disk.

Installing SUSE Linux Enterprise Server

For complete installation instructions, consult the relevant Novell documentation:

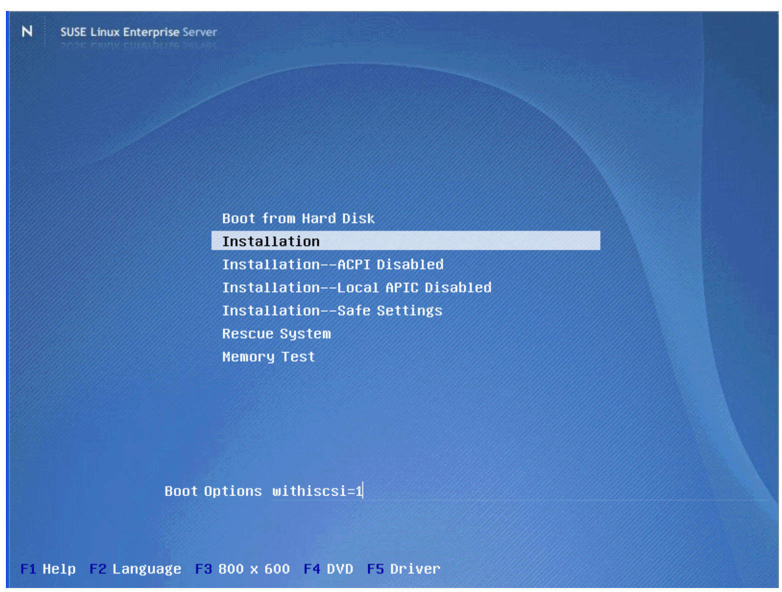
http://www.novell.com/documentation/

Prerequisites

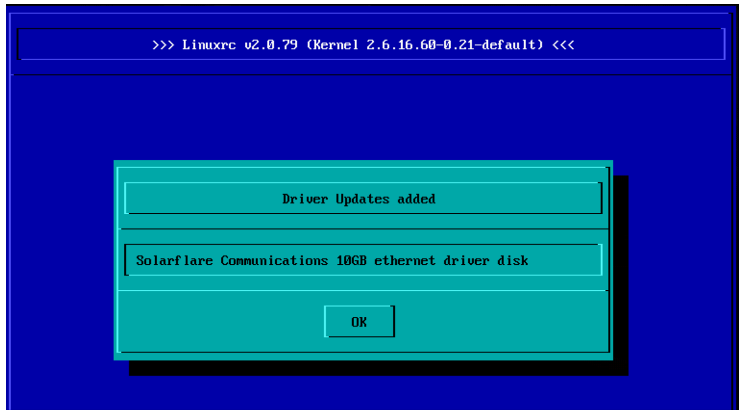
- Ensure you have all your iSCSI configuration information for the iSCSI target and iSCSI initiator. You will need to enter these details during the installation process.
- Ensure that the Solarflare Boot ROM is configured for iSCSI boot and can login to the selected iSCSI target.
- You will need the appropriate Solarflare driver disk. See Driver Disks for Unattended Installations on page 52 for more details.

Installation Process

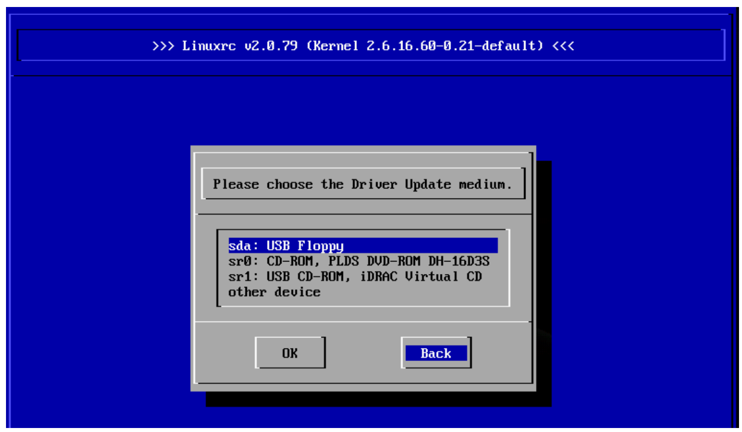
1. Boot from your DVD.
2. From the first installation screen, press **F5 Driver** and select **Yes**. Press **Return**.
3. Highlight Installation and enter the following Boot Option: `withiscsi=1`



4 If you see a **Driver Updates added** screen for a Solarflare driver disk, click **OK**.

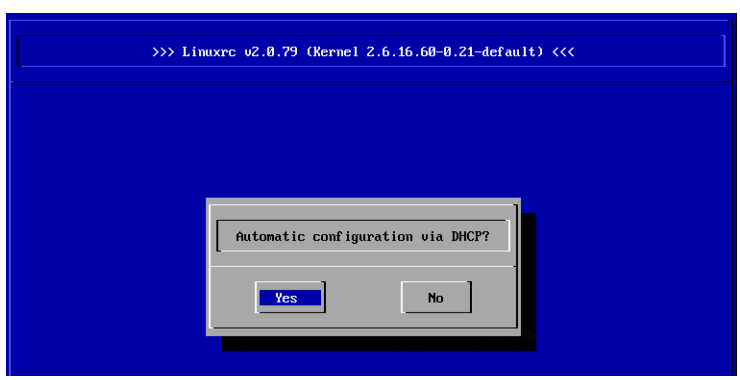


5 When prompted for further driver updates, click **Back** to return to the installer.



6 Select the network device. To check which is the Solarflare network adapter, press **Ctrl+Alt+F4**. To return to the Installation screen, press **Ctrl+Alt+F1**.

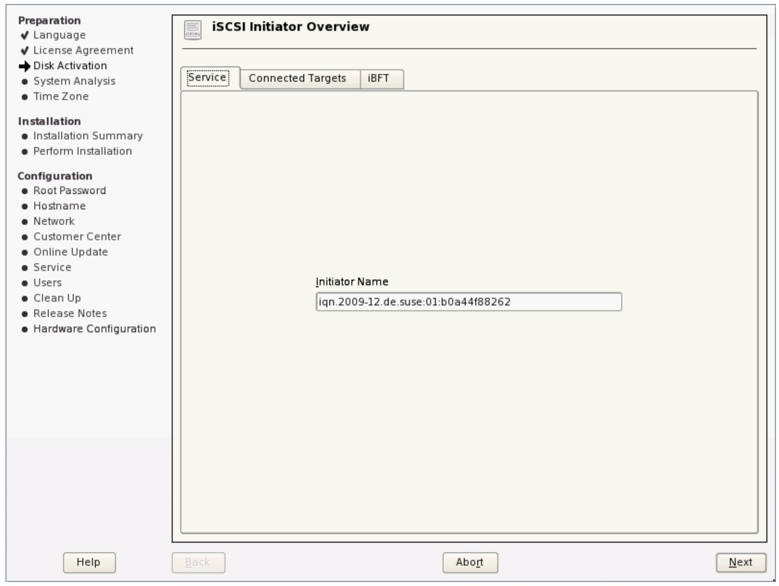
7 Select **Yes** from the **Automatic configuration via DHCP?** option.



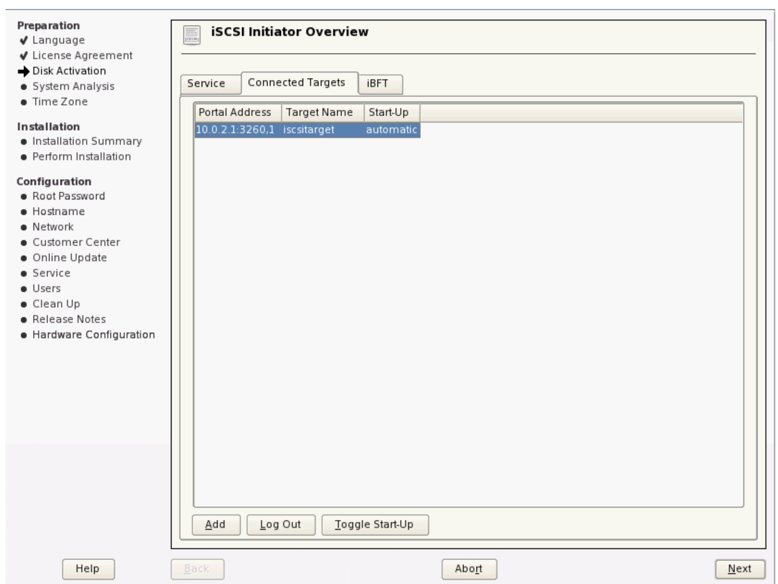
8 Follow the install steps until you reach the **Disk Activation** stage.

9 From the **Disk Activation > iSCSI Initiator Overview** stage, click the **Service** tab.

10 Note the SUSE auto generated **Initiator Name**, or replace this with your own.



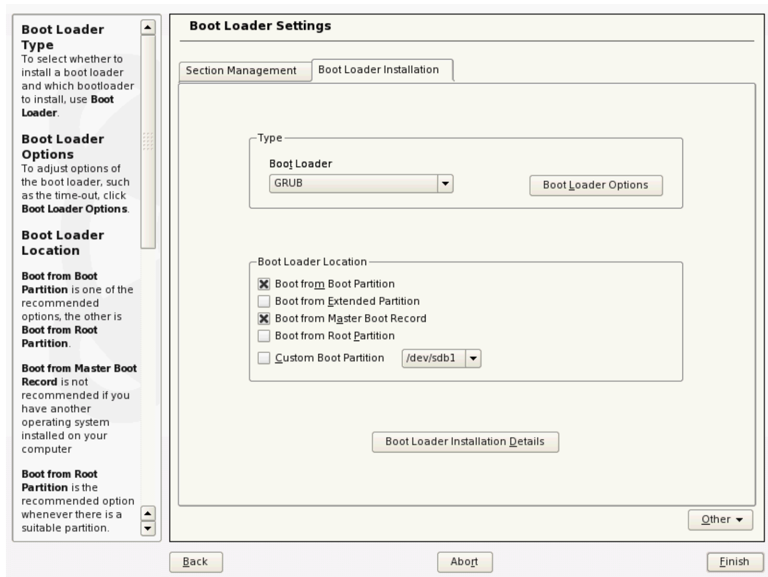
11 Click the **Connected Targets** tab. The target should be listed.



12 Ensure the **Start-Up** mode is correct for your installation. For SUSE Enterprise Linux Server 10, it should be **automatic**. For SUSE Enterprise Linux Server 11, it should be **onboot**. Click **Next** to continue.

13 From the **Installation Settings** screen, select the **Expert** tab, then click the **Booting** hyperlink. Select the **Boot Loader installation** tab.

14 Select **Boot from Master Boot Record** as well as **Boot from Boot Partition**. Click **Finish**.



15 When you reach the **Installation Summary** screen, select **Partitioning** to verify the installation device. Ensure that the desired iSCSI target is selected for the installation target. Click **Next**.

16 When the first stage of the install is complete, the system will reboot. Continue to the **Configure Boot Device Order** to add the iSCSI target and continue the installation process.

Following the server reboot, check that the iSCSI disk is in an appropriate place in the BIOS boot order. It may be displayed as 'Solarflare Boot Manager' or 'Hard drive C:', as there is no physical hard disk in the system.

If you don’t see either of the above options, check the messages output from the Solarflare Boot ROM during the boot process for DHCP or iSCSI login failures indicating a Boot ROM or DHCP configuration issue.

Installing Red Hat Enterprise Linux

For complete installation instructions, consult the relevant Red Hat documentation:

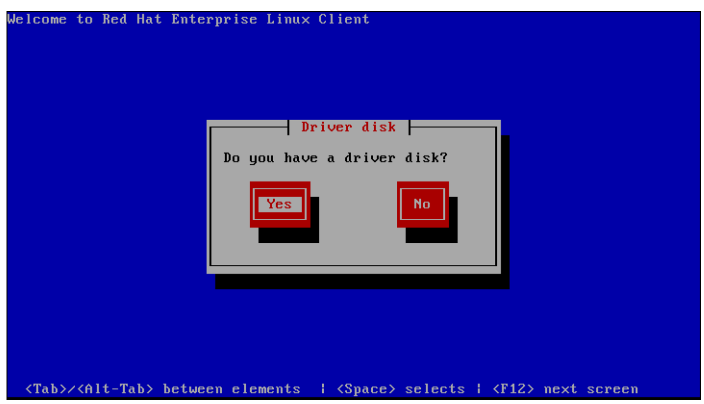
http://www.redhat.com/docs/manuals/enterprise/

Prerequisites

- Ensure you have all your iSCSI configuration information for the iSCSI target and iSCSI initiator. You will need to enter these details during the installation process.
- Ensure that Solarflare Boot ROM is configured for iSCSI boot and can login to the selected iSCSI target.
- You will need the appropriate Solarflare driver disk. See Driver Disks for Unattended Installations on page 52 for more details.

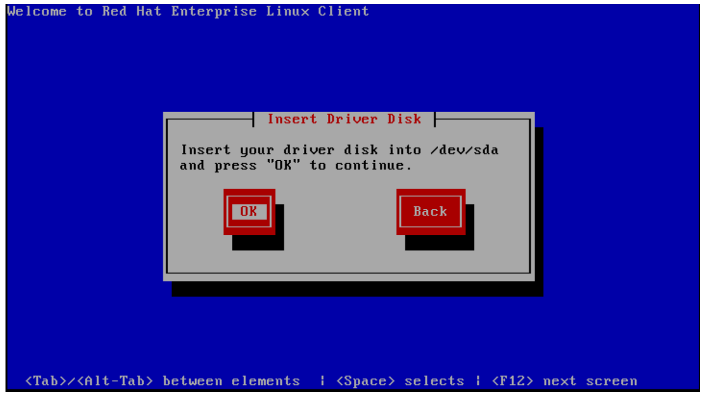
Installation Process

1. Boot from your DVD.
2. From the first installation screen, enter `linux dd`. Press **Return**.
3. When asked if you have a driver disk, select Yes.



4. A **Driver Disk Source** window is displayed. Select the source and select **Yes**.

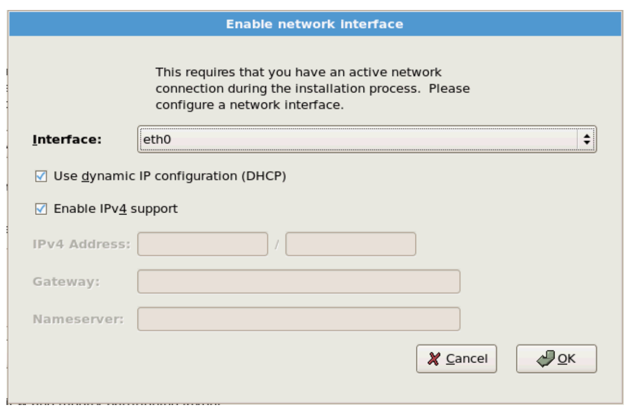
5 You will then be prompted to Insert your driver disk into the source specified in step 4.



6 You will be prompted to load more driver disks. Select **No**.

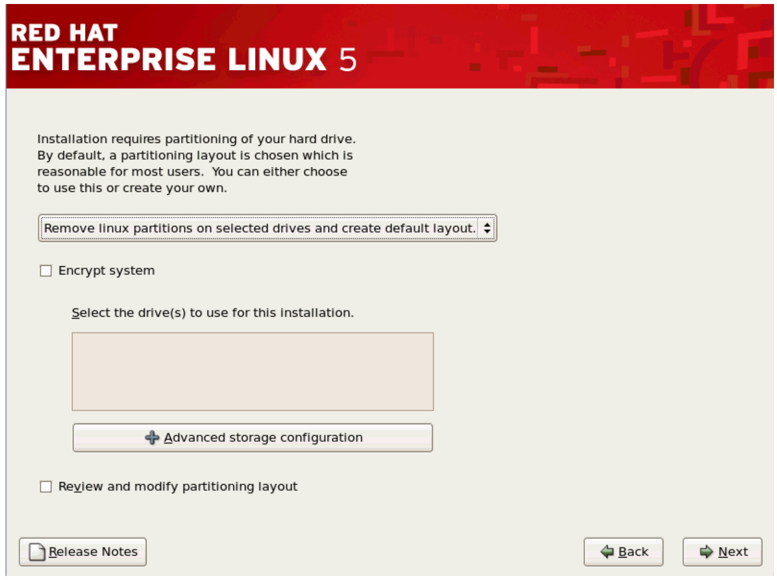
7 A **CD Found** screen will prompt you to test the CD before installation. Select **Skip**.

8 When an **Enable network interface** screen displays, select the **Solarflare adapter interface**. Ensure that **Use dynamic IP configuration (DHCP)** is selected.



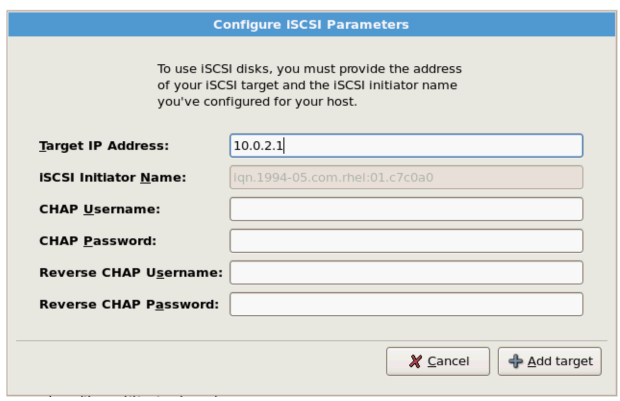
9 Follow the standard Red Hat installation steps until you reach the **Disk Partitioning Setup** menu.

10 From the **Disk Partitioning** menu, select **Advanced storage configuration** to add the iSCSI target.



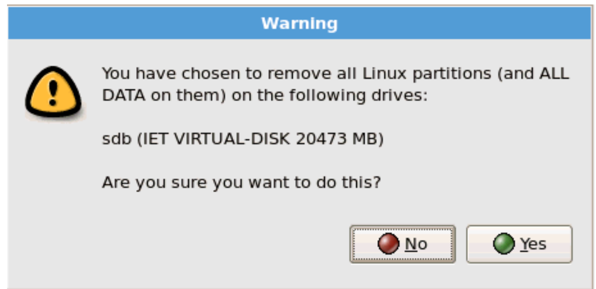
11 In the **Advanced Storage Options** window, select **iSCSI** and click **Add iSCSI target**.

12 In the **Configure iSCSI Parameters** dialog box, enter your **Target IP Address**.



13 Click **Add target** to continue.

14 Click **Next** from the screen in step 10. A warning is displayed regarding the removal of all partitions. As the assumption is made that this is a clean install, click **Yes**.



If the drive(s) used for installation is displaying the correct device for the iSCSI LUN you configured, proceed with the rest of the installation. If the device configuration displayed is incorrect, check your details.

Following the server reboot, check that the iSCSI disk is in an appropriate place in the BIOS boot order. It may be displayed as 'Solarflare Boot Manager' or as 'Hard drive C:', as there is no physical hard disk in the system.

If you don’t see either of the above options, check the messages output from the Solarflare Boot ROM the boot process for DHCP or iSCSI login failures indicating a Boot ROM or DHCP configuration issue.

11.8 Default Adapter Settings

Table 112 lists the various adapter settings and their default values. These are the values used if you select **Reset to Defaults** from the Boot Configuration Utility, or click **Default** from SAM.

Table 112: Default Adapter Settings

Setting	Default Value
Boot Image	Disabled
Link speed	Auto
Link up delay	5 seconds
Banner delay	2 seconds
Boot skip delay	5 seconds
Boot Type	PXE
Initiator DHCP	Enabled
Initiator-IQN-DHCP	Enabled
LUN busy retry count	2 seconds

Table 112: Default Adapter Settings

Setting	Default Value
Target-DHCP	Enabled
TCP port	3260
Boot LUN	0
DHCP Vendor	SFCgPXE
MPIO attempts	3
MSIX Limit	32

11.9 Multiple PF - PXE Boot

Using the sfboot configuration utility v4.5.0 (or later version) it is possible to PXE boot when multiple Physical Functions have been enabled. The primary function on each port (PF0/PF1) is a privileged function and can be selected for configuration. Other PFs inherit from their privileged function- so, for example, with two physical ports and 2 PFs per port:

- PF0 and PF2 will have the same boot-type
- PF1 and PF3 will have the same boot-type

Configuration of non-privileged functions is not currently supported.

In the following example 2 PFs (and 2 VFs) are enabled for each physical interface.

```
# sfboot Solarflare boot configuration utility [v4.5.0]

eth2:
  Boot image                            Option ROM only
    Link speed                          Negotiated automatically
    Link-up delay time                  5 seconds
    Banner delay time                   2 seconds
    Boot skip delay time                5 seconds
    Boot type                           Disabled
  Physical Functions per port           2
  MSI-X interrupt limit                 32
  Number of Virtual Functions           2
  VF MSI-X interrupt limit              8
  Firmware variant                      full feature / virtualization
  Insecure filters                      Disabled
  MAC spoofing                          Disabled
  VLAN tags                             100,110
  Switch mode                           Partitioning with SRIOV

eth3:
  Boot image                            Option ROM only
    Link speed                          Negotiated automatically
    Link-up delay time                  5 seconds
    Banner delay time                   2 seconds
```



```
    Boot skip delay time                5 seconds
    Boot type                           Disabled
  Physical Functions per port           2
  MSI-X interrupt limit                 32
  Number of Virtual Functions           2
  VF MSI-X interrupt limit              8
  Firmware variant                      full feature / virtualization
  Insecure filters                      Disabled
  MAC spoofing                          Disabled
  VLAN tags                             100,110
  Switch mode                           Partitioning with SRIOV


eth4:
  Interface-specific boot options are not available. Adapter-wide options
are available via eth2 (00-0F-53-25-39-90).

eth5:
  Interface-specific boot options are not available. Adapter-wide options
are available via eth2 (00-0F-53-25-39-90).
```

The Solarflare Boot Manger GUI utility (CTRL-B) will list all PFs, but only the privileged PFs can be selected for configuration.

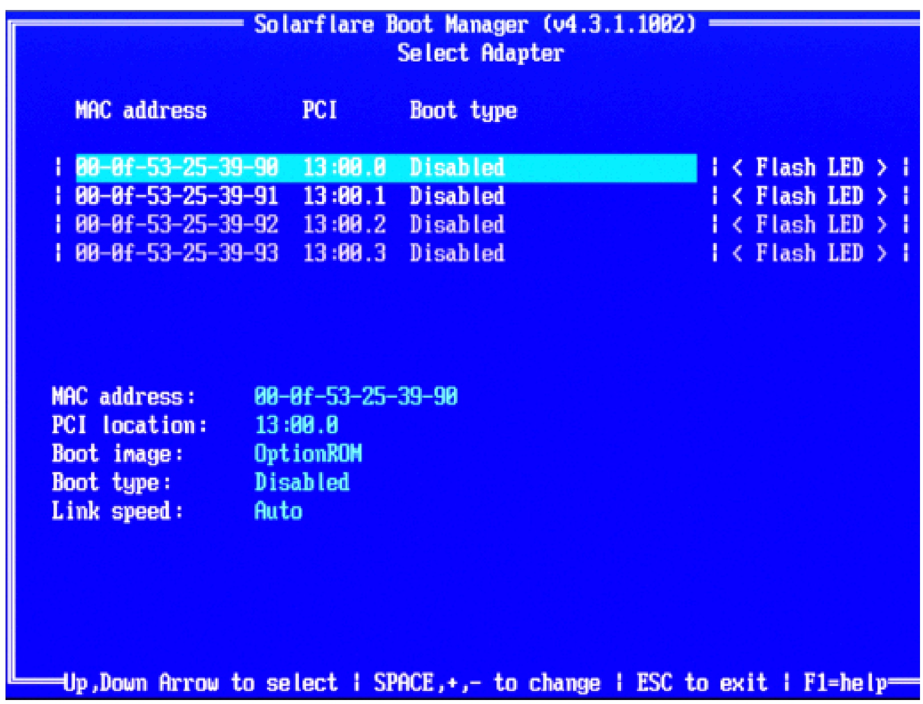


Figure 64: Boot Manager lists multiple PFs

Index

T

U

V

W

