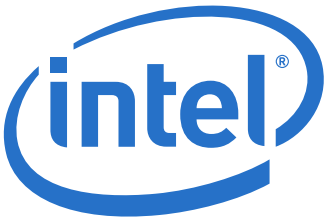


True Scale Fabric Suite Fabric Manager

User Guide

July 2015

Order Number: G91918004US

Contents

Figures

Tables

Revision History

Date	Revision	Description
May 2013	001US	Initial release
January 2014	002US	Updated document File Info meta data
August 2014	003US	Updated Support link in Section 1.4, "Technical Support" on page 18.
July 2015	004US	Document revision incremented for release 7.4

§ §

1.0 Introduction

This guide discusses the Intel® True Scale Fabric Suite Fabric Manager (FM). The FM provides comprehensive control of administrative functions using a commercial-grade subnet manager. With advanced routing algorithms, powerful diagnostic tools and full subnet manager failover, FM simplifies subnet, fabric, and individual component management, making even the largest fabrics easy to deploy and optimize.

1.1 Intended Audience

This manual is intended to provide network administrators and other qualified personnel a reference for configuration and administration task information for the FM.

1.2 Related Materials

- *Intel® True Scale Fabric Software Installation Guide*
- *Intel® True Scale Fabric Suite FastFabric User Guide*
- *Intel® True Scale Fabric Suite FastFabric Command Line Interface Reference Guide*
- *Intel® True Scale Fabric Suite Fabric Viewer Online Help*
- *Intel® True Scale Fabric Suite Software Release Notes*
- *Intel® True Scale Fabric OFED+ Host Software Release Notes*

1.3 Documentation Conventions

This guide uses the following documentation conventions:

- **NOTE:** provides additional information.
- ***CAUTION!*** indicates the presence of a hazard that has the potential of causing damage to data or equipment.
- **WARNING!!** indicates the presence of a hazard that has the potential of causing personal injury.
- Text in blue font indicates a hyperlink (jump) to a figure, table, or section in this guide, and links to Web sites are shown in underlined blue. For example:
  - Table 9-2 lists problems related to the user interface and remote agent.
  - See "Installation Checklist" on page 3-6.
  - For more information, visit www.intel.com.
- Text in **bold** font indicates user interface elements such as a menu items, buttons, check boxes, or column headings. For example:
  - Click the **Start** button, point to **Programs**, point to **Accessories**, and then click **Command Prompt**.
  - Under **Notification Options**, select the **Warning Alarms** check box.
- Text in `Courier` font indicates a file name, directory path, or command line text. For example:
  - To return to the root directory from anywhere in the file structure: Type `cd /root` and press **ENTER**.
  - Enter the following command: `sh ./install.bin`
- Key names and key strokes are indicated with **uppercase**:
  - Press **ctrl+P.**

  - Press the **up arrow** key.
- Text in *italics* indicates terms, emphasis, variables, or document titles. For example:
  - For a complete listing of license agreements, refer to the *Intel® Software End User License Agreement*.
  - What are *shortcut keys*?
  - To enter the date type *mm/dd/yyyy* (where *mm* is the month, *dd* is the day, and *yyyy* is the year).
- Topic titles between quotation marks identify related topics either within this manual or in the online help, which is also referred to as *the help system* throughout this document.

1.4 Technical Support

Intel True Scale Technical Support for products under warranty is available during local standard working hours excluding Intel Observed Holidays. For customers with extended service, consult your plan for available hours. For Support information, see the Support link at www.intel.com/truescale.

1.5 License Agreements

Refer to the *Intel® Software End User License Agreement* for a complete listing of all license agreements affecting this product.

§ §

2.0 Overview

This section gives an overview of the True Scale Fabric Suite Fabric Manager (FM)

2.1 Component Overview

The FM is a set of components that perform various functions for the management of the fabric. These components consist of:

- Subnet Manager (SM)
- Subnet Administration (SA)
- Performance Manager (PM)
- Baseboard Manager (BM)
- Fabric Executive (FE)

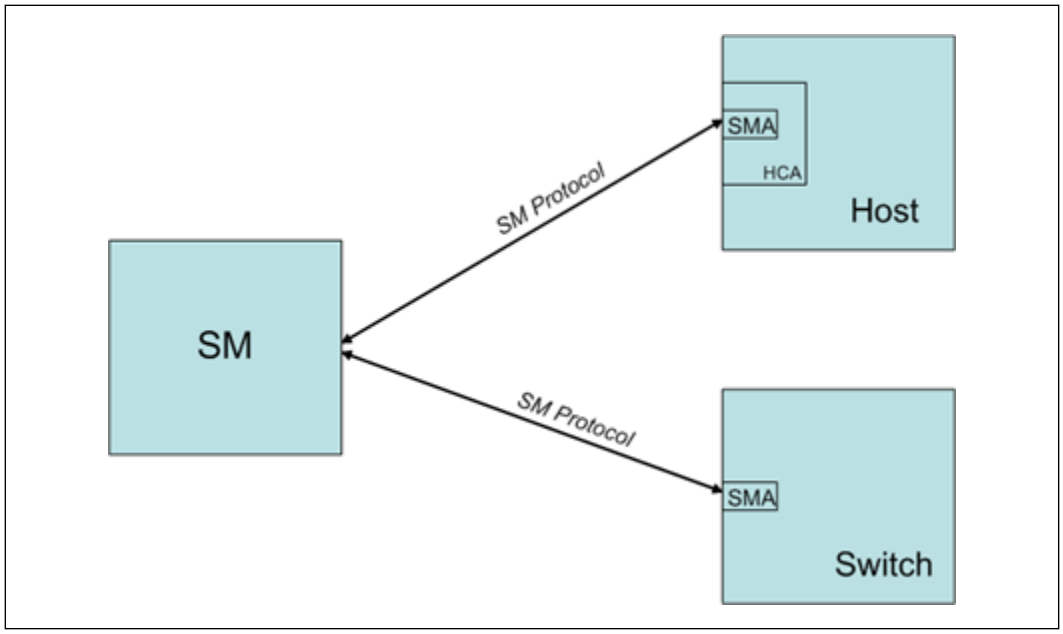
2.1.1 Subnet Manager

The Subnet Manager component performs all of the necessary subnet management functions as specified by the *InfiniBand** *Architecture Specification Release 1.2.1,* volume 1, chapter 14. Primarily, the SM is responsible for initializing the fabric and managing its topology. Some of its tasks include:

- Link and port initialization
- Route and path assignments
- Local Identifier (LID) assignments
- Switch forwarding table programming
- Programming True Scale vFabrics™
- Sweeping the fabric to discover topology changes, managing those changes when nodes are added and/or deleted

The SM communicates with the Subnet Management Agent (SMA) on each node of the fabric, using the SM packets. Refer to Figure 1.

Figure 1. FM Subnet Manager



When there is more than one SM on a given fabric, arbitration takes place between them to assign one as master and the rest as standby SMs. Refer to "FM Role Arbitration" on page 23 for SM arbitration rules.

SMs communicate with each other over the Intel® True Scale Solution for Infiniband* Architecture. If the standby SM's detect that the master SM can not be found (For example, the node on which it is running has lost communication with the fabric, or it has been shut down), then the standby SMs re-enter arbitration to choose a new master SM.

When running SM on a switch and on a host simultaneously requires that the ConfigConsistencyCheckMethod be set to 1 instead of the default 0 as described in Section 4.1.2, "Shared Parameters" on page 67. If the ConfigConsistencyCheckMethod method is left as default 0 then the standby SM will become Inactive.

2.1.2 Subnet Administration

The Subnet Administration function acts in tight coordination with the SM to perform data storage and retrieval of fabric information. The SM/SA is a single unified entity.

Through the use of SA messages, nodes on the fabric can gain access to fabric information such as:

- Node-to-node path information
- Fabric Topology and configuration
- Event notification
- Application Service information
- vFabric™ Information

2.1.3 Performance Manager

The Performance Manager component communicates with nodes to collect performance and error statistics. The PM communicates with the Performance Management Agent (PMA) on each node in the fabric, using the PM packets.

Examples of the type of statistics collected by the PM include:

- Link Utilization Bandwidth
- Link Packet Rates
- Link Congestion
- Number of Adaptive Routing changes to routing tables
- Error statistics, such as symbol errors or packet discards

The PM collects data on a per-port basis.

2.1.4 Baseboard Manager

The Baseboard Manager component communicates with nodes to provide an in-band mechanism for managing chassis. The BM communicates with the Baseboard Management Agent (BMA) on each node of the fabric, using the BM packets.

Examples of the type of hardware management that can be performed by the BM include:

- Retrieval of vital product data (VPD) (For example, serial number, manufacturing information)
- Retrieval of environmental data such as temperature and voltage sensor readings
- Adjustment of power and cooling resources (For example, fan speed adjustments)

2.1.5 Fabric Executive

The Fabric Executive component provides for out-of-band access to the FM. It permits the True Scale Fabric Suite Fabric Viewer (FV) utility to communicate over TCP/IP and access the rest of the FM.

The FE exposes a private TCP/IP socket, on which it listens for connect requests from the FV applications. The FV communicates with the FE using the TCP/IP protocol to retrieve fabric information for presentation to the FV user.

The FE then uses in-band InfiniBand* Technolgy packets to communicate with the other managers (SM, PM, BM). The FE does not have to run on the same host as the other managers.

2.2 Embedded and Host Solutions

The FM is able to be deployed as either a host-based or an embedded solution. The host solution uses a True Scale Fabric stack and an Host Channel Adapter (HCA) to access, and manage the fabric. The embedded solution resides on an internally managed switch, and uses a switch port to access and manage the fabric.

2.2.1 Host FM

The host FM deploys all FM components on a Linux server. The FM components are applications that run in the user space, and access the True Scale Fabric stack using the management datagram (MAD) interface provided by that stack.

The host FM can manage both small and large fabrics. It is ideal for managing large fabrics, as the FM software is able to make use of the large memory resources and high speed processor technology of standard servers.

The host FM is installed onto a Linux system as part of the Intel® True Scale Fabric Suite installation. The utilities are installed into the `/opt/ifs_fm/bin` directory. The configuration file is installed as `/etc/sysconfig/ifs_fm.xml`. It is started, restarted, and stopped using the `/etc/init.d/ifs_fm` utility (which standard linux commands such as "service" can use) with the "start", "restart", or "stop" parameter, respectively.

The Intel® FastFabric application can configure and control the host FM, query the SA, and analyze fabric configuration and status. In addition, there are a few host FM control applications which are installed in `/opt/ifs_fm/runtime` that will be discussed later in this guide.

2.2.2 Embedded FM

The embedded FM deploys all of the FM components as an embedded solution in a managed edge switch or the management module of a Intel® 12000 core switch. The embedded FM is able to manage small to moderate size fabrics, as the performance of the embedded software is limited by onboard memory and management processor constraints (refer to the *Intel® True Scale Fabric Switches 12000 Series Release Notes* for more information). For the small to moderate fabrics, the embedded FM is ideal in that the fabric is able to utilize an existing resource for the embedded FM purposes.

The embedded FM is accessed through the switch CLI using commands in the "SubnetManagement" command group, as well as a few screens in the Intel® Chassis Viewer GUI interface for a management module or edge switch. The majority of the configuration for the FM is accomplished using a `ifs_fm.xml` file which can be loaded onto the switch using the GUI or FastFabric.The CLI contains many commands that provide access to configuration items and operational status. Both the web interface and CLI provide controls for the embedded FM to be started automatically at chassis boot time. Additionally, they provide the ability to start and stop the embedded FM.

2.2.3 Choosing Between Host and Embedded FM Deployments

Please refer to the release notes of IFS software or embedded firmware for more guidance in choosing between deploying a host or embedded FM solution.

2.3 Multiple FMs in a Fabric

It is possible to deploy more than one FM in a fabric. Normally deploying more than one FM is done to provide redundancy. Multiple FMs are a way to ensure that management coverage of the fabric continues unabated in the case of the failure of one of the FMs.

2.3.1 FM State – Master and Standby

When multiple FMs are present in a fabric, one takes on the role of master FM through arbitration. All other FMs taken the role of standby FM and use SM-to-SM communication to monitor the presence of the master FM.

Note: It is important that the configuration of all SMs managing a fabric have the same configuration. It is especially important that fundamental parameters such as SubnetPrefix match. It is also recommended that all redundant SMs are at the same revision level. Failure to follow these recommendations can result in fabric disruptions when SM failover occurs. Using the Configuration Consistency Checking feature, the FM

can ensure that all redundant FMs have a comparable configuration. See "FM Configuration Consistency" on page 24 for more information.

2.3.2 FM Role Arbitration

FMs arbitrate for the master role using inband InfiniBand* Technolgy messages (For example, the SMInfo packet). The FMs use the value of the SM Priority parameter (a configurable value in the range 0-15) and then the GUID of the port on which the FM is running (for example, the switch or the HCA).

The arbitration rules are:

- The FM with the highest priority value is master FM
- In the case of FMs with equally high priority value, the lowest value GUID is the master FM

2.3.3 Master FM Failover

Failover from master to standby FM is essentially seamless. Since the master and standby FMs participate in a database synchronization mechanism, the new master FM is equipped with enough data to continue managing the fabric upon takeover.

The change of master FMs does not affect data traffic, nor does it affect communication between fabric nodes and the SM/SA.

2.3.4 Master Preservation – "Sticky" Failover

Multiple FMs in a fabric can be configured to support "sticky" failover. Normally when the user configures the FMs, using the SM Priority value, to designate the pecking order of master and standby FMs, that order always holds true during arbitration. Thus, upon failover of the master, a standby becomes the new master until the original master is restarted, whereby the original master reclaims the master role.

With the sticky failover feature, the FMs can be configured so that once a FM takes over the master role, it retains that role even if the original master returns to the fabric and arbitration rules dictate that it should reclaim the role. Using the sticky failover feature minimizes disruption to the fabric.

2.3.4.1 PM – Master, Standby, Failover, Sticky Failover

The PM is built into the SM, therefore the master PM will always be the exact same FM as the master SM.

2.3.4.2 BM – Master, Standby, Failover, Sticky Failover

The BM is built into the SM, therefore the master BM will always be the exact same FM as the master SM.

The sticky failover feature may be applied to the BM as well.

2.3.4.3 Sticky Failover and Elevated Priority

Elevated priority is supported for the SM/PM and BM. It is the priority the manager will run at, once it becomes master. This feature defaults to off (0).

By configuring an elevated priority, the user can configure a fabric to only failover managers when the master fails, rather than re-negotiating whenever a new manager comes online. This can be achieved by configuring the elevated priority to be higher than any manager's normal priority.

When elevated priority is configured, a renegotiation at normal priorities can be forced using the restore priority operation. This will reduce the master's priority back to normal and renegotiate for master. When the present master is not the preferred master, this will allow a failover back to the normally preferred master to occur. For a host FM this is accomplished using the `fm_cmd` utility. For an embedded FM, this is accomplished using the `smRestorePriority` chassis CLI command.

The intention of this feature is to allow failover to occur only once when a manager is experiencing intermittent failures, rather than repeatedly passing control back and forth, disrupting fabric operations. Therefore, it is a best practice to configure the SM, PM, and BM managers with the same elevated priority. However, this is not required.

A typical configuration would be an elevated priority of 15 for all FMs, a normal priority of 8 for the preferred master and a normal priority of 1 for the preferred standby.

Note: Using elevated priorities permits a secondary FM to boot first and then if the preferred primary FM boots afterward, the preferred primary will take over and elevate its priority. If the preferred primary boots first, it will run without an elevated priority until a secondary comes online, at which time the preferred primary will remain master and elevate its priority.

2.4 FM Configuration Consistency

When there are redundant FMs in a fabric, it is important they all have the same configuration for key fabric operational parameters. (However, other parameters, such as priority, can purposely be different between FMs.) This is required such that upon FM failover there will be no disruption nor changes in fabric operation. To ensure FM configuration consistency, there is an optional configuration check feature in the FM that is enabled by default. The Configuration Consistency Checking is applied to the Subnet Manager, Baseboard Manager, Performance Manager, and Fabric Executive.

When enabled, the configuration between redundant managers is checked using checksums. If an operational inconsistency is detected a message can be logged or the standby manager can be changed to inactive. See Appendix A for more information about CLI commands such as `config_diff` which can help compare FM configuration files.

When an inactivation is performed, for the SM, the standby SM state is set to InActive. For the PM and BM the standby shuts down with a log message. For the FE, inconsistencies only cause a warning log message.

2.4.1 Parameters Excluded from Configuration Consistency Checking

The following set of parameters are excluded from configuration consistency checking:

2.4.1.1 Common and Shared Configuration

The following parameters are excluded for all managers:

Start, Hca, Port, Debug, RmppDebug, Priority, ElevatedPriority, LogLevel, LogFile, LogMode, *_LogMask, SyslogMode, Name, PortGUID, CoreDumpLimit, CoreDumpDir

For Virtual Fabrics, only Enabled Virtual Fabrics, and the Applications and DeviceGroups which they contain are checked.

2.4.1.2 SM Configuration

The following SM specific parameters are excluded: TrapLogSuppressTriggerInterval, SmPerfDebug, SaPerfDebug, DebugDor, DebugVf, DebugJm, DebugLidAssign, LoopTestOn, LoopTestPackets, LoopTestFastMode, LID, DynamicPortAlloc, SaRmppChecksum

Also all MLIDShared, CongestionControl, and AdaptiveRouting parameters are ignored when the given section is not enabled.

2.4.1.3 BM Configuration

The following BM specific parameters are excluded:

DebugFlag

2.4.1.4 PM Configuration

The following PM specific parameters are excluded:

ThresholdsExceededMsgLimit (entire section), SweepErrorsLogThreshold

2.4.1.5 FE Configuration

The following FE specific parameters are excluded:

TcpPort

2.5 Congestion Control Architecture

The objective of True Scale Congestion Control Architecture (CCA) is to reduce the propagation of fabric congestion during oversubscription scenarios such as many-to-one traffic. By configuring CCA-specific switch and channel adapter parameters, the fabric can throttle the packet transmit rate on the sending hosts, thereby preventing the spread of congestion through the fabric. Refer to *InfiniBand* Architecture Specification Release 1.2.1*, *Volume 1,* Annex A10, for a complete description of Congestion Control.

Intel®'s implementation of CCA differs from the InfiniBand* Trade Association (IBTA) specification in three ways:

1. Intel®'s approach to fabric congestion is wholistic in nature and addresses congestion through a combination of features: Adaptive routing, dispersive routing and CCA.
2. The Intel® Adaptive Routing and Dispersive routing features are designed to optimize and balance traffic in the core ISLs of the fabric.
3. The Intel® CCA implementation in the FM is focused on preventing the spread of congestion into the core fabric from over subscribed HCAs (receivers of many to one traffic).

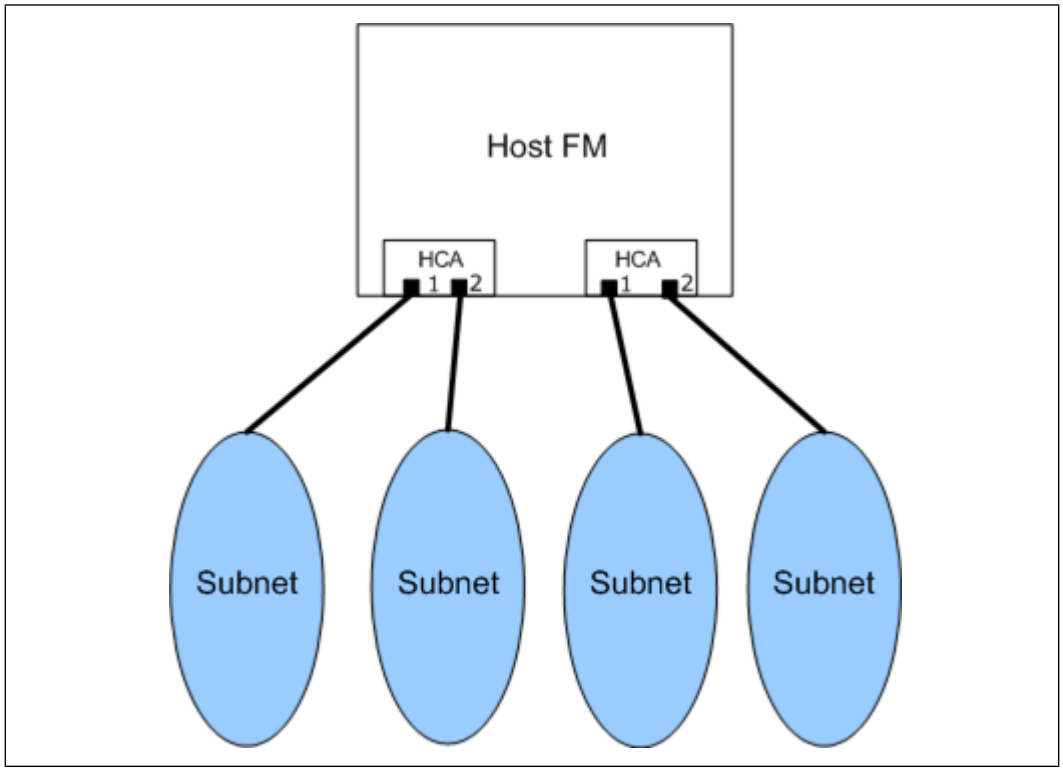
Intel® CCA allows CCA to be configured per MPI application through a simple set of PSM environment variables. Additional PSM debug logging can be enabled to facilitate analysis of PSM CCA events when needed.

The FM configuration file allows CCA settings to be configured for switches and non-Intel® IBTA compliant HCAs.

2.6 Multiple Subnet Support in Host FM

The host FM suite may be used to manage more than one subnet, sometimes described as “multiple planes”. Multiple planes are distinct isolated fabrics for which there is no interaction between ports in different fabrics. Refer to Figure 2.

Figure 2. Multiple Subnet Support in Host FM



As long as a host server has multiple HCA ports, the host FM can be configured to run separate “instances” over each HCA port available. Each HCA port is cabled to a separate plane, and the configuration denotes which port the instance should use to access the fabric.

The host FM can support four planes with four separate HCA ports on the server.

2.7 Fabric Viewer & FM

The Fabric Viewer (FV), a Java application that runs under the Windows or Linux operating system, communicates with the FM suite using the Fabric Executive (FE). The FV prompts the user for the IP address of the subnet, then opens a TCP/IP connection with the FE. The FE has access to the other FMs, namely the SM/SA, PM, and BM. In this way it is able to service requests from the FV and furnish relevant information to the FV about the fabric.

- The FV issues requests to the SM/SA to gather information about the subnet, such as topology and node information.
- The FV issues requests to the PM to aid in performance monitoring tasks.
- The FV uses requests to the BM to gather details about switch nodes, in order to aid in the Manage Chassis task.

- The FV can login to the host FM node or chassis to view and edit the FM configuration file.

Refer to the *Fabric Viewer Tasks* section of the *Intel® True Scale Fabric Suite Fabric Viewer Online Help* for more information.

2.8 FM Logging

The FM performs logging as other Intel® Fabric Set Component Family with Infiniband* Technology modules do. In the embedded environment, the user uses the logging controls to dictate where the log messages are sent (For example, internal buffer, external syslog server). In the host environment, the logs are sent to the system log on that system and may also be sent to a centralized syslog server.

Log messages follow the Intel® convention of severities, for instance, FATAL, ERROR, WARNING, INFO, and so on.

Many important fabric events are logged by the SM, such as ports going up or down, nodes not answering, nodes appearing or disappearing, SMs joining or leaving the fabric, SM state transitions, synchronization errors, and so on.

2.9 SM Algorithms & Features

2.9.1 Routing Algorithms in the SM

The primary routing algorithm selects, for every pair of nodes in the fabric, the quickest path available. In the case where there is more than one path available, load-balancing of the paths occurs over the available ports at each switch to get an even distribution of traffic over the fabric. Refer to Section 3.2.2, "Routing Algorithm" on page 30 for detailed information.

Spine-first routing is an optional feature that, when enabled, will favor paths that route through a chassis, rather than a path that exits and then re-enters a chassis. This helps eliminate cycles (for example, credit loops) on fabrics where such routes are possible. Refer to "Shortest Path" on page 31 for more information.

2.9.2 LID Mask Control (LMC)

The LMC feature provides a method for assigning multiple addresses, local identifiers (LIDs) to a single physical port. This allows for multiple paths through the fabric to be configured between a single pair of nodes. When enabled, the routing algorithm will attempt to ensure that these paths are placed on unique hardware when possible in order to reduce disruption in the case of switch failure. Each end-node port will have 2^{LMC} address LIDs. Refer to Section 3.2.4, "LMC, Dispersive Routing and Fabric Resiliency" on page 35 for detailed information.

2.9.3 Multicast Group Support

Multicast groups are used to direct one-to-many and many-to-many traffic. Nodes subscribe to multicast groups by issuing requests to the SM. The user may use SM CLI commands to manage multicast groups. Refer to Section 3.4, "Fabric Multicast Routing" on page 41 for detailed information.

2.10 FM's Subnet Manager

The FM implemented a complete InfiniBand* Technology-compliant Subnet Manager (SM). The SM is responsible for monitoring, initializing and configuring the fabric.

One of the critical roles of the SM is the initialization and configuration of routing tables in all the switches. The FMs SM supports a variety of routing algorithms which will be discussed in detail later in this guide. Among the capabilities are:

- Support for a wide range of fabric topologies, including Fat Tree, Clos network, Mesh/Torus and various irregular topologies
- Support for assigning multiple LIDs to end nodes and the carefully balanced programming of alternate routes through the fabric for use by dispersive routing, load balancing and failover techniques by various upper level protocols (ULP).
- Support for InfiniBand* Technology-compliant multicast, including MLID sharing, pre-creating groups, and other fine tuning of multicast performance
- Advanced monitoring and logging capabilities to quickly react to fabric changes and produce historical logs of changes and problems in the fabric
- Support for Virtual Fabrics with both QoS and partitioning capabilities
- Support for configuring and enabling adaptive routing in Intel® 12000 series switches

2.11 FM Interoperability

The FM supports many InfiniBand* Technology-compliant products in a fabric, and may be used in a fabric that is running any of the following:

- Intel® core switch chassis
- Intel® edge switches, both internally and externally managed
- Intel® HCAs
- Other InfiniBand* Technology-compliant HCAs and switches
- FastFabric
- Intel® OFED+ with its assorted Upper Level Protocols (ULPs) and applications
- 3rd Party Message Passing Interface (MPI) applications and middleware

The interoperability is seamless and requires no special configuration.

2.12 Terminology Clarification – “FM” vs. “SM”

In written documentation and verbal communication, there often arises confusion between the terms “FM” and “SM”. In some cases they may seem synonymous, while in others they denote distinct differences. Here are some guidelines for the usage of the terms:

- In general, SM refers specifically to the Subnet Manager, while FM refers to the True Scale Fabric Suite Fabric Manager, consisting of SM/SA, PM/PA, BM, and usually FE.
- When speaking of the product or set of running software processes, SM and FM may be interchangeable, due to the fact that the software is bundled and installed/operated together. Therefore, a phrase such as “start the SM” means to start all of the FM processes.

The scope of the terms are generally interpreted by the context in which it is being used. Therefore, if the specific managers are being discussed, then “FM” means the suite. Likewise, if general fabric management is being considered, then “SM” is being used to specify all of the managers.

§ §

3.0 Advanced FM Capabilities

3.1 Fabric Change Detection

An important aspect of the True Scale Fabric Suite Fabric Manager's (FM) Subnet Manager (SM) is its ability to rapidly detect and respond to fabric changes. This is accomplished using the Fabric Sweep process. The SM can start a fabric sweep due to any of the following:

- Fabric Change Trap from an HCA, a Target Channel Adapter (TCA) or a Switch. Typically it would be a port going up or down due to reboot, cable insertion/removal, and so on.
- Slow periodic sweep (`MaxSweepInterval`).
- Initial FM startup when Master
- Manually requested by user (`/opt/ifs_fm/bin/fm_cmd smForceSweep`)

The primary mechanism for detecting changes is traps. Traps are asynchronous notifications sent to the SM by devices (typically switches) when ports go up/down, when capabilities change, or links are unstable. The slow periodic sweep is used as a safety net in case a trap is lost or a device does not issue traps. All True Scale Fabric switches sold by Intel® support traps.

In extremely rare situations, the user may choose to force a sweep. This is generally not necessary when using fabrics constructed using Intel® supplied switches.

3.1.1 Handling Unstable Fabrics

During each sweep the SM analyzes the fabric and identifies what if anything has changed since the last sweep. If changes are detected, the SM will recompute the routes for the fabric and reprogram the devices in the fabric as needed.

During the analysis and reprogramming process the fabric may still be changing. In this case errors may occur. When more than the set `SweepErrorsThreshold` parameter of non-recoverable errors (such as a device going offline mid sweep) occur during a single sweep, the SM will abandon the sweep and start it over, to obtain a complete and accurate view of the fabric. The SM will not abandon the sweep more than the set `SweepAbandonThreshold` parameter of consecutive times, after which it will do the best it can. This helps to handle a fabric which is constantly changing, such as a fabric with an unstable link.

Similarly, if a port issues more than the `TrapThreshold` number of changes per minute, the SM considers the link unstable and will disable the port, removing it from the fabric, preventing any traffic to be routed over it.

Additionally the `Suppress1x` configuration parameter allows the SM to disable 1x links, removing low speed and low quality links from the fabric. All hardware available from Intel® is capable of 4x or more. 1x links typically represent devices with partially bad cables or hardware.

3.1.2 Tolerance of Slow Nodes

In rare cases, nodes under a heavy load, such as when running high stress MPI applications, will be slow to respond to SM sweeps. To avoid disrupting application runs, the SM can be configured using `NonRespTimeout` and `NonRespMaxCount` to be more tolerant of such devices and assume their capabilities have not changed since the last successful sweep.

The trade-off in increasing the tolerance is that loss of nodes will be detected much slower. Typically this capability is only relevant to HCAs. The risk in this feature mainly applies to nodes which hang but keep their link up so that the neighbor switch does not report a port state change.

Note: When using OFED, it is recommended to set `RENICE_IB_MAD=yes`, this will ensure rapid responses by the SMA and actually reduce overhead by avoiding the cost of retries. This option is enabled by default when using Intel® OFED+

3.1.3 Multicast Denial of Service

In rare cases, nodes send excessive multicast creates/deletes several times a second for the same group causing continuous SM sweeps. To stop the continuous SM sweeps, Multicast (MC) Denial of Service (DOS) can be set up in the configuration file to monitor the MC DOS Threshold, set up the interval of monitoring, and either bounce the port or disable the port.

3.2 Fabric Unicast Routing

One of the most complex aspects of configuring a large fabric is routing. The SM must configure a routing for the fabric which provides a careful balance between:

- Performance
- Resiliency to fabric disruptions
- Avoidance of deadlocks and credit loops
- Conservations of resources, such as LIDs and routing table entries

As a result of these sometimes conflicting goals, the SM allows for user configuration of many aspects of routing so the administrator can select the criteria for routing the fabric.

3.2.1 Credit Loops

Since the InfiniBand* Architecture has credit based link layer flow control, credit loops are possible. Under high stress, a credit loop can become a fabric deadlock which will force switch timers to discard packets. These deadlocks and discards can cause significant performance impacts.

The deadlocks are very rare and in practice, they only occur under high bandwidth applications, however it is better to route the fabric to avoid credit loops altogether.

There are many research papers on the topics of routing. Credit loop avoidance is a focus of many of the algorithms. Credit loops are avoidable for all the popular fabric topologies and the SM utilizes algorithms that are designed to avoid credit loops.

3.2.2 Routing Algorithm

The SM supports the following routing algorithms:

- shortest path
- fattree — optimized balanced routing for fat tree topologies
- Dimension Ordered Routing — Up/Down (dor-updown)

The routing algorithm is selectable using the `RoutingAlgorithm` parameter.

3.2.2.1 Shortest Path

This algorithm is the default and works very well for most fabrics. This algorithm always routes using a least cost path. In most fabrics there are many equal cost paths, in which case the SM will statically balance the number of paths using each Inter-Switch Link (ISL).

The Shortest Path algorithm has one option: `SpineFirstRouting`. When enabled, this will avoid credit loops in complex full bisectional bandwidth (FBB)—like fabrics which use Intel® 12000 Series modular switches. Given equal cost routes, SpineFirstRouting routes through chassis spine first. This avoids credit loops caused by routing using edge/leaf switches instead of spines.

SpineFirstRouting is enabled by default and has no ill side effects. Unlike simpler algorithms in other SMs, the Intel® SM's shortest path algorithm has sophisticated traffic balancing and routing algorithms which allow it to provide high performance for a wide variety of topologies

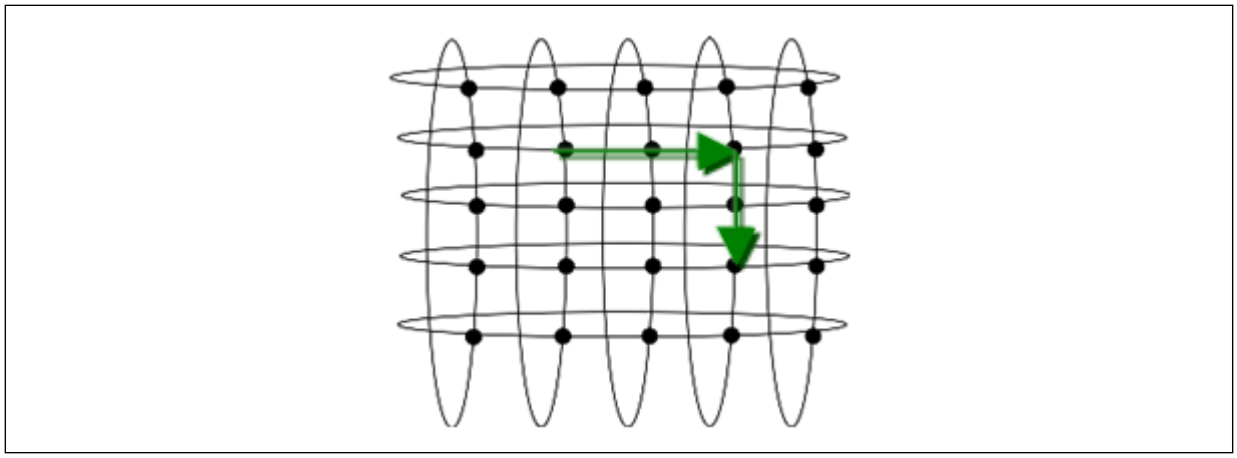
3.2.2.2 Fat Tree

The Fat Tree algorithm can provide better balancing of ISL traffic than the shortestpath algorithm. It accomplishes the balancing of ISL traffic through identification of the topology and figuring out what tier in the fabric each switch is at. To accomplish figure out the switch tier it needs to understand how many tiers of switch chips there are in the fabric and whether all HCAs and TCAs are at the same tier or on varied tiers of the fabric. Large switches typically have multiple internal tiers of switch chips. For example, all the Intel® 12800 series models have two tiers of switch chips.

3.2.2.3 Dimension Ordered Routing — Up/Down (dor-updown)

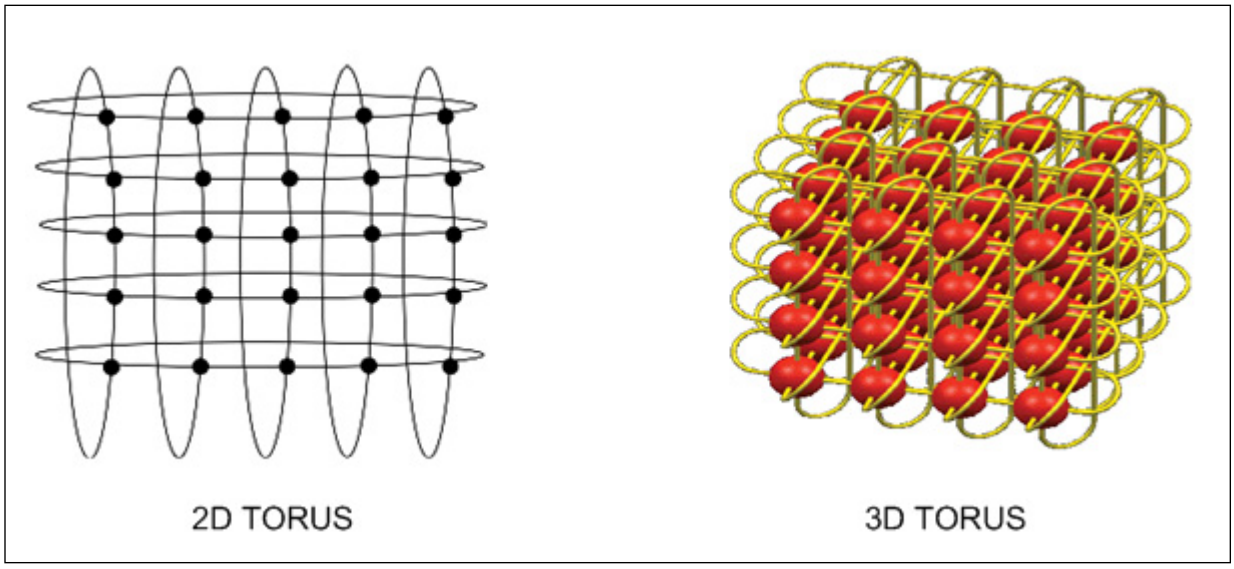
This routing algorithm is only for mesh and torus topologies. It provides shortest path routing while avoiding credit loops. Dimension Ordered Routing (DOR) refers to a specific method of routing a mesh or torus fabric that avoids credit loops while preserving shortest paths. Refer to Figure 3

Figure 3. Dimension Ordered Routing



In the context of DOR, mesh refers to an N-dimensional grid of switches, where each node is linked along each dimension to a neighbor in both the forward and reverse direction. One can easily imagine 2D planes where each switch is linked to 4 neighbor switches, and 3D cubes where each switch is linked to 6 neighbor switches. End nodes are evenly distributed over the switches. Refer to Figure 4

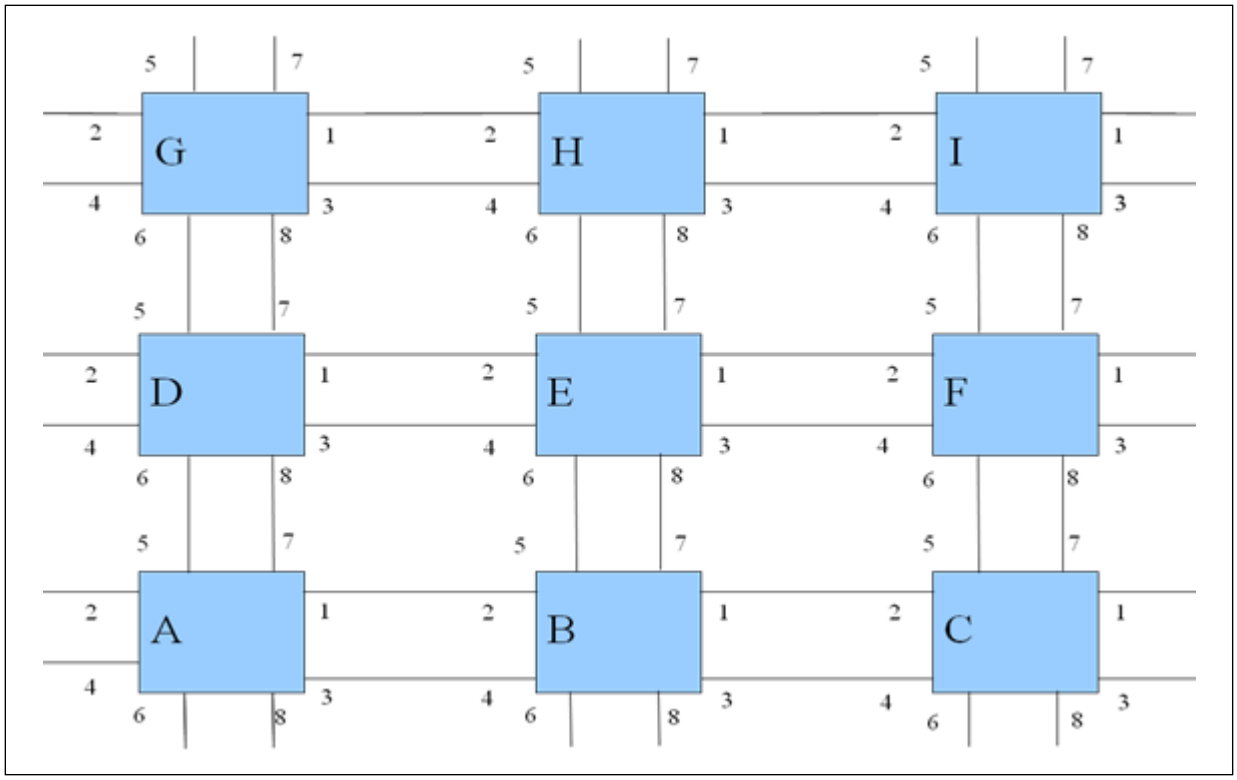
Figure 4. 2D and 3D Torus Structures



A torus fabric is structurally identical to a mesh, with the exception that the switches located along the edges are "wrapped around" and connected together, such that the fabric is seamless.

For this algorithm the fabric must be uniformly cabled such that a given port number on every switch is connected in the same “direction”. For example in a 2D mesh it may be that port 1,3 on every switch is connected to port 2,4 of the next switch to form the X dimension and port 5,7 connected to port 6,8 of another switch to form the Y dimension. Refer to Figure 5 for an example.

Figure 5. Sample Cabling for Part of a 2D Mesh with 2 ISLs in Each Direction



When routing a torus fabric, DOR uses 2 Virtual Lane (VL)s on ISLs (to avoid credit loops) and uses $2^{dimensions}$ Service Level (SL)s.

The FM's SM implements an augmented DOR algorithm which also handles fabric disruptions for mesh and torus topologies. To handle fabric disruptions, an additional VL and SL is used for an Up/Down routing algorithm which can route around the failure. Such routing around failures may be non-optimal, but it will be credit loop free. The Up-Down routing algorithm also uses an additional LID per Channel Adapter, therefore for dor-updown the LMC of the Channel Adapters will be 1 (2 LIDS per Channel Adapter) or greater. A Mesh requires a minimum of 2 VLs per ISL and a Torus requires a minimum of 3 VLs per ISL. Both a Mesh and a Torus require only 1 VL for HCA to Switch links. Refer to Appendix D, "QOS Options in a Mesh/Torus vFabric" for a table summarizing the various combinations of Mesh/Torus and vFabric/QoS along with the SL/VL requirements for each combination.

Since the InfiniBand* Architecture Standard is limited to 16 SLs, there is a limit of 3 dimensions for Torus fabrics with disruption handling (for a total of 9 SLs). Mesh fabrics do not have this limitation.

Note: Since a Mesh/Torus makes use of multiple SLs and LIDs, high performance applications must query the SA for a PathRecord for every Source/Dest pair. The SL will vary per PathRecord and is computed by the SM Based on the present topology of the fabric. To allow interoperability with non-ideal application implementations (such as verbs based MPIs which do not interact with the SA), the Intel® FM will configure the 1st SL and the 1st LID on each HCA for the Up/Down route. This permits such applications to run, albeit with potentially increased latency and congestion.

See Section 3.3, "Mesh/Torus Topology Support" on page 36 for more information about DOR-Up/Down routing.

3.2.3 Adaptive Routing

A serious limitation of static routing is that it must be done before traffic begins to flow, hence static routes are balanced using "best guesses" by the FM of potential application traffic patterns. However, once applications start to run, those routes may be non-ideal. Adaptive routing is a powerful capability of all Intel® 12000 series switches, which allows the switches to scalably adjust their routes while the applications are running to balance the routes based on actual traffic patterns.

The Intel® adaptive routing solution is highly scalable because it allows the FM to provide the topology awareness and program the switches with the rules for adaptive routing. Then the FM gets out of the way and permits the switches to dynamically and rapidly adjust the routes based on actual traffic patterns.

This approach ensures a scalable solution since as switches are added, each new switch will work in parallel with others to dynamically adapt to traffic. This approach avoids the FM becoming a bottleneck and scales very well.

Adaptive routing provides a few important capabilities and options:

1. Adaptive routing can rapidly route around fabric disruptions and lost ISLs. When adaptive routing is enabled, this capability will automatically occur and limit the amount of lag time between an ISL going down and the traffic being redirected to alternate routes.
2. Adaptive routing can automatically balance and rebalance the fabric routes based on traffic patterns. It has the unique ability to handle changing traffic patterns which may occur due to different computational phases or the impacts of starting or completing multiple applications which are running on the same fabric.
3. For Fat Tree topologies, an additional level of adaptive routing can be enabled (`Tier1FatTree`) which permits even more flexible adaptive routing and can better balance traffic among multiple core switches.

The configuration allows adaptive routing to be enabled for lost routes and congestion, for lost routes only, and optionally support adaptive routing across multiple switches in Fat Tree topologies (both FBB or oversubscribed fat trees).

The lost route parameter provides alternative routes to the switch firmware. When routes are lost, the switch firmware has immediate access to an alternate route instead of waiting for the FM to reprogram the switch tables. If this parameter is not set, the alternate routes are also used to redirect traffic during periods of congestion.

For pure fat trees topologies, the switch has more options for traffic redirection. If the `Tier1FatTree` parameter is enabled, tier1 routing is also enabled. When `Tier1FatTree` parameter is enabled, it enables adaptive routing across different spines. When `Tier1FatTree` parameter is disabled, adoption only occurs among ISLs connected between pairs of switches. `Tier1FatTree` parameter is ignored when DOR/UpDown routing is selected for Mesh/Torus.

The following is the adaptive routing section of the `ifs_fm.xml-sample` file:

```
<!-- Configures support for AdaptiveRouting in Intel QDR Switches -->

    <AdaptiveRouting>

      <!-- 1 = Enable, 0 = Disable -->

      <Enable>0</Enable>
```

```
        <!-- When set, only adjust routes when they are lost. -->
        <!-- If not set, adjust routes when they are lost and -->
        <!-- when congestion is indicated. -->
        <LostRouteOnly>0</LostRouteOnly>
        <!-- The topology is a pure fat tree.  Do maximum amount of -->
        <!-- adaptive routing based on this topology. -->
        <Tier1FatTree>0</Tier1FatTree>
    </AdaptiveRouting>
```

3.2.4 LMC, Dispersive Routing and Fabric Resiliency

The SM also supports LID Mask Control (LMC). LMC allows for more than 1 LID to be assigned to each end node port in the fabric, specifically 2^{LMC} LIDs will be assigned. This allows the SM to configure the fabric with multiple routes between each end node port, allowing applications to load balance traffic (for example, using algorithms such as PSM's dispersive routing) across multiple routes or provide rapid failover using techniques like Alternate Path Migration (APM).

The Intel® Performance Scaled Messaging (PSM) layer can take advantage of LMC to provide dispersive routing and load balance MPI across multiple routes. See the *Intel® True Scale Fabric OFED+ Host Software User Guide* for more information about PSM's dispersive routing capabilities.

When LMC is configured with a non-zero value, the SM assigns routes with the following goals in priority order:

1. Each LID for a given destination port is given as unique a route as possible through the fabric, using completely different switches when possible or at least different ports in the same switch. This approach provides optimal resiliency so that fabric disruptions can be recovered from using APM and other rapid failover techniques.
2. The overall assignment of Base LIDs to ISLs is statically balanced, such that applications which only use the Base LID will see balanced use of the fabric
3. The assignment of LIDs to ISLs is statically balanced, such that applications which use multiple LIDs for load balancing will see additional available bandwidth.

3.2.4.1 PathRecord Path Selection

When a non-zero LMC value is used, the SM will have multiple paths available between pairs of nodes. The FM permits configuration of the SM to specify which combinations of paths should be returned and in what order. Most multi-path applications will use the paths in the order given, so the first few returned will typically be used for various failover and dispersive routing techniques.

Most applications will use the first path or only the first few paths. When LMC!=0, there can be N=(1<<LMC) addresses per port. This means there are N^2 possible combinations of SLID and DLID which the SA could return in the Path Records. However there are really only N combinations which represent distinct outbound and return paths. All other combinations are different mixtures of those N outbound and N return paths.

Also important to note is that LMC for all Channel Adapters are typically the same, while LMC for switches will be less. Generally redundant paths and/or having a variety of paths is not critical for paths to switches (which are mainly used for management traffic), but can be important for applications talking Channel Adapters to Channel Adapters.

The FM `PathRecordSelection` parameter controls what combinations are returned and in what order. For examples below lets assume SGID LMC=1 (2 LIDs) and DGID LMC=2 (4 LIDs)

- `Minimal` – return no more than 1 path per lid: SLID1/DLID1, SLID2/DLID2 (since SGID has 2 lids stop)
- `Pairwise` – cover every lid on both sides at least once: SLID1/DLID1, SLID2/DLID2, SLID1/DLID3, SLID2/DLID4
- `OrderAll` – cover every combination, but start with pairwise set: SLID1/DLID1, SLID2/DLID2, SLID1/DLID3, SLID2/DLID4 SLID1/DLID2, SLID1/DLID4, SLID2/DLID1, SLID2/DLID3
- `SrcDstAll` – cover every combination with simple all src, all dst: SLID1/DLID1, SLID1/DLID2, SLID1/DLID3, SLID1/DLID4 SLID2/DLID1, SLID2/DLID2, SLID2/DLID3, SLID2/DLID4

3.3 Mesh/Torus Topology Support

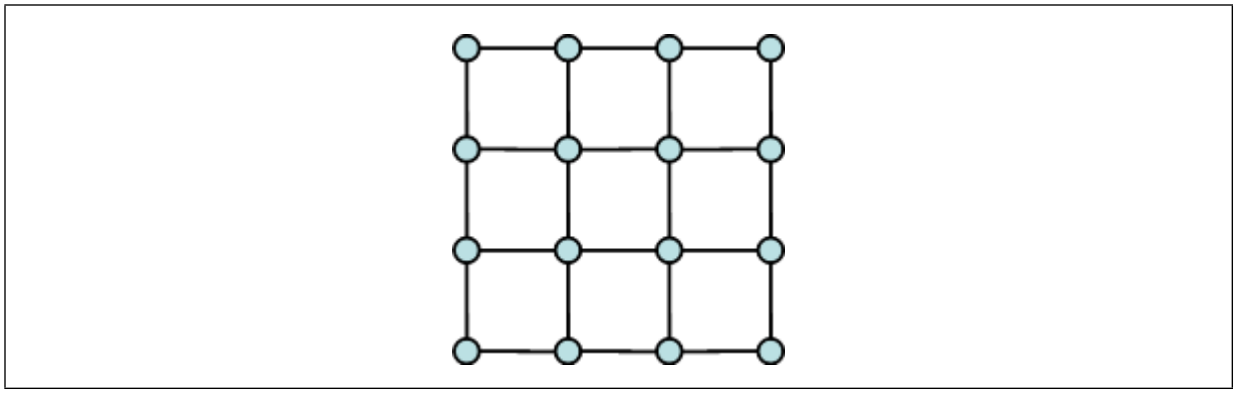
Mesh/Torus fabrics are a popular option when building large fabrics. This option can trade-off interconnect cost with performance. By their nature, Mesh/Torus fabrics are highly oversubscribed and can have significant differences in latency depending on which node pairs are communicating. Applications must be carefully placed and tuned to obtain optimum performance.

In contrast Fat-Tree networks offer much higher bisectional bandwidth and much more consistent node-to-node latency. When budget permits, a Fat-Tree is often recommended.

Mesh/Torus fabrics consist of a grid of edge-switches interconnected in a uniform pattern. Mesh/Torus fabrics must be built from individual edge-switches (such as 12200 or 12300 models or comparable blade switches). Mesh/Torus fabrics may not be built using larger multi-slot chassis such as the 12800 models.

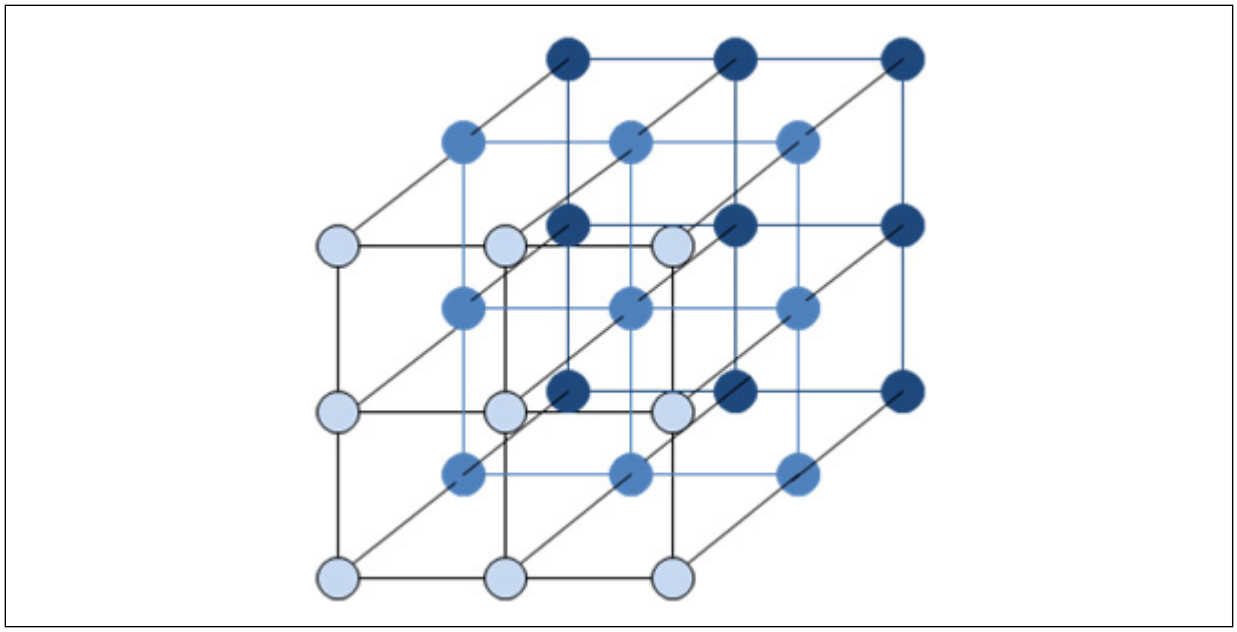
For example a two-dimensional 4x4 mesh can appear as shown in Figure 6.

Figure 6. 2D 4x4 Mesh Fabric Example



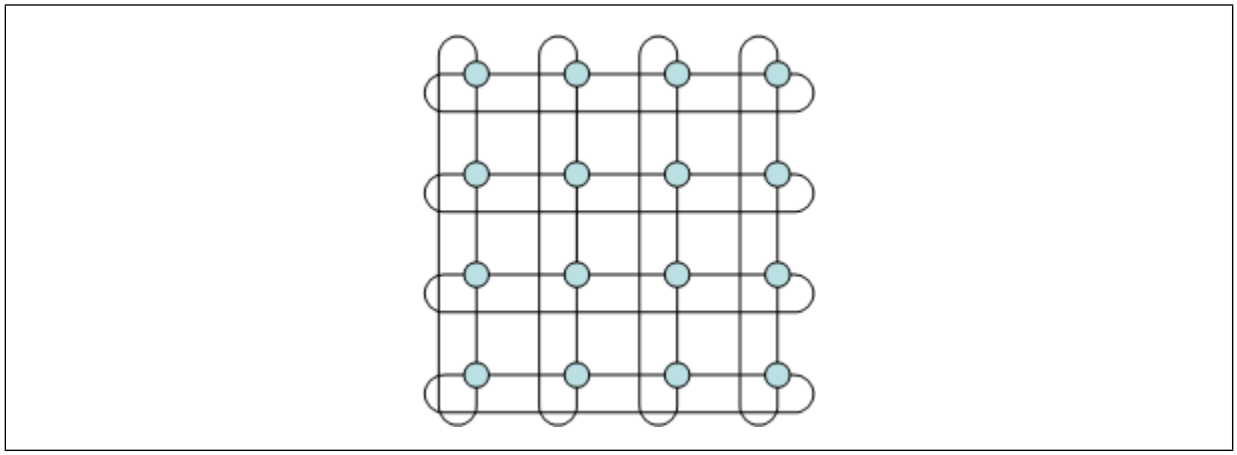
More dimensions can decrease node-to-node latency but often reduce node-to-node bandwidth and require more complex cable routing. For example a three-dimensional 3x3x3 mesh can appear as as shown in Figure 7.

Figure 7. 3D 3x3x3 Mesh Fabric Example



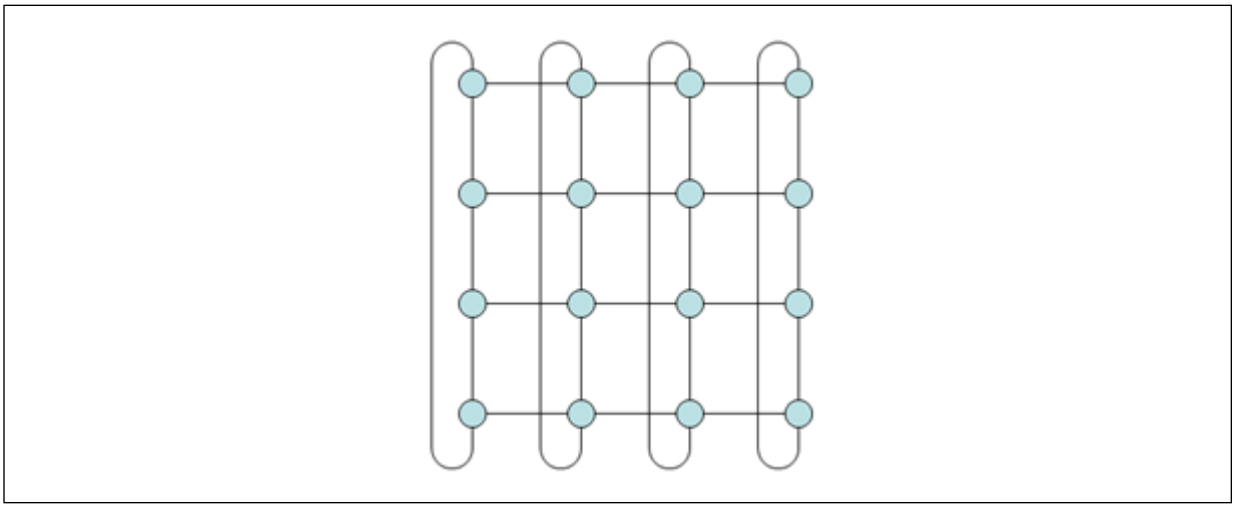
Another optimization for fabric latency is to add "wrap-around" or toriodal links. These links connect the "edges" of the mesh to provide a shorter path when nodes near one edge want to communicate to nodes near the other edge. For example a two-dimensional 4x4 torus can appear as shown in Figure 8

Figure 8. 2D 4x4 Torus Fabric Example



The FM allows the use of torus links to be configured per dimension. Some non-toroidal dimensions can simplify cabling or allow for greater flexibility in terms of number of QoS levels or number of dimensions. For example a two-dimensional 4x4 fabric with 1 Toroidal and 1 non-Toroidal dimension can appear as shown in Figure 9

Figure 9. 2D Mesh with 1 Toroidal Dimension Fabric Example



3.3.1 Disruption Handling

A very important part of any routing algorithm is its ability to handle fabric disruptions. In High Performance Computing (HPC), fabrics design changes are rare, however fabric disruptions are quite common. Disruptions can be the result of simple maintenance operations, such as replacing a cable or rebooting a switch. Disruptions can also be the result of more wide-spread events, such as loss-of-power for a significant part of the fabric.

Unlike Fat Tree topologies, Mesh fabrics have the potential for numerous credit loops. It is important that the routing algorithm ensure the fabric routing is deadlock free while still providing optimal latency. Also the routing algorithm must be able to properly route the fabric within the constraints of the hardware (number of VLs/SLs) while handling a potentially large number of fabric disruptions.

The FM's support for dor-updown provides an optimized solution to this complex challenge. It can optimally route the fabric deadlock free, and it can handle a large number of disruptions, all while staying within the constraints of existing hardware. As such it is an ideal solution and provides better operation of the fabric than some of the alternatives (For example, Lash, Lash-Tor, Segment Based Routing, and so on), all of which either compromise performance or are unable to handle more than a few disruptions within the capabilities of the hardware.

The updown spanning tree and the multicast spanning tree match so there will not be conflicts between the updown unicast traffic and the multicast traffic. Matching the two spanning trees also prevents credit loops and prevents failover speedups.

3.3.2 Path Record Query

As has been implied by the previous discussion, the dor-updown algorithm uses multiple LIDs, SLs and VLs. For applications to use the correct route through the fabric, they must use SA PathRecord queries to obtain the addressing information for communicating to another Channel Adapter. Unlike simpler algorithms, "cheating the standard" and bypassing the SM will result in non-optimal performance. Techniques such as out of band LID exchange (which is used by many MPI implementations) will provide sub-optimal performance.

To permit non-InfiniBand* Technology-compliant applications (such as the existing MVAPICH and OpenMPI implementations for verbs) to function in a Mesh/Torus fabric, the FM configures the Base LID and the 1st SL on each Channel Adapter for the Up/Down route. This route will provide reliable deadlock free operation, even if Channel Adapters simply exchange LIDs. It will also operate both for complete and disrupted fabrics. However, this route will provide greatly increased latency and reduced bandwidth as compared to proper use of the SM/SA.

Many applications use IPoIB for path resolution. Since IPoIB makes PathRecord queries, such applications will be given optimized routes and will function properly.

To permit optimized MPI performance for Mesh/Torus fabrics, the Intel® HCA with its Performance Scaled Messaging (PSM) technology should be used. PSM can perform PathRecord queries when its path_query option is enabled (see *Intel® True Scale Fabric OFED+ Host Software User Guide*).

To ensure scalability when using Intel® PSM with PathRecord queries enabled, the Distributed SA (dist_sa) must be enabled on every compute node. The Distributed SA synchronizes the node relevant PathRecord information with each end node such that job startup time is optimized. See the *Intel® True Scale Fabric OFED+ Host Software User Guide* for more information on the Distributed SA.

3.3.3 Virtual Fabrics

The FM permits use of Virtual Fabrics in a Mesh/Torus fabric. In such environments QoS and/or Security can be enabled to separate various applications or nodes. The QoS Options table in Appendix D, "QOS Options in a Mesh/Torus vFabric" shows some of the combinations of QoS options which are possible. For more information on Virtual Fabrics refer to Chapter 5.0, "Virtual Fabrics".

3.3.4 Dispersive and Multi-Path Routing

Mesh/Torus fabrics require at least 2 LIDs per Channel Adapter. The LMC is set to 1 by default. One LID is used for the optimized DOR route and one LID is used for the fall back Up/Down route

Often it is desirable to use advanced features such as dispersive routing in MPIs using Intel® PSM or other applications which may be able to take advantage of multiple paths for redundancy, load balancing or performance.

To permit this, the FM allows LMC>1 to be configured for Mesh/Torus fabrics. In which case one LID is used for the Up/Down route and all the remaining routes are used for DOR routes. So for example an LMC of two will cause four LIDs per Channel Adapter. One will be for Up/Down and the remaining three will be for optimized DOR routes. When the SM is queried for a PathRecord it will report only the three DOR LIDs for non-disrupted paths and only the one Up/Down LID for disrupted paths.

All alternate DOR routes will be optimized per the LMC and fabric resiliency section above.

Note: In order to take advantage of multi-path routing in a mesh/torus, there must be more than one ISL between switches per dimension. In order to avoid credit loops, all multi-path routes configured by the FM will obey the DOR routing rules. Use of distinct switch chassis is not available for DOR alternate routes.

3.3.5 Switch and HCA VLs

When operating a Mesh/Torus fabric, more than one VL is always required. Many devices ship by default with only one VL enabled. Prior to running with the dor-updown algorithm, the switches and HCAs must first be configured for the proper number of VLs and the appropriate MTU. Refer to Appendix D, "QOS Options in a Mesh/Torus vFabric" for more information on the number of VLs that are needed for various combinations.

3.3.6 Bootstraping the fabric

For fabrics with Intel® Externally Managed switches, it will be necessary to have the SM operational in order to configure the switches. Also other verification operations, such as topology verification may be needed prior to putting the fabric into normal operation.

Due to the stricter topology and VL requirements of the FM for Mesh/Torus fabrics, it can be helpful to pre-configure and verify this information prior to enabling the dor-updown routing algorithm in the FM.

It is important to understand that the FM's shortestpath algorithm can be used to temporarily route a Mesh/Torus for low bandwidth basic operations such as fabric topology verification and the configuration of externally managed switches. The shortestpath algorithm will be able to route the fabric in a usable manner for such operations, however it will have credit loops. Such credit loops will not cause issues for low bandwidth operations nor for operations where only two nodes in the fabric are using high bandwidth (such as loading firmware on a externally managed switch). However attempts to run real MPI jobs with multiple nodes or collectives or even FastFabric MPI Performance verification tools while using shortestpath are likely to induce credit loops and cause extremely poor performance.

3.3.7 Topology Configuration in the FM

When using the dor-updown algorithm, the Mesh/Torus topology information must be configured in the FM configuration file. Only the ISLs that constitute the different dimensions of the Mesh or Torus must be defined. All ISLs in the fabric which have not been configured in this section will be considered invalid and ignored when the dor-updown algorithm is selected as the routing algorithm.

The FM only needs the "ISL pattern" to be defined. It merely needs to know which pairs of ports will be connected to form each dimension. It does not need to know any specific details about the switches. Replacement of switches will not require any configuration changes.

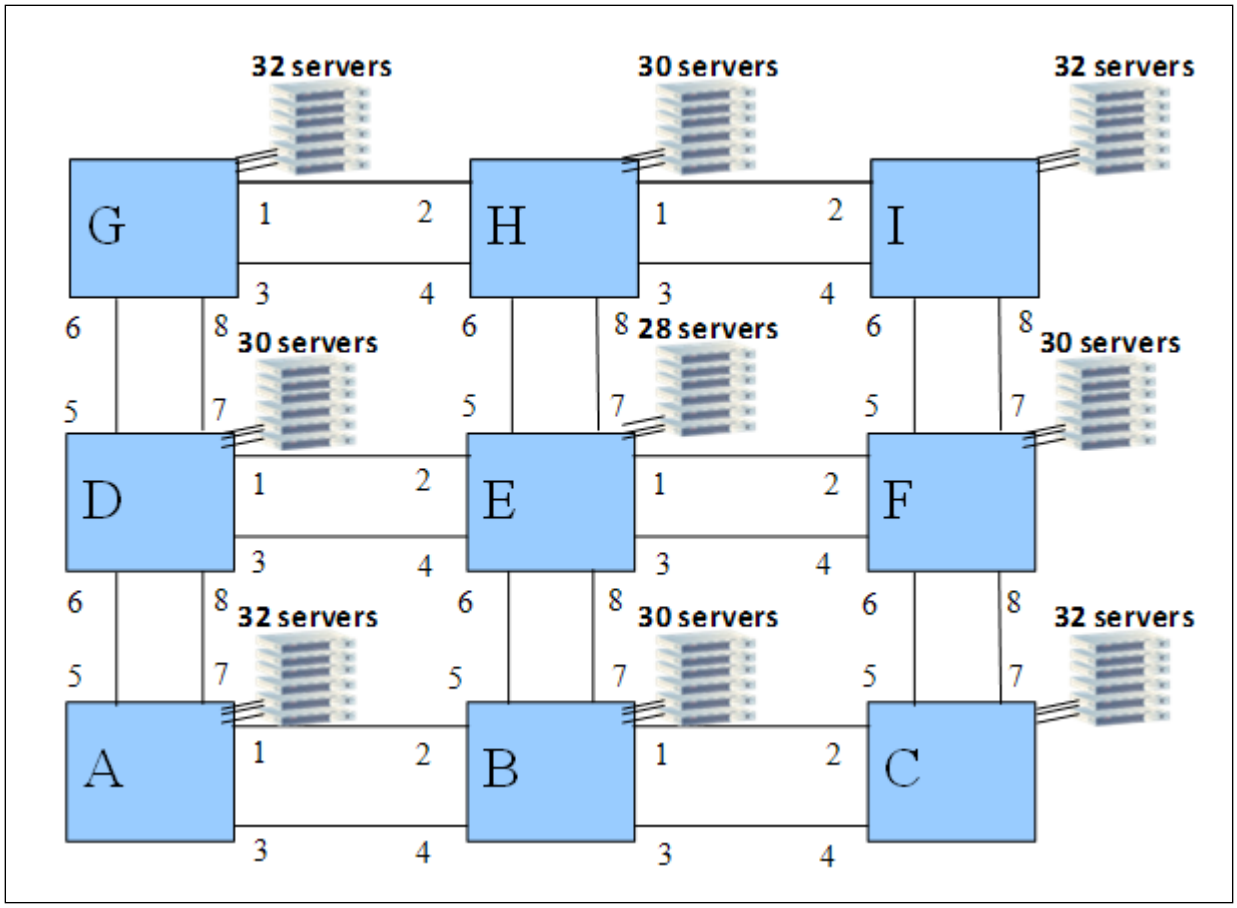
This is necessary so that the FM can be aware of the intended topology and therefore properly configure SLs and VLs. This approach allows the FM to minimize impacts when large fabric changes occur, such as the loss/reappearance of toroidal links or loss/reappearance of an entire dimension.

The SM will ignore the invalid links and proceed ahead with the sweep and will make the best case effort to finish the sweep, so that other tools (like FastFabric) can be used to verify the condition of the fabric. If there are warnings related to invalid links, it is highly recommended to verify the cable connections in the fabric. At the end of the sweep, the SM will show all the invalid ISL port pairs it found during the sweep.

In Mesh fabrics, the FM will allow Channel Adapters to be connected to unused switch ports at the edges of each mesh dimension. For example, in Figure 10, when using 36 port switches. There could be additional servers on edge and corner switches, such as:

- Switch E: 32 servers
- Switch B, D, F, H: 33 servers
- Switch A, C, G, I: 34 servers

Figure 10. Example of a 3X3 Mesh with extra Channel Adapters on Perimeter Switches



3.4 Fabric Multicast Routing

In addition to unicast, True Scale Fabric also supports multicast. Multicast allows a single packet sent by an application to be delivered to many recipients. True Scale FAbric Multicast is used for IPoIB broadcast for TCP/IP address resolution protocol (ARP), IPoIB multicast for UDP/IP multicast applications, and can also be directly used by other applications.

True Scale Fabric supports separate routes per Multicast Group. Each multicast group is identified by a unique 128-bit Multicast GID. Within the fabric, the SM assigns each active multicast group a 16-bit Multicast LID. The InfiniBand* Architecture Standard allows for up to 16K unique Multicast LIDs in a fabric.

To implement multicast, the SM must construct spanning tree routes throughout the fabric which will deliver exactly one copy of each sent packet to every interested node. The SM must configure such routes within the limitations of the hardware. Namely:

- Most switch chips have a limitation of only 1024 multicast LIDs
- There may be varied MTU and speed capabilities for different switches and Inter Switch Links.
- The fabric topology and hardware may change after applications have joined multicast group

To support efficient yet dependable routing of multicast, the SM allows the user to configure and control Multicast Routing.

3.4.1 Handling Fabric Changes

The InfiniBand* Architecture Standard has a limitation in that the SM must make the realizable decision for a multicast group at the time an application creates/joins a multicast group. This means the SM must determine if there is a path with the appropriate speed and MTU to meet the requested capabilities of the multicast group.

However later fabric changes could make the multicast group unrealizable. For example, removal or downgrade of high speed links, loss of switches, changes to switch MTU or speed configuration, to name a few. Unfortunately, in this case there is no standard way to notify end nodes that the multicast group is no longer viable.

To address this situation, the SM performs stricter multicast checking at Join/Create time. This means a multicast join/create is rejected if there are any switch-to-switch links which do not have at least the MTU or rate requested for the multicast group, reducing the chance that a simple fabric failure or change (for example, loss one-of-one link) could make the group unrealizable.

The `DisableStrictCheck` parameter controls this capability. When 1, this parameter disables the strict checking and accepts Join/Create request for which at least one viable fabric path exists. By default the parameter is 0, which allows for more strict checking.

In addition, the `MLIDTableCap` parameter is used to configure the maximum number of Multicast LIDs available in the fabric. This must be set to a value less than or equal to the smallest Multicast forwarding table size of all the switches which may be the in the fabric. It defaults to 1024, which is the typical capability of switches.

3.4.2 Conserving Multicast LIDs

Most switches with InfiniBand* Technology, have a limitation of 1024 multicast LIDs (MLIDs). However IPv6 (and possibly other applications) can create numerous multicast groups. In the case of IPv6, there is one Solicited-Node multicast group per HCA/TCA port. This results in an excessively large number of multicast groups. Also in large fabrics, this quickly exceeds `MLIDTableCap`. For example, a 2000 node fabric with IPv6 would need over 2000 multicast groups.

To address this situation, the SM can share a single MLID among multiple Multicast groups. Such sharing means both the routes, and destinations will be shared. This may deliver some unrequested multicast packets to end nodes, however unneeded packets will be silently discarded by the InfiniBand* Architecture transport layer in the HCA/TCA and will have no impact on applications.

The SM allows the administrator to configure sets of multicast groups which will share a given pool of Multicast LIDs. This is accomplished using the MLIDShare sections in the configuration file.

MLID sharing can conserve the hardware MLID tables so other uses of multicast can be optimized/efficient.

By default the SM will share a pool of 500 LIDs among all IPv6 solicited-node multicast groups. Thus in fabrics of 500 nodes or less, a unique LID will be used for every multicast group. However in larger fabrics, LIDs will be shared so that there are still over 500 unique LIDs available for other multicast groups, such as the IPoIB broadcast group and other multicast groups which may be used by applications.

3.4.3 Precreated Multicast Groups

The first end node which joins a multicast group will also create the multicast group. When a multicast group is created, critical parameters such as the MTU and speed of the multicast group are also established. The selection of these values must carefully balance the performance of the multicast group against the capabilities of the hardware which may need to participate in the group in the future. For example if an application on a DDR HCA with a 4K MTU creates a 4K DDR multicast group, it will prevent subsequent joins of the group by SDR or 2K MTU HCAs.

Some ULPs and applications, such as IPoIB, require key multicast groups, such as the IPv4 broadcast group, to be pre-created by the SM.

Pre-created multicast group configurations are specified in the `MulticastGroup` sections of the SM configuration files. When the multicast groups are pre-created, their MTU and speed are defined by the SM configuration file, allowing the administrator to be able to account for anticipated hardware capabilities and required performance.

3.4.4 Multicast Spanning Tree Root

Multicast routing is performed by computing a spanning tree for the fabric. The tree has a root switch and spans throughout the fabric to reach all of the switches and HCAs which are members of the multicast group.

The FM allows the root of the spanning tree to be configured using the `Sm.Multicast.RootSelectionAlgorithm` parameter.The SM's `MinCostImprovement` parameter can determine how much improvement is needed before a new spanning tree root is selected. Disruption to in-flight multicast traffic can be avoided or limited to cases where the fabric has changed significantly enough to provide sufficient benefit to justify a change by using these parameters.

The SM's DB Sync capability synchronizes the multicast root between the master and standby SMs. During SM failover the multicast root can be retained and limit disruption to multicast traffic in flight.

3.4.5 Multicast Spanning Tree Pruning

A complete tree will unconditionally include all switches. When HCAs request to join or leave the multicast group the SM only needs to program the switch immediately next to the HCA.

A pruned tree will omit switches which do not have HCAs as members of the group, as well as intermediate switches that do not need to be in the group. A pruned tree will reduce multicast traffic internal to the fabric when only a small subset of nodes are part of a given multicast group. The time to add or remove HCAs from the group can be significantly higher as many intermediate switches may need to also be programmed for the group.

The default is a complete tree. This has been found to work very well in HPC environments. Such environments typically have very little multicast traffic with the vast majority of traffic being IPoIB ARP packets which need to be broadcast to all nodes running IPoIB. The default allows IPoIB hosts to come up and down quicker.

3.5 Packet and Switch Timers

The InfiniBand* Architecture Standard allows for assorted timers and lifetimes to be set to avoid unforeseen situations that can cause packets to be lost or progress to be stalled causing widespread impacts from localized situations (For example, a hung server, broken ISL, and so on). The SM allows these timers to be configured by the adminstrator.

3.5.1 Switch Timers

Every switch supports the following standard timers:

- HeadOfQueueLife (HoqLife)
- SwitchLifeTime
- VLStallCount

These can be used to relieve fabric congestion and avoid fabric deadlocks by discarding packets. Discards help prevent back pressure from propagating deep into the core of the fabric, however such discards will cause end nodes to time-out and retransmit.

If a packet stays at the Head of a Switch Egress Port for more than HoqLife, it is discarded. Similarly a packet queued in a switch for more than SwitchLifetime is discarded. SwitchLifetime and HoqLife can also be set to infinite in which case no discards will occur.

VLStallCount controls a second tier more aggressive discard. If VLStallCount packets in a row are discarded due to HoqLife by a given VL on an egress port, that egress port's VL enters the VL Stalled State and discards all that VL's egress packets for 8*HoqLife.

Packets discarded for any of these reasons will be included in the TxDiscards counter for the Port, which can be queried using FastFabric. Such discards are also included in the Congestion information monitored by the PM and available using FastFabric tools such as `iba_top`, `iba_rfm` and `iba_paquery`. A congestion which is severe enough to cause packet discards is given a heavy weight so that it will not go unnoticed.

3.5.2 Packet LifeTime

Within an HCA/TCA every Reliable Queue Pair (QP) has a time-out configured. If there is no acknowledgment (ACK) for a transmitted QP packet within the time-out, the QP will retry the send. There is a limit on retries (up to 7) after which the QP will fail with a Retry Timeout Exceeded error.

The InfiniBand* Architecture Standard defines that the timeout for a QP should be computed based on the Packet LifeTime reported by the SA in a PathRecord. The LifeTime represents the one way transit time through the fabric. Therefore, the actual QP timeout will be at least 2x the Packet LifeTime (plus some overhead to allow for processing delays in the TCA/HCA at each end of the fabric).

Careful selection of Packet LifeTime (and QP time-outs) is important. If time-outs are set too large, then the impact of a lost packet could be significant. Conversely if the time-outs are set to low, then minor fabric delays could cause unnecessary retries and possibly even Retry Timeout Exceeded errors and the resulting disruption of applications.

The SM allows for two approaches to configure Packet LifeTime

- Constant
- Dynamic

Note: Some applications, especially MPI middleware, have independent configuration of QP time-outs and ignore the values provided by the SM. For such applications configuration of SM Packet LifeTime will have no effect.

The constant approach causes the SM to return the same value for Packet LifeTime for all queries.

The dynamic approach causes the SM to return a different value for the Packet LifeTime depending on the number of hops through the fabric in the given path. This allows the SM to account for the fact that longer routes have more opportunities for congestion, queuing delays, and so on and should have larger time-outs. When using the dynamic approach, a unique Packet LifeTime can be configured for one hop to nine hop paths. Paths greater than nine hops will all use the nine hop value.

3.6 Fabric Sweeping

The SM periodically sweeps the fabric. During a sweep the fabric is analyzed, routes are computed and switches and Channel Adapters are configured. The SM sweep algorithms attempt to carefully balance:

- Responsiveness to fabric changes
- Limit performance overhead of SM
- Efficiency of fabric analysis and configuration
- Potential hardware limitations
- Handle possibility that fabric is changing while being analyzed/configured

The SM performs sweeps at fixed intervals. The SM immediately performs a sweep when a switch reports a port state change trap. Such traps indicate a link has come up or down. Generally traps will trigger rapid sweeps to respond to fabric changes. The fixed sweeps are a safety net in case traps are lost or there are switches whose attempts to send traps are failing. To limit overhead, the SM fixed sweep is very slow (5 minutes).

During a sweep, the SM must query and configure many devices in the fabric. These operations occur using the SM protocol over VL15. VL15 is a non-flow-controlled VL. This combined with the fact that the fabric could be changing while the SM is sweeping, means that packets may be lost during the sweep.

To optimize the performance of the sweep, the SM can issue multiple concurrent packets to a given device or the fabric. The number to issue at once must be carefully balanced between the capabilities of the hardware and the goal of making the sweep faster.

The SM allows configuration of sweep characteristics using a number of parameters.

3.6.1 Optimized Fabric Programming

The InfiniBand* Architecture Standard allows the SM to route SMA packets using "Directed Routed" or "LID Routed" mechanisms. Packets can also be routed with a mixture of these two mechanisms. LID routed packets follow the normal routing used by other traffic and are fully routed by hardware with low latency. Directed Routed packets have the route explicitly specified in the packet, hop by hop, and allows the SM to access components in the fabric prior to having the Switch Routing tables fully

programmed. Directed route packets are typically routed in switch firmware and experience higher latency at each switch hop. Typically LID routed SM packets incur much lower latency while traversing the fabric.

To optimize fabric programming in large fabrics, the FM supports “Early LID Routing”. This allows the SM to program routing tables in a “cresting wave” approach such that the majority of fabric programming can use LID routed packets or packets which are LID routed all the way to their final hop.The net result of this advanced mechanism is fast programming of the fabric and therefore more rapid fabric initialization and change handling. This feature can be controlled using the `SmaEnableLRDR` and `EhcaSmaEnableLRDR` parameters.

3.6.2 Scalable SMA Retries

The *InfiniBand* Architecture Specification Release 1.2.1* defines that SMA packets are not flow controlled. However to optimize fabric programming time, the FM can be configured to issue multiple SMA packets in parallel. This approach can result in occasional packet loss which can have a negative effect on the SMs performance. To allow for rapid recovery from such packet loss, while not causing excessive retries to sluggish nodes, the FM allows for scalable retries with a increasing randomized backoff algorithm with each attempt.

3.7 Link Speed Negotiation

When running with some DDR or QDR devices, links may only come up at SDR speeds. To address this issue the FM supports a FM based ability to negotiate link speeds and permit DDR or QDR operation for such devices. This capability is enabled using the `LinkSpeedOverride` option.

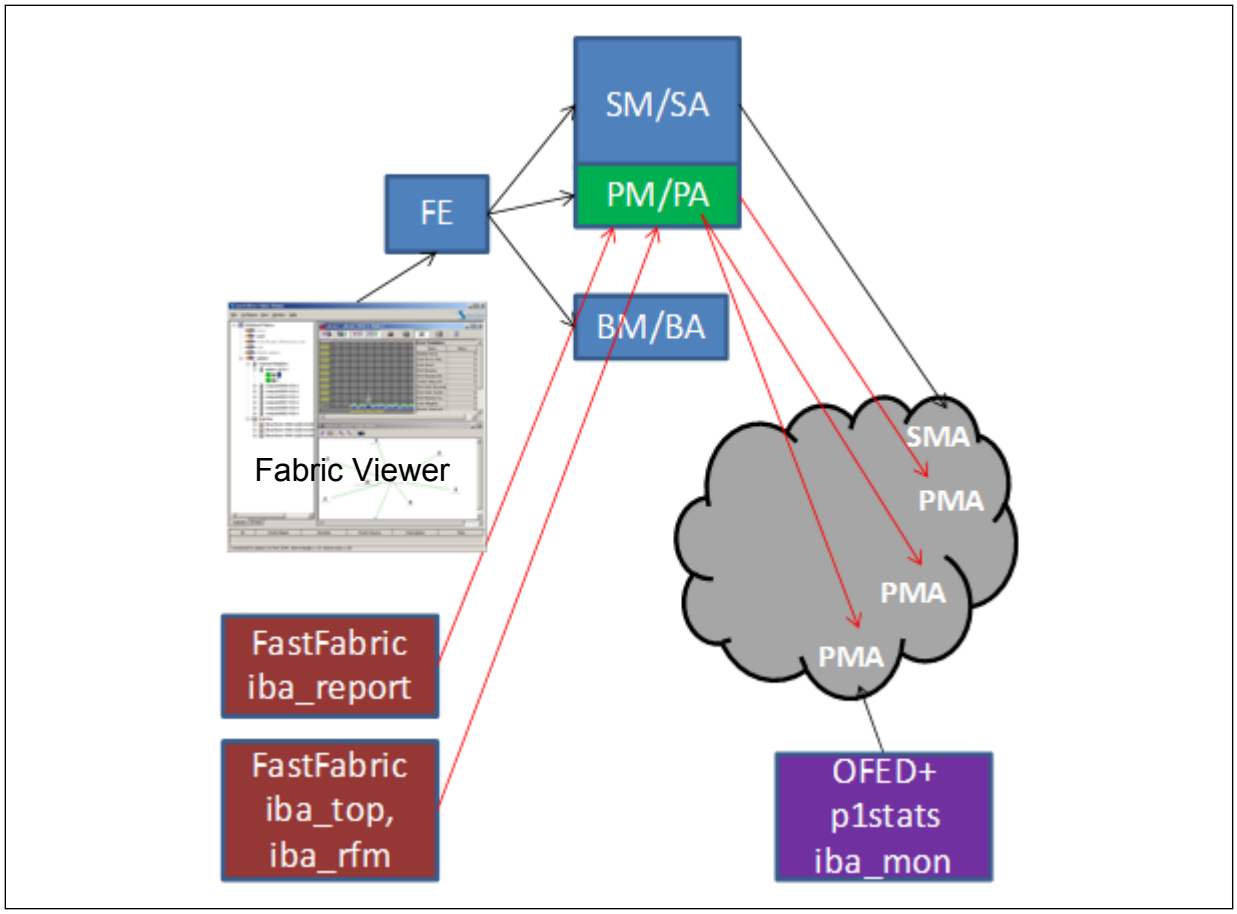
When this capability is enabled, the FM will look for links which are running at SDR speeds but have LinkSpeedSupported values which include DDR and/or QDR on both sides of the link. In which case, the FM will adjust the LinkSpeedEnabled to be DDR or QDR (respectively) and bounce the link to have it run at the new speed. This works hand -in-hand with the Intel® OFED+ s20tune tool, which upon noticing a link which is Down for more than 10 seconds and has only DDR or QDR enabled, will restore the LinkSpeedEnabled on the HCA to match LinkSpeedSupported. This will permit the negotiation process to start over.

Note: Intel® recommends using IBTA compliant switch and HCA firmware. The Firmware is available for all QDR switches, check with your switch vendor to obtain the firmware. All Intel® QDR switches are IBTA compliant. Many DDR switches are not IBTA compliant, however Intel® Switches with level 5.0.4 or later firmware can inter-operate with legacy DDR devices without the need for these FM options.

3.8 Performance Manager

The Performance Manager (PM) is the Fabric Management entity responsible for monitoring fabric information related to the port level counters and the picture that they convey. Each port in the fabric monitors counters that tally information such as the amount of data transmitted and received, as well as occurrences of errors that indicate problems at the port and link level. Refer to Figure 11 for an example of the management paths.

Figure 11. Management Paths



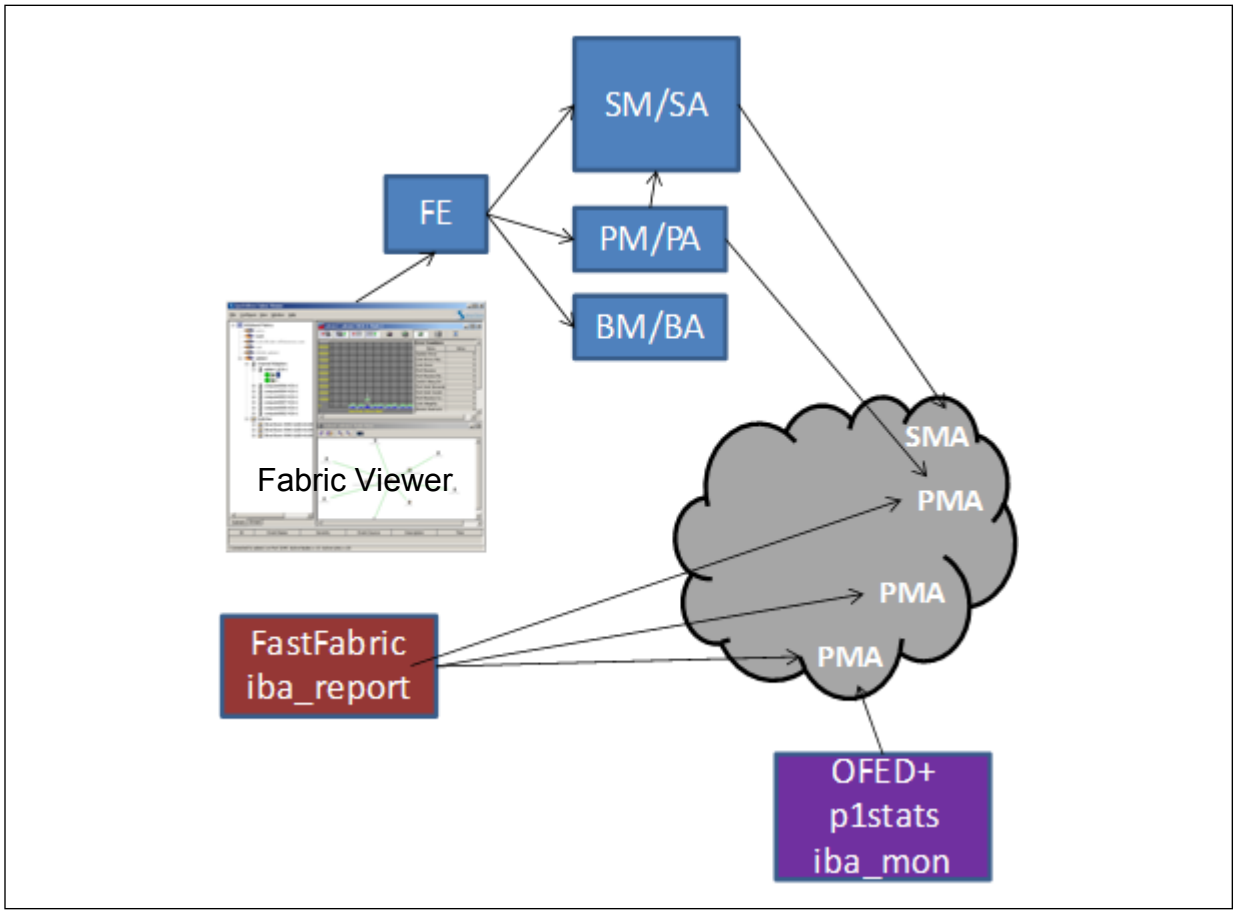
The management of the port counters is not centralized without PM and various management entities in the fabric are able to clear (reset to zero) these counters, without regard for coordination among monitoring tools. The PM can perform regular sweeps of the fabric to gather and summarize both error and performance statistics. This capability allows for centralized control of PMAs and can avoid the potential confusion from many tools (`iba_report`, `iba_mon`, and so on) directly clearing PMA counters.

The Performance Administration (PA) entity allows a centralized control point for querying the performance data in the fabric. The relationship of the PM and PA mirrors that of the Subnet Manager and Subnet Administration (SM/SA) entities; the PM is the "workhorse", performing communication with the Performance Management Agents (PMAs) in the fabric, and, with the use of a calculation "engine", stores the information in the PA database. The PA is then available as a service, similar to the SA, to provide performance information to client applications in the fabric. Refer to Figure 12 for an example of the management paths prior to version 6.0.

Note: The PM when enabled takes control of all PMAs in the system. As such:

- Tools which directly access the PMAs should not be used. Such as `iba_mon`, and chassis port thresholding (`ismAutoClearConf` for Intel® Chassis should be set to disabled)
- Tools which query the PMA counters may yield unexpected results.

Figure 12. Management Paths Prior to Release 6.0



3.8.1 Port Groups

The PA separates the ports in the fabric into groups, which it then monitors according to the performance of the ports in the group. The following pre-defined groups are built into in the PM/PA:

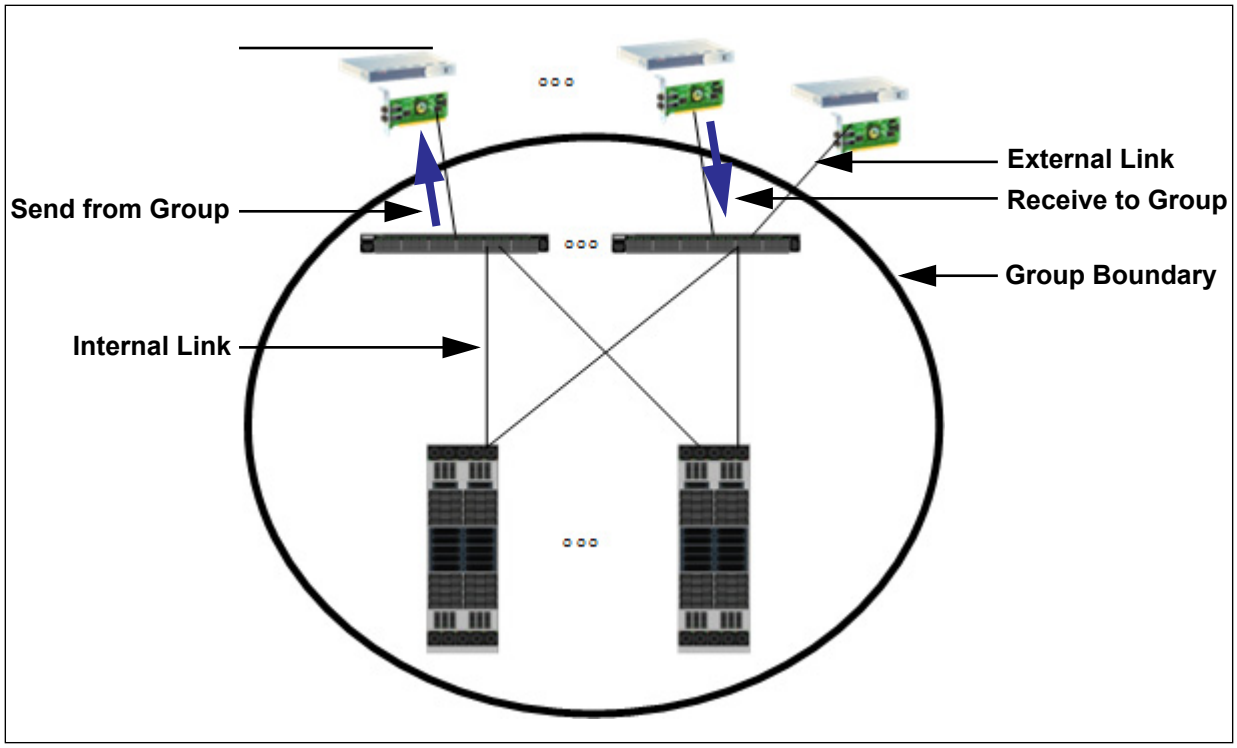
- All – all ports in the fabric
- HCAs – all ports on HCAs in the fabric
- TCAs – all ports on TCAs in the fabric
- SWs – all ports on switches in the fabric

Non-Enhanced Switch Port 0's do not support PMA statistics and are omitted from all Groups.

Within groups, the ports are defined as internal and external.

A port for which both it and its neighbor are in the group is considered an Internal Port. For example ISLs are Internal to the SWs group. Refer to Figure 13 for an example of internal and external links.

Figure 13. Internal and External Links



A port for which one side of the link is in the group but its neighbor is not in the group, is considered an External port. For example a link between an HCA and a Switch will tabulate both the Switch Port and the HCA port as External to the SWs group.

For External ports, Send (data leaving the group) and Recv (data entering the group) statistics are accumulated. Error counters are also accumulated to reflect the overall status of the link.

Note: Some Ports do not support a PMA (such as un-managed Switch Port 0) also for some devices the PM may be told to avoid the PMA by the FM configuration. Such devices are omitted from all Groups.

3.8.2 PM Sweeps

The PM performs regular sweeps of the fabric to get and analyze PMA counters for all the devices in the fabric. The PM Sweep Interval is configurable using the `Pm.SweepInterval` parameter. If set to 0, PM automatic sweeps are disabled. If `Pm.SweepInterval` is not specified, the default is 0, therefore when an upgrade and continue using the old configuration file is accomplished, the PM engine will be disabled.

For fabrics with devices which support 64-bit counters, the value can be set much larger. Larger values will reduce the frequency at which data is gathered and decrease OS jitter and overhead on the FM node. Smaller values will increase the frequency of sweeps (and the frequency at which fresh data is available to tools like `iba_report`, `iba_top`, `iba_rfm`, `iba_paquery`, and so on).

Each sweep typically involves only a few packets per node, so even at faster sweep rates, overhead per node is quite modest. Note that at full QDR speeds a 32-bit data movement counter can overflow in slightly over four seconds. However 64-bit counters take over a hundred years to overflow.

See the *Intel® True Scale Fabric Suite FastFabric User Guide* for more information about tools which can query and/or analyze the data available from the PM (`iba_report`, `iba_top`, `iba_rfm`, `iba_paquery`, and so on).

3.8.3 PA Images

The PM is constantly monitoring the port counter information in the fabric using periodic sweeps. Each sweep gathers the port counters in the fabric and performs reduction and summary calculations of the data to provide group level information in the PA.

Each sweep is stored as a PA image. An image is a historical representation of the fabric data, a time slice. The number of images stored in the PA is finite, with the oldest images being replaced with new ones as resources dictate. The images are identifiable and may be retrieved for examination as long as they still exist in the PA (if they have not been replaced with a newer image due to resource constraints).

The recent history is retained in the memory of the PM. The amount of recent history is configurable using `Pm.TotalImages`.

The PM also supports a `FreezeFrame` capability for the history. This can allow a client to freeze an existing image to allow extended analysis without subsequent sweeps reusing the image. This is used both for the viewing of any non-live data in `iba_top` and `iba_rfm` and the bookmarking of historical data for later analysis. The number of freeze frames is configurable using `Pm.FreezeFrameImages`. This permits tools such as `iba_top` and `iba_rfm` to view recent historical data. The history includes performance, error and topology information. When viewing recent history changes, any of these areas can be viewed and analyzed.

3.8.4 PA Client Queries

PA services available to clients include information about the PM/PA configuration, groups, images, and ports.

Using a set of vendor-specific MAD packets, client applications in FastFabric communicate with the PA to retrieve data that is then available to show to the end user.

Examples of queries supported by the PA include:

- PM/PA configuration – information such as sweep rate, maximum number of images, and so on.
- List of groups
- Group configuration – list of ports in the group, including node information
- Group information – a summary of the statistical categories being monitored for the group
- Group focus – a "top talkers" or "top offender" type summary giving the highest or lowest in contributors to a particular focus area such as bandwidth or error situation
- Port counter management, including retrieval and clearing
- Image management, including configuration retrieval and freeze frame management

3.8.5 Error Thresholding

Thresholds can be configured for each class of PMA Error Counter. These thresholds are used for summary information reported by `iba_top` and other FastFabric tools. In addition the `PM.ThresholdsExceededMsgLimit` parameters in the FM configuration can specify how many threshold exceeded events for each class of counters should be logged. This logging can provide a long term history of fabric errors.

3.8.6 Counter Classification

The PM classifies counters into a number of higher level conditions. Each condition can be computed using counters on either side of the link (that is, the associated port or its neighbor port). The computations and use of counters is done such that the condition is associated with the port which is the most likely root cause or the one which will be experiencing the results of the condition (such as packet loss). Tools such as `iba_top` report the conditions for the associated port. Detailed displays can then show the condition and the devices for both sides of a link.

Figure 14 is an example of utilization. Bandwidth (MBps) and Packet Rate (KPps)) are tallied on the send side, using the maximum of the associated port's send counters and the neighbor port's receive counters

Figure 14. Utilization

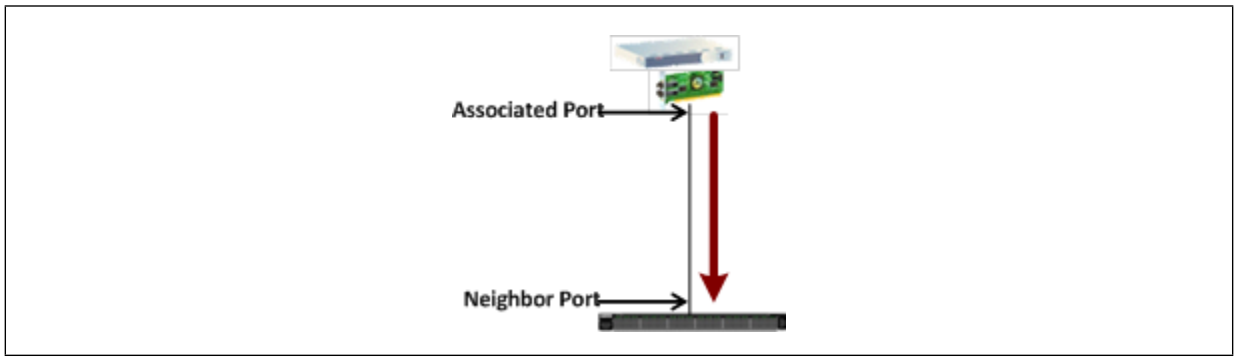
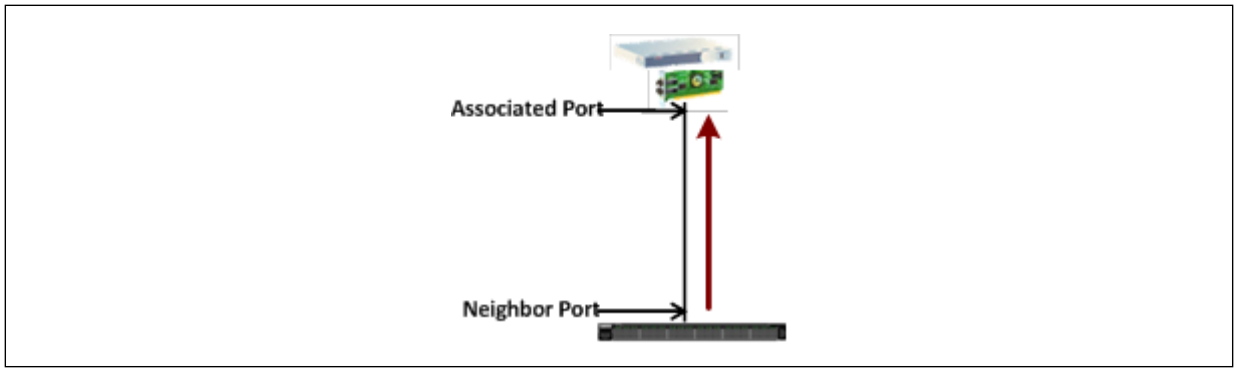


Figure 15 is an example of condition. Integrity, Congestion, SmaCongestion, Security, and Routing are tallied on receive side.

Figure 15. Condition



The conditions are shown in Table 1.

Table 1. PM Conditions and Associated Ports Counter

Condition	Associated Port	Neighbor Port
Bandwidth (MBps) (MAX used)	PortXmitData	PortRcvData
Packet Rate (KPps) (MAX used)	PortXmitPkts	PortRcvPkts
Integrity (weighted sum of below)	SymbolErrorCounter LinkErrorRecoveryCounter LinkDownedCounter PortRcvErrors LocalLinkIntegrityErrors	ExcessiveBufferOverrunErrors
Congestion (weighted sum of below)		XmitDiscards Congestion percent (from PortXmitCongestion relative to link rate) Inefficiency percent (from Congestion percent relative to link utilization percent)
SmaCongestion	VL15Dropped	
Security (sum)	PortRcvConstraintErrors	PortXmitConstraintErrors (sum)
Routing	PortRcvSwitchRelayErrors	
Adaptive Routing	PortAdaptiveRouting	

The results of the associations of conditions to ports is as follows:

- **Bandwidth/Packet Rate** – Associated with port generating the traffic. Typically transmit value of one port matches receive value of the neighbor. However the larger of the two is reported for the link. Bandwidth is tracked as an absolute value as well as percentage of wire speed. Packet rate is only tracked as an absolute value.
- **Integrity** – Associated with the port which is having problems receiving packets. The Integrity value is computed as a weighted sum, the weight for each counter is configurable. ExcessiveBufferOverruns can mean that the local port is not properly receiving link flow control, so it is incorrectly causing an overflow on the neighbor. Link integrity problems can ultimately be due to hardware on either end of the link or the cable.
- **Congestion** – Associated with the port which is unable to process the incoming data rate. This port is causing backpressure and fabric congestion. The value is computed as a weighted sum, the weight of each counter is configurable. Some of the values such as Inefficiency percent are computed based on ratios of congestion to the link utilization.
- **SmaCongestion** – Associated with the port which is unable to process the inbound VL15 traffic. This port is discarding VL15 SMA packets.
- **Security** – Associated with the port which is attempting to be cracked and therefore packets are being sent to this port which violate security
- **Routing** – Associated with the port which is receiving packets it cannot route. This is only applicable to switch ports on Intel® Switches (many 3rd party switches incorrectly tabulate this counter so it is ignored for non-Intel® switches).
- **Adaptive Routing** – Associated with the port which is doing the adapting. This is only applicable to switch ports on Intel® 12000 series switches with 6.0 or newer firmware versions. See Section 3.2.3, “Adaptive Routing” on page 34 for more information.

Note: The PortRcvRemotePhysicalErrors counter is not included in any of these conditions. This counter indicates packets which were marked bad due to discovery of a packet error during switch cut through. When such errors occur the counters included in Integrity will allow the issue to be pinpointed. In contrast this counter will occur for every hop along the route and is not very useful for the root causing the Integrity issue.

3.8.7 Congestion Statistics

Congestion is a weighted sum of five items:

- **XmitDiscard** – Count of packets sent which were discarded due to congestion, link going down or invalid MTU/routing. Note that packets are only discarded in the event of extreme congestion.
- **XmitCongestionPct10** – Percentage of time that the outgoing port was congested and the outbound queue was deep. This value is only available for Intel® switches and requires `PmaSwVendorEnable` to be enabled. This is expressed as an integer *10. So for example 100% is reported as 1,000
- **XmitInefficiencyPct10** – Percentage of time that the outgoing port was congested when it was transmitting. Computed as XmitCongestionPct/(XmitCongestionPct+Utilization%) This value is only available for Intel® switches. This is expressed as an integer *10. So for example 100% is reported as 1,000
- **XmitWaitCongestionPct10** – Percentage of time that the outgoing port was congested and unable to transmit. This value is only available for Intel® HCAs and requires that `PmaCaVendorEnable` to be enabled. This is expressed as an integer*10. So for example 100% is reported as 1,000.
- **XmitWaitInefficiencyPct10** – Percentage of time that the outgoing port was congested when it could have been transmitting. Computed as XmitWaitCongestionPct/(XmitWaitCongestionPct+Utilization%). This value is only available for Intel® HCAs. This is expressed as an integer*10. So for example 100% is reported as 1,000.

The default configuration combines XmitDiscard, XmitInefficiencyPct10, and XmitWaitInefficiencyPct10 with XmitDiscard heavily weighted such that any packet discards are weighted higher than inefficiencies. The default threshold of 100 represents 10% inefficiency and/or any packet loss due to extreme congestion.

It is left to the user to decide if the use of XmitCongestionPct10 and XmitWaitCongestionPct10 is of more interest for the given fabric. Typically XmitInefficiencyPct10 and XmitWaitInefficiencyPct10 will report a value higher than XmitCongestionPct10 and XmitWaitCongestionPct10.

3.8.8 Histogram Data

For each condition a histogram is tracked. Each histogram has a set of buckets which count the number of ports in a given group with a certain range of performance/errors.

For Bandwidth, ten histogram buckets are tracked, each 10% wide. The buckets each count how many ports in a given group had this level of utilization relative to wire speed.

For Errors (Integrity, Congestion, SmaCongestion, Security, Routing), five histogram buckets are tracked. The first four are 25% wide and indicate the ports whose error value is the given percentage of the configured threshold. The fifth bucket counts the number of ports whose error rate exceeded the threshold. The overall summary error status for a given group is based on the highest error rate for any port tabulated by the group.

3.8.9 Support for iba_report

In addition to tracking interval counters, the PM also keeps one set of long running counters per PMA. These counters will be accessed by `iba_report` and provide compatible support for existing `iba_report` error analysis and other FastFabric tools, such as `fabric_analysis` and `all_analysis`.

As such, when the PM is running, `iba_report` will access data from the PM, therefore options such as `-o` errors and `-C` will access PM internal copies of PMA counters and will not affect the actual hardware PMA counters.

The long running counters can also be very useful to analyze long term error rates, using tools such as `fabric_analysis`, while still providing short term statistics in `iba_top` and other tools.

3.8.10 64 Bit PMA Counter Support

The PM can take advantage of 64-bit PMA counters for devices which support them. It also takes advantage of switches which support the `AllPortSelect` mechanism for clearing counters. Intel® 12000 series switches running 5.2.0.0.9 or later firmware support both features.

Use of 64-bit counters permits a much longer `SweepInterval` for the PM and can reduce the PM contributions to "OS Jitter". Use of 64-bit counters is enabled by default. The `Pm.Pma64Enable` configuration parameter can be used to enable/disable use of 64-bit counters. When enabled the PMA will query the device, using `ClassPortInfo`, to determine if it supports 64-bit counters and will only use 64-bit counters for devices which support them.

3.8.11 Scalable PMA Retries

To optimize fabric programming time, the FM can be configured to issue multiple PMA packets in parallel. This approach can result in occasional packet loss which can have a negative effect on PM performance.

To allow for rapid recovery from such packet loss, while not causing excessive retries to sluggish nodes, the FM allows for scalable retries which increase with each attempt.

3.8.12 Support for IBM eHCA

Some versions of the IBM eHCA do not fully support a PMA in the Logical Channel Adapters. The `Pm.EhcaPmaAvoid` configuration parameter can be used to disable PMA queries of eHCA Logical Channel Adapters. This will not typically impact the data available since the neighbor Logical Switch port inside the eHCA will provide the same data movement information. Since the links between Logical Channel Adapters and Logical Switches are purely logical, not physical, no signal integrity errors will be reported for these links. Intel® recommends to use the default of `EhcaPmaAvoid=1` to disable these queries and avoid potential problems.

3.9 Multiple FM Instances per Management Host

Table 2 associates managers and their acronyms.

Table 2. Fabric Manager Instances

Fabric Manager	Acronym
Baseboard Manager	BM
Fabric Executive	FE
Performance Manager	PM
Subnet Manager	SM

Note: A host can be connected to as many subnets as the number of ports the HCAs contain. If a host contains one 4x HCA and both Ports are connected to different subnets, the host can run a maximum of two BM, PM, FE and SM instances, if adequate memory is available.

3.9.1 Subnet Configuration Example

Fabric managers can run on various hosts. It is suggested that one or two FM hosts be dedicated per Subnet for fabric management. The remaining hosts can be used to run applications.

Note: Running systems management and application software on hosts that are dedicated to Subnet management is discouraged.

Table 3 shows an example subnet in which hosts A and B share subnet management responsibilities; host A runs the primary Subnet Manager while host B is the standby. A Fabric Executive is active on both hosts while the Baseboard Manager is assigned to only one host.

Table 3. Subnet Configuration Example

Host	BM	FE	PM	SM
A	0	1	1	1
B	1	1	1	1
C,D,...	0	0	0	0

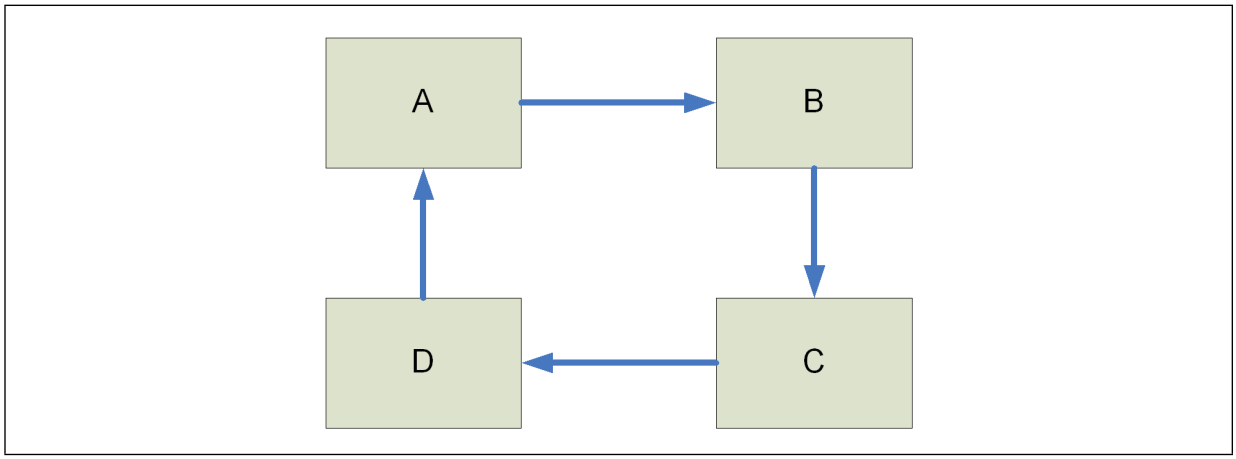
3.10 SM Loop Test

3.10.1 Loop Test introduction

The SM Looptest is a diagnostic test facility in the SM. As part of this test, the SM stress tests inter-switch links (ISLs) by continuously passing traffic through them. Other tools like FastFabric can be used to monitor the links for signal integrity issues or other errors. The advantage of the Looptest, is that it provides a guaranteed way to test all of the ISLs in the fabric, without the need for a large number of end hosts or applications.

When the Looptest is started, the SM deliberately sets up certain additional routes with loops in addition to the normal routes that are setup to enable communication with end ports. These loop routes only include ISLs and are not used for communication with end ports, but are used only to stress test the ISLs. Each loop route starts from a switch, A (Figure 16), uses one of its ISL to another switch and pass through a set of other ISLs and switches (say B, C, D) and ends up back at switch A but via a different ISL.

Figure 16. ISL Loop Routes



Warning: Loop test should not be run when applications are running on the fabric. The loop test introduces high volume traffic on the switch ISLs which would slow down normal application traffic.

Warning: Since there are additional LIDs for each of the loop route LIDs, the user may observe a large number of additional LIDs in the switch LID Forwarding Tables (LFTs) when the loop test is running.

As part of loop test the SM sets up a large number of these loop routes so as to cover all ISLs in the fabric. The SM associates a LID with each of the loop routes in such a way that a packet sent to that LID will enter the loop and spin around the loop utilizing the ISLs on that loop. The SM injects a packet to into each loop by sending a packet to each of the LIDs associated with the loop routes. Those packets loop around the loops and continuously pass traffic on the ISLs.

Once loop test is stopped, the SM invalidates the loop LIDs, which in turn would cause the loop packets to be dropped by the switches and stop the utilization of the switch ISLs.

3.10.2 Important loop test parameters

- **Loop path length** – The number of hops the SM checks to find a loop that leads back to the same switch is termed as the loop path length.
- **Number of packets injected** – The number of 256-byte packets the SM injects into each of the loops. These packets would loop around on the ISLs.

These parameters can be configured as part of the loop test.

3.10.3 Loop Test Fast Mode

Intel® recommends the customers use the fast mode for ISL validation and link integrity testing. In the fast mode, the loop test doesn't attempt to include each ISL in all possible loops, but includes it in at least the specified number of loops (this value is controlled using the MinISLRedundancy parameter). When using fast mode the computations are less expensive and finish faster which allows the loop test to be started quickly. The loop test fast mode uses a MinISLRedundancy value of four by default (for example, each ISL is included in at least four loops). There is an important reason for including an ISL in multiple loops – when an ISL goes down in the fabric, the loop that this ISL is part of, is broken and therefore other ISLs in the loop will no longer

be tested. But if each ISL is included in multiple loops, the other ISLs in the broken loop are also part of other loops and will continue to see traffic and therefore continue to get tested (albeit at a slightly lower utilization).

In typical fast mode operations (with the default MinISLRedundancy of 4), injecting four packets into each loop is sufficient to get a high utilization on the ISLs.

3.10.4 Loop Test Default Mode

By default the loop test runs in default mode. In the default mode, the SM uses an exhaustive approach to setup loop routes and will include each ISL in as many loops as possible. This ensures that each ISL is exactly in the same number of loops and hence will see the same amount of utilization. But finding all possible loops is computationally intensive and can take a long amount of time.

In the default mode, to keep the computations to find loops to a manageable level, the SM by default uses only a path length of three i.e. three hop loops. Three hop loops are usually sufficient for checking ISLs when there are external cables between leafs of the same switch chassis or in other small fabrics but will not be sufficient for checking all ISLs in large fabrics that involve multiple switch chassis. For such fabrics the Loop Test Fast Mode should be used.

In typical default mode operations, injecting one packet into each loop is sufficient to get a high utilization on the ISLs included in the loops.

3.10.5 SM Loop Test Setup and Control Options

There are two ways to run the SM loop test. The test can be run using CLI commands or can be configured to run using the FM Configuration file. Both of these methods will be discussed in the following sections.

- Chapter 3.0, “Run the SM Loop Test using CLI Commands”
- “Setup the Configuration File to Run the SM Loop Test” on page 59

3.10.6 Run the SM Loop Test using CLI Commands

3.10.6.1 Requirements to run the SM Loop Test

- FM must be running for host or embedded nodes in order for the SM loop test to run.
- The FM must be master.
- To run in fast mode:
  - Fast mode must be enabled using the Loop Test Fast Mode command when starting loop test.
  - Enabling fast mode will automatically set the path length to 4 and the inject on each sweep to disabled.
  - The fast mode can only be disabled by running the Stop command that stops the loop test completely.

3.10.6.2 Loop Setup Options

- Set up the path length to set the number of hops that the FM checks to find a loop that leads back to the same switch

```
/opt/ifs_fm/bin/fm_cmd smLooptestPathLength
```

Values are 2-4 with a default of 3
Loop path length is set to 4 for Fast Mode

Note: Intel® recommends to change this parameter before starting loop test. If the parameter is changed after starting loop test, packets will have to be injected with `fm_cmd smLooptestInjectPackets` option.

- Set the minimum number of loops in which to include each ISL

```
/opt/ifs_fm/bin/fm_cmd smLooptestMinISLRedundancy
```

Applicable only when running in Fast Mode
Default is 4

3.10.6.3 Packet Injection Options

- Packets are injected if the loop test is started normally and a number of packets has been specified as an argument to the command:

```
/opt/ifs_fm/bin/fm_cmd smLooptestStart 4
```

- Four Packets are injected by default if loop test is started in Fast Mode:

```
/opt/ifs_fm/bin/fm_cmd smLooptestFastModeStart
```

In fast mode, once loop test is started packets are injected only once. They are not injected at every sweep.

- Packets can be injected once the loop test has been started:

```
/opt/ifs_fm/bin/fm_cmd smLooptestInjectPackets numPkts
```

- Packets can be injected to specific switch node:

```
/opt/ifs_fm/bin/fm_cmd smLooptestInjectAtNode SwNodeIndex
```

- Packets can be injected on each sweep:

```
/opt/ifs_fm/bin/fm_cmd smLooptestInjectEachSweep
```

The `fm_cmd smLooptestInjectEachSweep` parameter can be used in both fast and default mode loop test case to control whether each sweep will inject a packet or not. In fast mode, the default is not to inject on each sweep, but it can be changed to inject on each sweep after the loop test is started. In default mode, this parameter can be changed either before or after loop test is started to enable packet injection on each sweep.

3.10.6.4 Other SM Loop Test Commands

For a full list of commands that can be used for the SM Loop Test refer to "fm_cmd" on page 197.

To stop the SM Loop Test:

```
/opt/ifs_fm/bin/fm_cmd smLooptestStop
```

3.10.7 Setup the Configuration File to Run the SM Loop Test

The FM configuration file can be set up to automatically run the SM loop test when the master FM is started. The parameters are added to the `Miscellaneous` section of the configuration file and can include the following three items:

- Run at start up:

```
<LoopTestOn>1</LoopTestOn>
```

  1 = enabled
  0 = disabled

- Run in fast mode:

```
<LoopTestFastMode>1</LoopTestFastMode>
```

  1 = enabled
  0 = disabled
  Once fast mode is enabled, whenever loop test is started it will use the fast mode.

- Set the number of packets injected for each test run:

```
<LoopTestPackets>4</LoopTestPackets>
```

  With fast mode, you will need around 4 packets to get a high utilization on the links.

3.10.8 Reports from SM LoopTest

The following reports can be requested for the SM loop test:

3.10.8.1 SM Loop Test Show Loop Paths

Figure 17 is an example of the SM Loop Test Show Loop Paths Report

Figure 17. SM Loop Test Show Loop Paths Report Example

```
# /opt/ifs_fm/bin/fm_cmd smLooptestShowLoopPaths
   Connecting to LOCAL FM instance 0
   Successfully sent Loop Test Path show for node index (all) to local SM instance

   Node Idx: 1, Guid: 0x00066a2400000000 Desc 9024 #0
   Node Idx: 2, Guid: 0x00066a2400000003 Desc 9024 #3

-------------------------------------------------------------------------------
   Node     Node                              Node     Path
   Idx      Lid             NODE GUID        #Ports    LID      PATH[n:p->n:p]
   ----   ---------  --------------------   -------- ------   ------------------
    1       0x0001    0x00066a2400000000       36      0x0080  1:1->2:1 2:2->1:2
```

3.10.8.2 SM Loop Test Show Switch LFT

Figure 18 is an example of the SM Loop Test Show Switch LFT Report

Figure 18. SM Loop Test Show Switch LFT Report Example

```
# /opt/ifs_fm/bin/fm_cmd smLooptestShowSwitchLft

Connecting to LOCAL FM instance 0
Successfully sent Loop Test LFT show for node index (all) to local SM instance

Node[0001]  LID=0x0001  GUID=0x00066a2400000000 [9024 #0] Linear Forwarding Table
      LID      PORT
      ------   ----
      0x0001   0000
      0x0002   0001
      0x0003   0010
      0x0004   0020
```

3.10.8.3 SM Loop Test Show Topology

Figure 19 is an example of the SM Loop Test Show Topology Report

Figure 19. SM Loop Test Show Topology Report Example

```
# /opt/ifs_fm/bin/fm_cmd smLooptestShowTopology
Connecting to LOCAL FM instance 0
Successfully sent Loop Test topology show to local SM instance

sm_state= MASTER count= 29796 LMC= 0, Topology Pass count= 4, Priority= 0, Mkey= 0x0000000000000000

-------------------------------------------------------------------------------------------
george HCA-1
-------------------------------------------------------------------------------------------
Node[  0] => 0079159a00117500  (1)  ports=1, path=
Port ----  GUID  ---- (S)    LID      LMC      _VL_ __MTU__ __WIDTH__ ___SPEED____  CAP_MASK N#  P#
 1   0079159a00117500 4  LID=000d  LMC=0000   2  1  4k  2k  1X/4X  4X 2.5-10 2.5   0761086a 1  19

-------------------------------------------------------------------------------------------
9024 #0
-------------------------------------------------------------------------------------------
Node[  1] => 0000000000066a24  (2)  ports=36, path= 1
Port ---- GUID ----  (S)   LID       LMC      _VL_ __MTU__ __WIDTH__ __SPEED___  CAP_MASK N# P#
 0   0000000000066a24 4  LID=0001  LMC=0000   8  8  2k  2k 1X/4X 4X  2.5/5 5.0   00100848 1   0 1
 1   0000000000000000 4                       8  8  2k  2k 1X/4X 4X  2.5/5 5.0   00000048 2   1
 2   0000000000000000 4                       8  8  2k  2k 1X/4X 4X  2.5/5 5.0   00000048 2   2
```

3.10.8.4 SM Loop Test Show Configuration

Figure 20 is an example of the SM Loop Test Show Configuration Report

Figure 20. SM Loop Test Show Configuration Report Example

```
# /opt/ifs_fm/bin/fm_cmd smLooptestShowConfig
    Connecting to LOCAL FM instance 0
    Successfully sent Loop Test configuration show to local SM instance
    Loop Test is running with following parameters:
    Max Path Length    #Packets    Inject Point
----------------    -------    ----------------
        4             00005       All Nodes
```

§ §

4.0 Fabric Manager Configuration

This chapter uses the InfiniBand* Architecture Standard terms in Table 4.

Table 4. InfiniBand* Architecture Standard Terms

Term	Definition
Endnode	An endnode is any node that contains a Channel Adapter and thus it has multiple queue pairs and is permitted to establish connections, end to end context, and generate messages. Also referred to as HCA or TCA, two specific types of endnodes.
Host	One or more HCAs governed by a single memory/CPU complex.
Subnet	A set of Ports using InfiniBand* Architecture, and associated Links, that have a common Subnet ID and are managed by a common Subnet Manager.
Port	Location on a Channel Adapter or Switch to which a link connects. There may be multiple Ports on a single Channel Adapter each with different context information that must be maintained. Switches/switch elements contain more than one port by definition.

4.1 Configuring the FM

This section describes how to configure a host or embedded Fabric Manager and subnet.

The Fabric Manager configuration file, `ifs_fm.xml`, defines the following:

- FM management entities to be run on this FM host or chassis for each subnet to which the host is attached (by definition a chassis is only connected to one subnet)
- Operational parameters for the fabric, such as routing and time-outs
- vFabrics™ to be configured
- Configuration for each FM component, such as sweep rates, logging options, and retries

The configuration file formats for the host and embedded FMs are the same. There are some additional parameters which are available on the host FM. If those parameters are present in the configuration file, they will be ignored on the embedded FM, therefore you can use the same configuration file for both host and embedded managers.

To make a change to a FM's configuration, edit the `ifs_fm.xml` configuration file and restart the FM.

On a host this is done by editing `/etc/sysconfig/ifs_fm.xml` then restarting the FM using the Fabric Manager startup script, `/etc/init.d/ifs_fm restart`.

On a chassis this is done by downloading the `ifs_fm.xml` file to a host, editing it, uploading the new file to the chassis, then restarting the FM using the `smControl` command. The FastFabric Tools also provide CLI and interactive tools to assist in managing the configuration and operation of embedded FMs. Refer to the *Intel® True Scale Fabric Suite FastFabric User Guide* for information on using FastFabric to transfer the xml file to the switch.

The use of XML for the FM configuration file offers a number of significant and powerful advantages:

- XML is a standard, there are many 3rd party tools which can edit and read it
- 3rd party editors understand XML and can do text highlighting and syntax checking

- Existing FastFabric tools such as `xml_extract`, `xml_indent`, `xml_filter`, and `xml_generate` can edit the file
- Existing linux tools such as `xml_grep` can search in the `.xml` file

4.1.1 Manager Instances

A given FM host can run multiple FM instances. Each instance is assigned to a single local HCA port. An Embedded FM in a chassis supports only one instance.

Note: A host can be connected to as many subnets as the number of ports the HCAs contain. If a host contains one 4x HCA and both Ports are connected to different subnets, the host can run a maximum of two BM, PM, FE and SM instances, if adequate memory is available.

4.1.1.1 Configuration File Syntax

The `ifs_fm.xml` file uses standard XML syntax to define configuration settings.

While XML is quite powerful, only a handful of basic syntaxes are needed to create FM configuration files.

Comments are ignored during parsing and may be used to annotate the file as needed. The following is a comment:

```
<!-- this is a comment -->
```

In the sample configuration file, comments are kept to a single line for readability, however a comment can span multiple lines if needed.

A parameter is specified using an XML tag such as:

```
<parameter>value</parameter>
```

All XML tag names are case sensitive. Each parameter starts with the start tag <parameter> and ends with a corresponding end tag: </parameter>. The names of the parameters are as described in the remainder of this section and as shown in the sample `ifs_fm.xml` configuration file. Between the start and end tags a value for the parameter can be provided.

The majority of parameters have numeric values. Numbers can be specified in decimal or hex. Hex numbers should start with 0x. For example: 0x1c is the hex representation of 28 decimal.

A section in the file can contain multiple related parameters. Sections also begin with a start tag and end with an end tag. For example:

```
<section>

<parameter1>value</parameter1>

<parameter2>value</parameter2>

</section>
```

A section may contain zero or more parameters. Some sections contain other sections. The indentation in this example (and the sample `ifs_fm.xml` configuration file) is used for readability but indentation is not required. If a section contains no parameters, it can be omitted.

Throughout this guide, the abbreviation `section.parameter` or `section.subsection` is used to indicate a section containing a given parameter or subsection.

The xml parser will quietly ignore any unrecognized parameter or section. This capability allows for forward and backward compatibility of the configuration files. It will be possible to use newer configuration files (which may have additional parameters) on an older version of the FM. This has the negative risk that a mistyped parameter may be silently ignored and defaulted. To help avoid such mistakes it is recommended to start with the sample configuration file and edit or cut and paste parameters as needed. Also the `config_check` tool can be run in strict mode (`-s` option) to report any unrecognized tags.

The default FM configuration file provided shows all of the tags and provides detailed comments as to their meaning.

The first line in the file must be the following:

```
<?xml version="1.0" encoding="utf-8"?>
```

The `ifs_fm.xml` configuration file is organized into a few top level sections as follows:

```
<?xml version="1.0" encoding="utf-8"?>

<Config>

<!-- Common FM configuration, applies to all FM instances/subnets  -->


<Common>

</Common>


<!-- A single FM Instance/subnet -->

<Fm>

</Fm>


<!-- A single FM Instance/subnet -->

<Fm>

</Fm>


<!-- A single FM Instance/subnet -->

<Fm>

</Fm>
```

```
<!-- A single FM Instance/subnet -->

<Fm>

</Fm>


</Config>
```

The Common section allows for the definition of parameters which will apply to all FM instances. In a typical configuration file, the majority of parameters are defined in the Common section and the Fm sections will only define a few instance specific parameters. If needed, any parameter set in the Common section may be overridden for a given FM Instance by also defining it in the Fm section.

The embedded FM only supports a single FM instance. Only the first Fm section in the ifs_fm.xml file will be used by the embedded FM, any additional Fm sections will be ignored.

The Common and each Fm sections have a similar layout. They each contain the following subsections:

```
<Applications>

</Applications>


<DeviceGroups>

</DeviceGroups>


<VirtualFabrics>

</VirtualFabrics>


<Shared>

</Shared>


<Sm>

</Sm>


<Fe>

</Fe>


<Pm>

</Pm>
```

```
<Bm>

</Bm>
```

The Applications, DeviceGroups and VirtualFabrics sections are used to configure vFabrics. This will be covered in greater detail in Chapter 5.0, "Virtual Fabrics".

The Shared section allows for definition of parameters which will apply to all managers (Sm, Fe, Pm, and Bm). This section typically defines logging and redundancy options. If needed, any parameter set in the Shared section may be overridden for a given manager by also defining it in the Sm, Fe, Pm and/or Bm sections.

4.1.2 Shared Parameters

Table 5 describes parameters which may be used in the Shared subsection of either the Common or Fm sections.

Table 5. Shared Parameters

Parameter	Default Value	Description
SubnetSize	2560	Maximum number of endports (connected HCA and TCA ports) that the fabric is expected to support. Since several internal parameters are derived from this, it's important that this number not be set to less than the true number of connected end ports in the fabric. For the embedded SM, if this value exceeds the capabilities of the chassis, the value given will be appropriately reduced. Refer to the *Intel® True Scale Fabric Switches 12000 Series Release Notes* for information on ESM subnet sizes supported.
CoreDumpLimit	0	Max core file size in bytes. Can be unlimited or a numeric value. Suffix of K, M or G can be used to aid specification of larger values, such as 400M for 400 Megabytes A value of 0 disables generation of core dumps. For the Embedded FM, this parameter is ignored. This capability uses the core dump ulimit (setrlimit) capabilities of linux and sets the soft limit. If the hard limit is set to 0 or the hard limit is smaller than CoreDumpLimit the FM will not be able to create dumps and a message will be logged on FM startup.
CoreDumpDir	/var/crash/ifs_fm	Directory to dump to. A unique file will be created for each core file per the standard linux kernel configuration This directory will be created, but its parent directory must pre-exist. For the Embedded FM, this parameter is ignored. Default parent directory of /var/crash/ may not exist on some systems. Make sure to create this directory or use an appropriate existing directory name before enabled FM core dumps.
Priority ElevatedPriority	0 0	The startup priority of each manager in this Fm Instance. 0-15 is allowed. Priority and Elevated Priority control failover for SM, PM and BM. Priority is used during initial negotiation, high Priority wins. ElevatedPriority is assumed by the winning master. This can provide sticky failover and prevent fallback when previous master comes back online.

Table 5. Shared Parameters (Continued)

Parameter	Default Value	Description
LogLevel	2	Sets log level option for SM, PM, BM and FE: 0 = disable vast majority of logging output 1 = fatal, error, warn (syslog CRIT, ERR, WARN) 2 = +notice, INFIIINFO (progress messages) (syslog NOTICE, INFO) 3 = +INFO (syslog DEBUG) 4 = +VERBOSE and some packet data (syslog DEBUG) 5 = +debug trace info (syslog DEBUG) This parameter is ignored for the Embedded FM. Refer to the *Intel® True Scale Fabric Switches 12000 Series User Guide* for information on configuring chassis logging options.
LogFile		Sets log output location for SM, PM, BM and FE. By default (or if this parameter is empty) log output is accomplished using syslog. However, if a LogFile is specified, logging will be done to the given file. LogMode further controls logging. This parameter is ignored for the Intel® Embedded FM. Refer to the *Intel® True Scale Fabric Switches 12000 Series User Guide* for information on configuring chassis logging options.
LogMode	0	Controls mode for logging by SM, PM, BM and FE. low two bits control logging as follows: Low Bit (0/1): 0 - use normal logging levels 1 - logging is quieted by downgrading the majority of fatal, error, warn and infiniinfo log messages to level 3 (INFO) and only outputting user actionable events when LogLevel is 1 or 2 Next Bit (0/2) (only affects logging when LogFile specified): 0 - user actionable events go to syslog and LogFile 2 - when LogFile specified, nothing goes to syslog This parameter is ignored for the Embedded FM. Refer to the *Intel® True Scale Fabric Switches 12000 Series User Guide* for information on configuring chassis logging options.
SyslogFacility	local6	For the Host FM, controls what syslog facility code is used for log messages, Allowed values are: auth, authpriv, cron, daemon, ftp, kern, local0-local7, lpr, mail, news, syslog, user, or uucp. For the Embedded FM, this parameter is ignored
ConfigConsistencyCheckLevel	2	Controls the Configuration Consistency Check for SM, PM, and BM. Checking can be completely disabled, or can be set to take action by deactivating Standby SM, PM, or BM if configuration does not - pass the consistency check criteria. 0 = disable Configuration Consistency Checking 1 = enable Configuration Consistency Checking without taking action (only log a message) 2= enable Configuration Consistency Checking and take action (log message and move standby to inactive state)

Table 5. Shared Parameters (Continued)

Parameter	Default Value	Description
ConfigConsistencyCheckMethod	0	Controls the checksum generation method for Configuration Consistency Checking between redundant FMs. MD5 cannot be used when a Host FM and Embedded FM are being used as a redundant pair. In which case the simple additive checksum must be used. The simple additive checksum method may fail to detect some configuration differences. 0 = use MD5 checksum method 1 = use simple additive checksum method
DefaultPKey	0xffff	The PKey for PM, BM and FE. This should be the Default PKey so they can manage all nodes.
Debug RmppDebug	0	Additional parameters for debug/development use - These enable debugging modes for SM, PM, BM and FE. RmppDebug can be overridden by other RmppDebug in individual components.
CS_LogMask MAI_LogMask CAL_LogMask DVR_LogMask IFS_LogMask SM_LogMask SA_LogMask PM_LogMask PA_LogMask BM_LogMask FE_LogMask APP_LogMask	0x1fff	For advanced users, these parameters can provide more precise control over per subsystem logging. For typical configurations these should be omitted and the LogLevel parameter should be used instead. For each subsystem there can be a LogMask. The mask selects severities of log messages to enable and is a sum of the following values: 0x1=fatal 0x2=actionable error 0x4=actionable warning 0x8=actionable notice 0x10=actionable info 0x20=error 0x40=warn 0x80=notice 0x100=progress 0x200=info 0x400=verbose 0x800=data 0x1000=debug1 0x2000=debug2 0x4000=debug3 0x8000=debug4 0x10000=func call 0x20000=func args 0x40000=func exit For Embedded SM corresponding Chassis Logging must also be enabled and Sm configuration applies to all managers For Host SM, the linux syslog service will need to have an appropriate level of logging enabled.

For example:

```
<Shared>

<SubnetSize>2560</SubnetSize>

<CoreDumpLimit>0</CoreDumpLimit>

<CoreDumpDir>/var/crash/ifs_fm</CoreDumpDir>

<Priority>0</Priority>
```

```
<ElevatedPriority>0</ElevatedPriority>

<LogLevel>2</LogLevel>

<LogMode>0</LogMode>

<SyslogFacility>local6</SyslogFacility>

<ConfigConsistencyCheckLevel>2</ConfigConsistencyCheckLevel>

<ConfigConsistencyCheckMethod>0</ConfigConsistencyCheckMethod>

<Debug>0</Debug>

<RmppDebug>0</RmppDebug>

</Shared>
```

4.1.3 Controlling FM Startup

The embedded FM's startup is controlled using the Chassis CLI commands `smConfig`, `smControl` and `smPmBmStart`. For the embedded FM the various `<Start>` parameters will be ignored.

Note: The remainder of this section applies only to the host FM.

Unlike other `Shared` section parameters, the `Start` parameter is specially handled. The `Start` parameter may be used in the `Fm.Shared` section or any `Sm`, `Fe`, `Pm`, or `Bm` section (in `Common` or `Fm`). However, it can not be used in the `Common.Shared` section and will be ignored if it is used there. In order for a given FM Instance to be started, the `Fm.Shared` section must have a Start section parameter with a value of 1. Within a given FM Instance, only the managers whose `Start` section parameters are also 1 (they default to 1) will be started. For example:

```
<Common>

<Sm>

<Start>1</Start>

</Sm>

<Fe>

<Start>1</Start>

</Fe>

<Pm>

<Start>1</Start>

</Pm>

<Bm>

<Start>0</Start>

</Bm>

</Common>
```

```
<Fm>
<Shared>
<Start>1</Start>
</Shared>
</Fm>

<Fm>
<Shared>
<Start>0</Start>
</Shared>
</Fm>
```

The above example, will start the Sm, Fe and Pm for the first FM Instance, but it will not start the Bm. The second FM Instance will not be started at all.

The following example, will start the Sm, Fe and Bm for the first FM Instance, but it will not start the Pm. The second FM Instance will be started with an Sm, Fe, and Pm but no Bm.

```
<Common>
<Sm>
<Start>1</Start>
</Sm>
<Fe>
<Start>1</Start>
</Fe>
<Pm>
<Start>1</Start>
</Pm>
<Bm>
<Start>0</Start>
</Bm>
</Common>

<Fm>
<Shared>
<Start>1</Start>
```

```
</Shared>

<Pm>

<Start>0</Start> <!-- override Common.Pm -->

</Pm>

<Bm>

<Start>1</Start> <!-- override Common.Bm -->

</Bm>

</Fm>

<Fm>

<Shared>

<Start>1</Start>

</Shared>

</Fm>
```

Typically, individual managers will be controlled in the `Common` section and Fm Instance startup will be controlled in the `Fm.Shared` section. In most configurations, only a single FM Instance will be run with all managers started, in which case the default `ifs_fm.xml` file can be used as provided.

Note: The SM and the PM are unified. As such in order to enable the PM to Start, its corresponding SM must be enabled. However it is valid to enable the SM to start without enabling the PM.

4.1.4 Sm Parameters

The following tables describe parameters that can be used in the `Sm` subsection of either the `Common` or `Fm` sections.

Any parameter which can be used in the `Common.Shared` section may also be used in the `Common.Sm` or `Fm.Sm` sections.

4.1.4.1 SM Redundancy

The parameters in Table 6 control SM Failover, they complement the Priority and ElevatedPriority parameters that are used in the `Shared` section and can also be used in the `Sm` section if needed.

Table 6. Sm Redundancy Parameters

Parameter	Default Value	Description
MasterPingInterval	5	MasterPingInterval is the interval in seconds at which the secondary SM pings the master (the ping occurs using an inband SM packet).
MasterPingMaxFail	3	Number of times a secondary's ping of the master must fail before renegotiation of a new master occurs.
DbSyncInterval	15	Secondary SMs will synchronize their fabric database with the master at DbSyncInterval minutes. If set to 0, Db synchronization is disabled.

4.1.4.2 Fabric Routing

The following Routing Algorithms are supported

- shortestpath - pick shortest path and balance lids on ISLs
- fattree — shortest path with better balancing for fat tree topology
- SpineFirstRouting - option to avoid credit loops in complex fabrics with Intel® Switches. Given equal length routes, routes through chassis spine first. Hence avoids loops caused by routing via edge/leaf switches instead of spines.
- dor-updown - Direction Ordered routing with updown. This routing algorithm is for mesh and torus topologies. For torus topologies, dor-updown requires 2 VLs on Inter Switch Links and uses $2^{dimensions}$ SLs (limit of 4 dimensions).

  When a fabric disruption occurs, an additional VL and SL is used for an Up/Down style non-optimal routing around the failure (therefore, limited to 3 dimensions).

  When multiple SLs are used, applications must query for a PathRecord for every Source/Distribution pair since the SL will vary per PathRecord.

The parameters in Table 7 control Fabric Unicast Routing.

Table 7. Sm Unicast Routing Parameters

Parameter	Default Value	Description
RoutingAlgorithm	shortestpath	Selects the routing algorithm. This can be `shortestpath`, `fattree`, or `dor-updown`. See "Fabric Unicast Routing" on page 30 for more information.
SpineFirstRouting	1	A routing option applicable to shortestpath routing.See "Fabric Unicast Routing" on page 30 for more information.
Lmc	0	Lmc for LID assignment to HCAs and TCAs. 2^{Lmc} LIDs will be assigned per port.See "Fabric Unicast Routing" on page 30 for more information.
LmcE0	0	Lmc for LID assignment to Switches with an Enhanced Port 0 capability. 2^{Lmc} LIDs will be assigned per Switch Port 0 with Enhanced Port 0 capability.See "Fabric Unicast Routing" on page 30 for more information.

4.1.4.3 Mesh/Torus Topology

The `MeshTorusTopology` parameter section controls Fabric Unicast Routing for Mesh/Torus fabrics. These parameters describe how the fabric is constructed. These parameters are required when the `RoutingAlgorithm` is `dor-updown`. They are not

permitted for other Routing Algorithms. The following is the example for the configuration of a 2D Torus with two ISLs in each dimension, copied from the `ifs_fm.xml-sample` file.

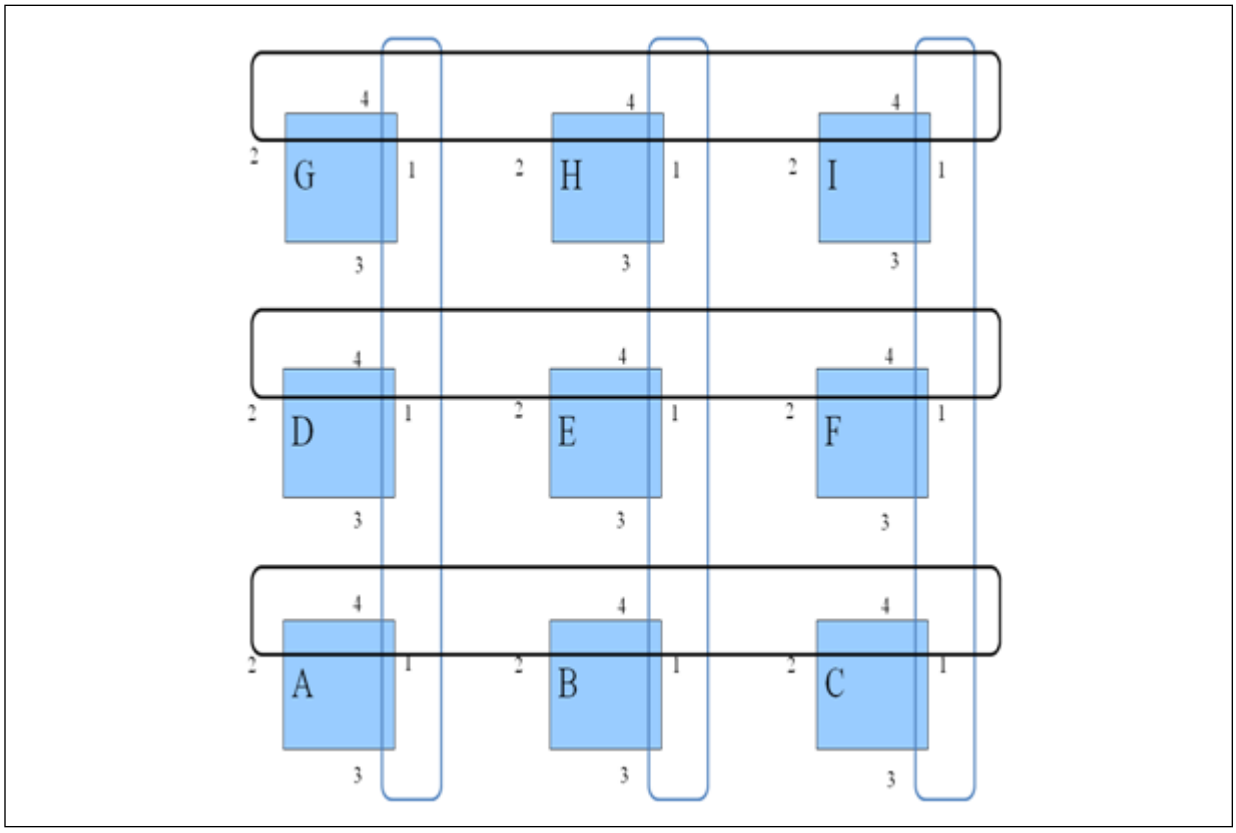
```
<!-- *************** Mesh/Torus Topology *************************** -->

    <!-- When the RoutingAlgorithm is set to dor-updown, the following -->

    <!-- section is used to configure Mesh/Torus parameters.  -->

    <!-- MeshTorusTopology:  -->

    <!--    To properly handle fabric disruptions or installation mistakes -->

    <!--    the SM needs to understand which switch ports will be used for -->

    <!--    ISLs and which dimension each ISL is associated with. -->

    <!--    The fabric must be constructed with all ISL connections in a -->

    <!--    given dimension using consistent port numbers on each switch. -->

    <!--    Also each dimension may be toroidal or non-toroidal.  -->

    <!--    For non-toroidal dimensions, the extra switch ports on the edges -->

    <!--    of the mesh may be used for additional CAs. -->

    <!--    The following Policies and Controls can be specified: -->

    <!--    Dimension: -->

    <!--        Port pair:  <PortPair>i,j</PortPair> -->

    <!--             The ports on neighbor switches which will be connected -->

    <!--             for ISLs in a given dimension. -->

    <!--             There should be 1 or more port pairs per dimension. -->

    <!--             List one port pair per neighbor switch ISL -->

    <!--        Toroidal:   <Toroidal>1</Toroidal>   -->

    <!--           Indicates that the dimension is toroidal (a closed loop) -->

    <!--        WarnThreshold:     -->

    <!--             Maximum number of warnings to be logged for each set -->

    <!--             of invalid ISL connections found. -->

    <!--             ISL connections which are in conflict with -->

    <!--             the Dimensions and PortPair definitions specified -->

    <!--             in the MeshTorusTopology are considered invalid. -->

    <!--             and will be logged and ignored. -->

    <!--             Default value is 5. Max value is 255. -->

    <!--        UpDownMcastSameSpanningTree: -->

    <!--             If set to 1, UpDown spanning tree will match the -->
```

```
<!--              multicast spanning tree. Multicast spanning tree  -->
<!--              settings can be controlled via RootSelectionAlgorithm -->
<!--              configuration option. -->
<!--              If set to 0, UpDown spanning tree will not depend -->
<!--              on multicast spanning tree settings and the root of -->
<!--              updown spanning tree will be the switch next to SM. -->
<!--              Default value is 1. -->


<!-- Example of 2D Torus with two ISLs in each dimension -->
<!-- <MeshTorusTopology>                    -->
<!--   <Dimension>                          -->
<!--      <PortPair>1,2</PortPair>          -->
<!--      <PortPair>3,4</PortPair>          -->
<!--      <Toroidal>1</Toroidal>            -->
<!--   </Dimension>                         -->
<!--   <Dimension>                          -->
<!--      <PortPair>5,6</PortPair>          -->
<!--      <PortPair>7,8</PortPair>          -->
<!--      <Toroidal>1</Toroidal>            -->
<!--   </Dimension>                         -->

<!--   <WarnThreshold>5</WarnThreshold>     -->
<!--   <UpDownMcastSameSpanningTree>1</UpDownMcastSameSpanningTree> -->
<!-- </MeshTorusTopology>                   -->
```

The `MeshTorusTopology` section describes the interswitch links which compose the topology of the fabric. All ISLs in the fabric which have not been configured in this section will be considered invalid and ignored when the `dor-updown` algorithm is selected as the routing algorithm.

In a Mesh/Torus topology the dimensions are defined by the ISLs between specific switch port numbers. In this FM Configuration example, the First Dimension (X) uses connections between port 1 and port 2 of each switch and the Second Dimension (Y) uses connections between port 3 and port 4 of each switch (refer to Figure 21). If there is more than 1 ISL between switches in a single dimension, additional `PortPair` attributes can be listed for the Dimension.

Figure 21. Example of a 2D Torus Fabric Configuration



If a dimension is toroidal, that is, the switches along the edges of the dimension are wrapped around and connected together then the dimension must be defined as Toroidal by setting the Toroidal parameter for the dimension to be 1. If a dimension is not specified as toroidal, any such wrap around links will be ignored.

The FM will generate warnings in the log file for the ISLs that it finds which are not consistent with how the topology has been defined in the `MeshTorusTopology` section of the configuration file as shown in Table 8. The `WarnThreshold` parameter can be used to specify the maximum number of warnings that should be logged for each unique invalid ISL port pair found.

If a given dimension has exactly two planes, then the PortPair can indicate identical ports are connected such as `<PortPair>3, 3<PortPair>` which indicates port 3 on one switch in the dimension is connected to port 3 on the neighbor switch in that dimension. Such dimensions cannot be toroidal.

The FM constructs a spanning tree of the switches as part of the dor-updown algorithm, and uses it as a basis for setting up the updown routes in the fabric. If UpDownMcastSpanningTree parameter is set to 1, the updown algorithm will use the same settings as the Multicast spanning tree RootSelectionAlgorithm configuration parameter (in the Multicast section of the configuration file) to build the updown spanning tree. Therefore, the updown spanning tree and the multicast spanning tree will be similar. This will make sure that the updown unicast traffic does not conflict with the multicast traffic and will prevent credit loops. Another advantage is that the syncing of spanning tree root between master FM and stand-by FM also applies to the updown

spanning tree root. If the RootSelectionAlgorithm is set to LeastTotalCost or LeastWorstCaseCost, then updown routes will remain the same on FM failover (assuming there are no other major changes in the fabric)

If UpDownMcastSameSpanningTree is set to 0, the updown spanning tree is built independent of the multicast spanning tree settings. The switch running the Embedded FM or the switch next to the HCA running the master Host FM is used as the root of the updown spanning tree. Default value of `UpDownMcastSameSpanningTree` is 1.

As a consequence of using the same spanning tree for updown and multicast, the multicast SL should match the dor-updown SL. The user can specify the SL in the MulticastGroup configuration (which is used to create the default multicast groups at startup). This SL is generally left unspecified and the SM has always used the BaseSL of the associated VirtualFabric when creating the multicast groups. Configuration checks are used to enforce use of the BaseSL. If the SL is configured for a MulticastGroup and it does not match the BaseSL of the associated VirtualFabric, an error will be logged and the group will be disabled.

The parameters in Table 8 control Mesh/Torus topology for Mesh/Torus fabrics.

Table 8. Sm Mesh/Torus Topology Parameters

Parameter	Default Value	Description
PortPair		Specifies the Pair of switch ports which will be connected between neighbor ports in the given dimension. There must be 1 or more port pairs per dimension. Each port pair defines a single ISL which will be consistently cabled between all switches in the dimension.
Toroidal	0	Indicates that the dimension is toroidal (a closed loop). Specify 1 for Toroidal dimensions and 0 for mesh dimensions.
WarnThreshold	5	Maximum number of warnings to be logged for each set of invalid inter-switch port pair connections found. Inter-switch connections which are inconsistent with the `Dimensions` and `PortPair` definitions specified in the `MeshTorusTopology` are considered invalid. Max value is 255.
UpDownMcastSameSpanningTree	1	If set to 1, UpDown spanning tree will match the multicast spanning tree. Multicast spanning tree settings can be controlled via RootSelectionAlgorithm configuration option. If set to 0, UpDown spanning tree will not depend on multicast spanning tree settings and the root of updown spanning tree will be the switch next to SM. Default value is 1. -->

4.1.4.4 Fat Tree Topology

When the `RoutingAlgorithm` is set to fattree, the `FatTreeTopology` section is used to configure fat tree parameters. To properly handle routing decisions, the SM needs to understand some fabric topology information. Specifically, how many tiers are in the fat tree and which tier a switch belongs to. If all Channel Adapters are on the same tier, the SM can easily determine the fat tree topology and tier a given switch resides in. If they are not, an attempt will be made to find the core switches based on the node description. This works well if the core switches are Intel® 12000 series switches.

```
<!-- ***************** Fat Tree Topology ***************************** -->
```

```
<!-- When the RoutingAlgorithm is set to fattree, the following -->
<!-- section is used to configure fat tree parameters.  -->
<!-- FatTreeTopology:  -->
<!--    To properly handle routing decisions, the SM need to understand -->
<!--    some fabric topology information.  Specifically, how many -->
<!--    tiers are in the fat tree and which tier a switch belongs to. -->
<!--    If all CAs are on the same tier, the SM can easily determine -->
<!--    the fat tree topology and tier a given switch resides in. -->
<!--    If they are not, an attempt  will be made to find the core switches -->
<!--    based on node description. This works well if the core switches are -->
<!--    Intel 12000 series switches. -->
<FatTreeTopology>
  <!-- The number of tiers in the fat tree -->
  <TierCount>3</TierCount>
  <!-- Discovery algorithm a bit more streamlined if CAs on same tier, -->
  <!-- but this is not a requirement. -->
  <CAsOnSameTier>1</CAsOnSameTier>
  <!-- When Systematic routing algorithm is in use, nodes may -->
  <!-- be specifed for exclusion from initial round-robin -->
  <!-- to give better route balancing of remaining nodes. -->
  <!-- This may be useful in assymetric fat trees or to -->
  <!-- initially balance across compute nodes in the tree. -->
  <!-- <NodeDesc>hca1</NodeDesc>  -->
  <!-- <NodeDesc>hca2</NodeDesc>  -->
</FatTreeTopology>
```

The parameters in Table 10 control Fat Tree Topology.

Table 9. Sm Fat Tree Topology Parameters

Parameter	Default Value	Description
TierCount	3	The number of tiers in the fat tree
CAsOnSameTier	1	Discovery algorithm a bit more streamlined if CAs on same tier, but this is not a requirement.
NodeDesc		When Systematic routing algorithm is in use, nodes may be specified for exclusion from initial round-robin to give better route balancing of remaining nodes. This may be useful in asymmetric fat trees or to initially balance across compute nodes in the tree.

4.1.4.5 InfiniBand* Technology-compliant Multicast

InfiniBand* Trade Association (IBTA) has a limitation in that the SM must make the realizable decision for a Multicast group at Multicast Join/Create time. However later fabric changes (removal of links, loss of switches) could make the multicast group unrealizable, but there is no notice in IBTA which the SM could send the end node.

To address this situation, the SM performs stricter Multicast checking at Join/Create time. This means a Multicast join/create is rejected if there are any switch to switch links which do not have at least the MTU or Rate requested for the Multicast group. The rejection reduces the chance that a simple fabric failure could make the group unrealizable.

The parameters in Table 10 control Fabric Multicast Routing. These are all part of the `Multicast` subsection within the `Sm` section.

Table 10. Sm Multicast Routing Parameters

Parameter	Default Value	Description
DisableStrictCheck	0	When 1, disables the strict checking and accepts Join/Create for which at least 1 viable fabric path exists.
MLIDTableCap	1024	Number of Multicast LIDs available in fabric. Must be set to less than or equal to the smallest Multicast forwarding table size of all switches in fabric.
MLIDShare		Describes a set of MGIDs which may share a common pool of Multicast LIDs. Such sharing permits control over conservation of Multicast LIDs. See Table 11 for the parameters that can be used within each `MLIDShare` subsection. Up to 32 `MLIDShare` subsections are allowed.
MulticastGroup		Describes a set of multicast groups which should be pre-created by the SM. See Table 11 for the parameters that can be used within each `MulticastGroup` subsection. This subsection can be used multiple times with each occurrence specifying a different set of multicast groups to be created.

Table 10. Sm Multicast Routing Parameters (Continued)

Parameter	Default Value	Description
EnablePruning	0	Should multicast spanning tree be pruned to its minimal size. When disabled (0), multicast join and leave is optimized by building the spanning tree to include all switches, such that a HCA join/leave only requires a single switch update When enabled (1), multicast traffic will only propagate through the minimal set of switches, reducing overhead in the fabric Recommend to be disabled (0) for typical fabrics where multicast is mainly used to support IPoIB ARP.
RootSelectionAlgorithm	LeastTotalCost	Controls how the root switch for the multicast spanning tree is selected. One of the following algorithm names can be specified to select the root switch for the spanning tree (Names of algorithm are not case sensitive). `LeastTotalCost` — A switch with the smallest sum of costs to other switches will be selected. `LeastWorstCaseCost` — A switch that has the least worst case cost to other switches will be selected. `SMNeighbor` — The Switch next to the SM will be selected. `LeastTotalCost` and `LeastWorstCaseCost` tend to select a switch that is more in the center of the fabric as the root of the multicast spanning tree. With these options, the information about the selected switch is also communicated to the standby SMs so that the multicast spanning tree computed by the standby SMs will match that of the master SM. That way if the master SM fails over to a standby SM, the multicast spanning tree for the fabric will still remain the same and the multicast forwarding tables in the switches do not have to be reprogrammed. The multicast traffic will not be disrupted. Since the `SMNeighbor` algorithm will select the switch next to the SM, if a master SM fails over to the standby, the spanning tree computed by the standby SM will differ from that of the master and the multicast forwarding tables in the switch will have to be reprogrammed which can result in some disruptions to the multicast traffic.
MinCostImprovement	50%	When using `LeastTotalCost` or `LeastWorstCaseCost`, the `MinCostImprovement` parameter controls when the multicast spanning tree root should be changed if there are changes in the number of switches in the fabric. The root will be changed only if in the new topology a switch's cost is `MinCostImprovement` (expressed as a percentage) better than that of the current root switch or when the current root switch goes offline. A higher value of `MinCostImprovement` would result in the multicast spanning tree root not being changed too frequently and therefore would cause less disruptions in multicast traffic. A smaller value would select the best switch and provide better multicast latency but can cause disruptions to multicast traffic upon fabric changes.

4.1.4.6 Fabric Multicast MLID Sharing

IPv6 (and possibly other applications) can create numerous Multicast groups. Each Multicast group needs an MLID.

In the case of IPv6, there is one Solicited-Node multicast group per CA port. This results in an excessively large number of multicast groups. Also in large fabrics, this quickly exceeds MLIDTableCap. To address this situation, the MLIDShare fields allow groups of Multicast GIDs to share a limited number of MLIDs. This can conserve the Hardware MLID tables so other uses of Multicast can be optimized and efficient.

the MLIDShare fields allow specification of 32 separate groups of MGIDs. Each group of MGIDs are combined into its own unique set of MLIDs. Each MGID is ANDed with MGIDMask then compared to MGIDValue. On a match the MGID will be given a MLID within a pool of MaxMLIDs.

MGIDs are 128 bits specified as two colon separated, 64-bit values, IPv6 Solicited-Node multicast, combined to 500 MLIDs

```
#SM_0_mcastGrpMGidLimitValue_0#!
```

Example of IPv6 Solicited-Node multicast combined to 500 MLIDs

```
#SM_0_mcastGrpMGidLimitValue_0#=
```

The following is an example of a MLIDShare section:

```
<MLIDShare>
        <Enable>1</Enable>
        <MGIDMask>0xffffffffffffffff:0xfffffffff000000</MGIDMask>
        <MGIDValue>0xff12601bffff0000:0x00000001ff000000</MGIDValue>
        <MaxMLIDs>500</MaxMLIDs>
</MLIDShare>
```

The parameters in Table 11control Fabric Multicast LID Sharing. These are all part of the MLIDShare subsection of the Multicast subsection within the Sm section. Each MLIDShare section defines a pool of Multicast LIDs which will be shared among all Multicast GIDs which match the given MGIDValue after being masked by the MGIDMask.

Table 11. Sm Multicast MLIDShare Parameters

Parameter	Default Value	Description
Enable	1	Enables or disables the given MLIDShare section. This provides a convenient way to disable a MLIDShare section without needing to delete it from the configuration file.
MGIDMask	0xffffffffffffffff:0xfffffffff000000	128-bit mask (represented as two 64-bit numbers separated by a colon). to apply to an MGID.
MGIDValue	0xff12601bffff0000:0x00000001ff000000	128-bit value (represented as two 64-bit numbers separated by a colon). to compare to an MGID (after the MGID is masked).
MaxMLIDs	500	Maximum number of Multicast LIDs to allocate to this shared Pool.

4.1.4.7 Pre-Created Multicast Groups

Multicast Groups which are pre-created by the SM are all part of the MulticastGroup subsection of the Multicast subsection within the Sm section. Each MulticastGroup section defines one or more Multicast Groups which will be pre-created by the Sm.

OFED requires a pre-created MulticastGroup for IPoIB. The configuration can specify other groups that are also needed. Every pre-created MulticastGroup can have one or more MGIDs. The MGID must be unique among all MulticastGroups within an FM instance.

When defined at Common level the MGID must be unique within all instances. MGIDs are specified as two 64-bit values separated by a colon (:). A single MGID can be specified as <MGID>0xabc:0x123567</MGID> in the MulticastGroup section. If no MGIDs are specified, the IPv4 broadcast, IPv4 all nodes and the IPv6 all nodes MGIDs will be created.

The following is a sample of IPoIB IPv4 and IPv6 multicast for all VFs which have IPoIB as an application

```
<MulticastGroup>

        <Create>1</Create>

        <MTU>2048</MTU>

        <Rate>10g</Rate>

        <!-- <SL>0</SL> -->

        <QKey>0x0</QKey>

        <FlowLabel>0x0</FlowLabel>

        <TClass>0x0</TClass>

</MulticastGroup>
```

The following is a sample of IPoIB, IPv4, and IPv6 multicast for 0x9001/0x1001 PKey. This can be useful if there are multiple IPoIB vFabrics and different multicast parameters (Rate, MTU, etc) are desired for each IPoIB vFabric. Since IPoIB MGID includes PKey, we specify PKey not VirtualFabric. MGIDs specified must use the Full PKey (0x8000-bit set)

```
<MulticastGroup>

        <Create>0</Create>

        <PKey>0x1001</PKey>

        <!-- PKey 0x9001/0x1001 is part of IPv4 MGID below -->

        <!-- MGID = 0xffFS401bPPPP0000:00000000GGGGGGGG -->

        <!-- where F=flags, S=scope, P=PKey and G=IP Multicast Group -->

        <MGID>0xff12401b90010000:0x00000000ffffffff</MGID> <!-- bcast -->

        <MGID>0xff12401b90010000:0x0000000000000001</MGID> <!-- all nodes -->

        <MGID>0xff12401b90010000:0x0000000000000002</MGID> <!-- all routers -->
```

```
        <MGID>0xff12401b90010000:0x0000000000000016</MGID>
        <!-- PKey 0x9001/0x1001 is part of IPv6 MGIDs below -->
        <!-- MGID = 0xffFS601bPPPPGGGG:GGGGGGGGGGGGGGGG -->
        <!-- where F=flags, S=scope, P=PKey and G=IP Multicast Group -->
        <MGID>0xff12601b90010000:0x0000000000000001</MGID> <!-- all nodes -->
        <MGID>0xff12601b90010000:0x0000000000000002</MGID> <!-- all routers -->
        <MGID>0xff12601b90010000:0x0000000000000016</MGID>
        <MTU>2048</MTU>
        <Rate>10g</Rate>
        <QKey>0x0</QKey>
        <FlowLabel>0x0</FlowLabel>
        <TClass>0x0</TClass>
</MulticastGroup>
```

The parameters in Table 12 define the policies and controls for the Multicast Groups

Table 12. Sm Multicast Group Pre-Creation Parameters

Parameter	Default Value	Description
Create	1	Enables or disables the creation of the given set of Multicast Groups. This provides a convenient way to disable a MulticastGroup section without needing to delete it from the configuration file
VirtualFabric or PKey		Controls the virtual fabric for which the `MulticastGroup` is created. Alternatively a `PKey` may be specified. If neither is specified, the MGID will be created for all Virtual Fabrics which contain the MGID as an Application
Rate or Rate_Int	10g	The Static Rate for the multicast group. Only nodes and paths which have a rate greater than tor equal to this value will be able to join the group. This also sets the upper bound for the performance of the multicast group. Rate can either be specified in natural format using `Rate` or in IBTA Int values using `Rate_Int`: 2=2.5g, 3=10g, 4=30g, 5=5g, 6=20g, 7=40g, 8=60g, 9=80g, 10=120g
MTU or MTU_Int	2048	The MTU for the multicast group. Only nodes and paths which have an MTU greater than or equal to this value will be able to join the group. This also sets the upper bound for the message sizes which may be sent to the multicast group. MTU can either be specified in natural format using `MTU` or in IBTA Int values using `MTU_In`>: 1=256, 2=512, 3=1024, 4=2048, 5=4096
SL	0	The Service Level for the Multicast Group. If unspecified, this will default to the BaseSL of the corresponding Virtual Fabric.

Table 12. Sm Multicast Group Pre-Creation Parameters (Continued)

Parameter	Default Value	Description
QKey	0	The QKey to be used for the group.
FlowLabel	0	The FlowLabel to be used for the group. This value is not presently used.
TClass	0	The Traffic Class to be used for the group. This value is not presently used.

4.1.4.8 Fabric Programming

SwitchLifetime, HoqLife, and VLStallCount can be used to relieve fabric congestion and avoid fabric deadlocks by discarding packets. Discards help prevent back pressure from propagating deep into the core of the fabric, however such discards will cause end nodes to need to timeout and retransmit.

If a packet stays at the Head of a Switch Egress Port for more than HoqLife, it is discarded. Similarly a packet queued in a switch for more than SwitchLifetime is discarded.

Specified as integer time using ns, us, ms, s, m, or h as units. SwitchLifetime and HoqLife can also be set to infinite. VLStallCount controls a second tier more aggressive discard. If VlStallCount packets in a row are discarded due to HoqLife by a given VL on an egress port. That egress port's VL enters the VL Stalled State and discards all that VL's egress packets for 8*HoqLife. Packets discarded for any of these reasons will be included in the TxDiscards counter which is queryable via FastFabric.

The parameters in Table 13 control Fabric Configuration and Programming.

Table 13. Sm Fabric Configuration Parameters

Parameter	Default Value	Description
SmKey	0	64-bit key of SM, redundant SMs must have same key
MKey	0	64-bit mkey to secure SMA ports this SM manages
SwitchLifetime	33ms	A packet queued in a switch for more than `SwitchLifetime` is discarded. Specified as integer time using ns, us, ms, s, m, or h as units. Can also be set to infinite
HoqLife	8ms	If a packet stays at the Head of a Switch Egress Port for more than HoqLife, it is discarded. Specified as integer time using ns, us, ms, s, m, or h as units. Can also be set to infinite
VlStallCount	7	Controls a second tier more aggressive discard. If `VlStallCount` packets in a row are discarded due to HoqLife by a given VL on an egress port. That egress port's VL enters the VL Stalled State and discards all that VL's egress packets for 8*`HoqLife`.
CongestionControl		Configures support for IBTA Congestion Control. For Congestion Control configuration information refer to Section 4.1.4.8.4, "Congestion Control" on page 90
CongestionControl	0	Configures support for IBTA Congestion Control
AdaptiveRouting		Configures support for AdaptiveRouting in Intel® QDR Switches. For Adaptive Routing configuration information refer to Section 4.1.4.8.5, "Adaptive Routing" on page 92
SaRespTime	1s	maximum anticipated SA response time

Table 13. Sm Fabric Configuration Parameters (Continued)

Parameter	Default Value	Description
PacketLifetime	1s	When `DynamicPacketLifetime` is disabled, the `PacketLifetime` is reported as the `PKtLifetime` for all paths
DynamicPacketLifetime		Configures support for `DynamicPacketLifetime`. For `Dynamic Packet Lifetime` configuration information refer to Table 19.
SmaBatchSize	4	Max parallel requests to a given SMA
PathSelection	Minimal	Controls PathRecord path selection and ordering. Most applications will use the first path or only the first few paths. When LMC=0 this setting makes no difference because there is only a src/dst address per pair of ports. However when LMC!=0, there can be N=(1<<LMC) addresses per port. This means there are N^2 possible combinations of SLID and DLID which the SA could return in the Path Records. However there are really only N combinations which represent distinct outbound and return paths. All other combinations are different mixtures of those N outbound and N return paths. Also important to note, is that LMC for all CAs are typically the same, while LMC for switches will be less. Generally redundant paths and/or having a variety of paths is not critical for paths to switches, but can be important for applications talking Channel Adapter to Channel Adapter Controls what combinations are returned and in what order. For the examples below, let's assume SGID LMC=1 (2 LIDs) and DGID LMC=2 (4 LIDs) `Minimal` – return no more than 1 path per lid SLID1/DLID1, SLID2/DLID2 (since SGID has 2 lids stop) `Pairwise` – cover every lid on both sides at least once SLID1/DLID1, SLID2/DLID2, SLID1/DLID3, SLID2/DLID4 `OrderAll` – cover every combination, but start with pairwise set SLID1/DLID1, SLID2/DLID2, SLID1/DLID3, SLID2/DLID4 SLID1/DLID2, SLID1/DLID4, SLID2/DLID1, SLID2/DLID3 `SrcDstAll` – cover every combination with simple all src, all dst SLID1/DLID1, SLID1/DLID2, SLID1/DLID3, SLID1/DLID4 SLID2/DLID1, SLID2/DLID2, SLID2/DLID3, SLID2/DLID4
QueryValidation	0	When 1, enables IBTA compliant SA query operation for `GetTable` (PathRecord). As such a SGID and NumbPath is required When 0, allows interop with some non-compliant OFED queries and permits `GetTable` (PathRecord) to specify either a SGID or SLID and defaults `numbPath` to 127 if not specified.
SmaBatchSize	2	This and the following two parameters control how many concurrent SMA requests the SM can have in flight while programming the SMAs in the fabric.
EhcaSmaBatchSize	1	Max parallel reqs to ehca SMA
MaxParallelReqs	4	Max devices to req in parallel

Table 13. Sm Fabric Configuration Parameters (Continued)

Parameter	Default Value	Description
SmaEnableLRDR	1	Controls use of mixed LID Routed (LR) and Directed Routed (DR) SMPs during programming of the fabric. The use of mixed LR-DR SMPs can improve SM sweep time and reduce the time for SM to respond to fabric changes. All Intel® HCAs and Switches support LR-DR SMPs. It can have the following values 0 – SM will not use any mixed LR-DR SMPs 1 – SM will use mixed LR-DR SMPs. 2 – SM will attempt using mixed LR-DR SMPs, but if a particular switch does not respond to these SMPs, it will fallback to using pure DR SMPs for initializing nodes connected to that switch.
EhcaSmaEnableLRDR	0	Controls use of mixed LID Routed (LR) and Directed Route (DR) SMPs during programming of IBM eHCA Logical Switches 0 – never send a LRDR SMP to an IBM eHCA Logical switch 1 – enable use of LRDR SMPs for IBM eHCA Logical switches. This only has effect when SmaEnableLRDR is 1 or 2.
LinkSpeedOverride	0	When enabled (1), the SM will adjust the Link Speed of devices which come up in SDR but support Link Speeds of DDR or QDR on both sides of the link. When disabled (0), the SM will not alter link speeds. This feature is used to handle some DDR or QDR devices whose links come up at SDR when connected to newer InfiniBand* Technology-compliant devices.

4.1.4.8.1 Congestion Control

The parameters in Table 14 control Congestion Control configuration for the Switch and Channel Adapter parameters in the fabric. The following is a sample of the Adaptive Routing section of the FM configuration file.

```
<CongestionControl>

      <!-- 1 = Enable, 0 = Disable -->

      <Enable>0</Enable>

      <!-- Turn on additional debug logging for CCA -->

      <!-- 1 = Enable, 0 = Disable -->

      <Debug>0</Debug>

      <!-- Number of attempts made when discovering whether a node -->

      <!-- supports Congestion Control.  Increasing this value increases -->

      <!-- resiliency, but also increases the timeout against nodes that -->

      <!-- do not support Contestion Control. -->

      <DiscoveryAttempts>3</DiscoveryAttempts>

      <!-- This setting allows discovery to occur regardless of whether -->

      <!-- CC is enabled or disabled thus allowing CC to be cleared in a -->

      <!-- topology after being disabled -->
```

```
        <!-- 1 = Enable, 0 = Disable -->
        <DiscoverAlways>0</DiscoverAlways>
        <!-- CC settings applicable to all switches. -->
        <!-- Default settings for switch are selected to provide the best -->
        <!-- congestion control for various traffic mixes but may require -->
        <!-- tuning for best performance -->
        <Switch>
          <!-- IBTA CC SwitchCongestionSetting:Threshold -->
          <!-- A value in the range 0 to 15.  Higher values indicate more -->
          <!-- aggressive congestion thresholds. -->
          <Threshold>2</Threshold>
          <!-- IBTA CC SwitchCongestionSetting:Packet_Size -->
          <!-- Minimum packet size to mark in units of credits. -->
          <PacketSize>0</PacketSize>
          <!-- IBTA CC SwitchCongestionSetting:CS_Threshold -->
          <!-- A value in the range 0 to 15.  Higher values indicate more -->
          <!-- aggressive credit starvation thresholds. -->
          <CsThreshold>0</CsThreshold>
          <!-- IBTA CC SwitchCongestionSetting:CS_ReturnDelay -->
          <CsReturnDelay>0</CsReturnDelay>
          <!-- IBTA CC SwitchCongestionSetting:Marking_Rate -->
          <MarkingRate>0</MarkingRate>
          <!-- IBTA CC SwitchCongestionSetting:Control_Map bit0 indicates that -->
          <!-- the victim mask is valid-->
          <!-- A value of 0 will cause marking of only sources of congestion -->
          <!-- A value of 1 will cause marking of both sources and victims of
congestion -->
          <VictimMarkingEnable>1</VictimMarkingEnable>
        </Switch>
        <!-- CC settings applicable to all channel adapters. -->
        <!-- When using Intel HCAs, the settings for Congestion Control can -->
        <!-- be configured with the following environment variables such as -->
        <!-- the ones shown here and may require tuning for best performance -->
        <!-- PSM_DISABLE_CCA 0 -->
```

```
    <!-- PSM_CCTI_INCREMENT 1 -->
    <!-- PSM_CCTI_TIMER 1 -->
    <!-- PSM_CCTI_TABLE_SIZE 128 -->
    <Ca>
      <!-- IBTA CC CACongestionSetting:Port_Control -->
      <!-- Determines whether congestion control is handled per-QP or -->
      <!-- per-SL.  Valid values are "qp" or "sl". -->
      <Basis>qp</Basis>
      <!-- IBTA CC CACongestionEntry:CCTI_Increase -->
      <Increase>5</Increase>
      <!-- IBTA CC CACongestionEntry:CCTI_Timer -->
      <Timer>10</Timer>
      <!-- IBTA CC CACongestionEntry:Trigger_Threshold -->
      <Threshold>8</Threshold>
      <!-- IBTA CC CACongestionEntry:CCTI_Min -->
      <Min>0</Min>
      <!-- IBTA CC CongestionControlTable:CCTI_Limit -->
      <Limit>128</Limit>
      <!-- Maximum injection rate delay in us, used to determine how the -->
      <!-- congestion control table entries are generated. -->
      <DesiredMaxDelay>21</DesiredMaxDelay>
    </Ca>
  </CongestionControl>
```

Table 14. Congestion Control Parameters

Parameter	Default Value	Description
Enable	0	Enables (1) or disables (0) Congestion Control.
Debug	0	Enables (1) or disables (0) additional debug logging for CCA
DiscoveryAttempts	3	The number of attempts made when discovering whether a node supports Congestion Control. Increasing this value increases resiliency, but also increases the timeout against nodes that do not support Congestion Control.

Table 14. Congestion Control Parameters (Continued)

Parameter	Default Value	Description
DiscoverAlways	0	Allows discovery to occur regardless of whether CC is enabled or disabled thus allowing CC to be cleared in a topology after being disabled. 1 = Enable 0 = Disable
Switch	N/A	CC settings applicable to all switches. Default settings for switch are selected to provide the best congestion control for various traffic mixes but may require tuning for best performance. Refer to Section 4.1.4.8.2, "Congestion Control Switch Settings" on page 89 for the switch parameters.
Ca	N/A	CC settings applicable to third party channel adapters. When using Intel® HCAs, the settings for Congestion Control can be configured with the following environment variables such as the ones shown here and may require tuning for best performance PSM_DISABLE_CCA 0 PSM_CCTI_INCREMENT 1 PSM_CCTI_TIMER 1 PSM_CCTI_TABLE_SIZE 128

4.1.4.8.2 Congestion Control Switch Settings

The parameters in Table 15 control Congestion Control settings applicable to all switches. Default settings for a switch are selected to provide the best congestion control for various traffic mixes but may require tuning for best performance.

Table 15. Congestion Control Switch Settings Parameters

Parameter	Default Value	Description
Threshold	2	A value in the range 0 to 15. Higher values indicate more aggressive congestion thresholds.
PacketSize	0	Minimum packet size to mark in units of credits.
CsThreshold	3	A value in the range 0 to 15. Higher values indicate more aggressive credit starvation thresholds.
CsReturnDelay	0	The return delay for credit starvation.
MarkingRate	0	The marking rate for switch congestion
VictimMarkingEnable	1	Control_Map bit0 indicates that the victim mask is valid. A value of 0 will cause marking of only sources of congestion A value of 1 will cause marking of both sources and victims of congestion

4.1.4.8.3 Congestion Control Channel Adapter Settings

Congestion Control settings are applicable to third party channel adapters. When using Intel® HCAs, the settings for Congestion Control can be configured with the following environment variables and may require tuning for best performance:

```
PSM_DISABLE_CCA 0

PSM_CCTI_INCREMENT 1

PSM_CCTI_TIMER 1
```

```
PSM_CCTI_TABLE_SIZE 128
```

The parameters in Table 16 control Congestion Control settings applicable to third party Channel Adapters. Default settings for a Channel Adapter are selected to provide the best congestion control for various traffic mixes but may require tuning for best performance.

Table 16. Congestion Control Channel Adapters Settings Parameters

Parameter	Default Value	Description
Basis	qp	Determines whether congestion control is handled per-Queue Pair (QP) or per-Service Level (SL). Valid values are `qp` or `sl`.
Increase	5	Channel Adapter Congestion Control Table Index (CCTI) Increase. The amount by which a CCTI will be increased, when a packet which was marked as having gone through a point of congestion, is received.
Timer	10	Channel Adapter CCTI Timer that is a cyclic timer set up by the congestion manager, associated with an SL, which on expiration is reset and decreases the CCTI
Threshold	8	Channel Adapter Trigger Threshold When a CCTI is equal to the Trigger Threshold this event is triggered and information pertaining to the event is logged by the Channel Adapter
Min	0	Channel Adapter CCTI minimum. This is the lowest value that CCTI can be reduced to; the default value is zero.
Limit	128	CCTI_Limit is the bounding value for a CCTI. CCTI cannot be greater than CCTI_Limit.
DesiredMaxDelay	21	Maximum injection rate delay in us, used to determine how the congestion control table entries are generated.

4.1.4.8.4 Congestion Control

The parameters in Table 18 control IBTA Congestion Control configuration for the switches and Channel Adapters in the fabric. The following is a sample of the Congestion Control section of the FM configuration file.

```
<CongestionControl>

<Enable>0</Enable>

<DiscoveryAttempts>3</DiscoveryAttempts>

<Switch>

<Threshold>8</Threshold>

<PacketSize>0</PacketSize>

<CsThreshold>0</CsThreshold>

<CsReturnDelay>0</CsReturnDelay>

<MarkingRate>0</MarkingRate>

<VictimMarkingEnable>0</VictimMarkingEnable>
```

```
</Switch>

<Ca>

<Basis>qp</Basis>

<Increase>5</Increase>

<Timer>10</Timer>

<Threshold>8</Threshold>

<Min>0</Min>

<Limit>128</Limit>

<DesiredMaxDelay>21</DesiredMaxDelay>

</Ca>

</CongestionControl>
```

Table 17. Sm Congestion Control Parameters

Parameter	Default Value	Description
Enable	0	1 = Enable `Congestion Control` 0 = Disable `Congestion Control`
DiscoveryAttempts	3	A number of attempts are made when discovering whether a node supports Congestion Control. Increasing this value increases resiliency, but also increases the timeout against nodes that do not support Congestion Control.
Switch		Section for CongestionControl settings applicable to all switches. The parameters for the switches are shown in the following six rows.
Switch.Threshold	8	A value in the range 0 to 15. Higher values indicate more aggressive congestion thresholds.
Switch.PacketSize	0	Minimum packet size to mark in units of credits.
Switch.CsThreshold	0	A value in the range 0 to 15. Higher values indicate more aggressive credit starvation thresholds.
Switch.CsReturnDelay	0	The time to wait before returning the packets.
Switch.MarkingRate	0	The rate for marking both sources and victims.
Switch.VictimMarkingEnable	0	`Control_Map` bit0 indicates that the victim mask is valid. A value of 0 will cause marking of only sources of congestion. A value of 1 will cause marking of both sources and victims of congestion.
Ca		Section for `CongestionControl` settings applicable to all Channel Adapters. The parameters for the Channel Adapters are shown in the following seven rows.
Ca.Basis	qp	Determines whether congestion control is handled per-Queue Pair (QP) or per-SL. Valid values are “qp” or “sl”.
Ca.Increase	5	The amount by which a CCTI will be increased, when a packet which was marked as having gone through a point of congestion, is received. See “Annex A10 “of InfiniBand* Architecture Specification Release 1.2.1, *Volume 1*.

Table 17. Sm Congestion Control Parameters (Continued)

Parameter	Default Value	Description
Ca.Timer	10	When the timer expires it will be reset to its specified value, and the CCTI will be decremented by one. See "Annex A10" of InfiniBand* Architecture Specification Release 1.2.1, *Volume 1.*
Ca.Threshold	8	When the CCTI is equal to this value, an event is logged in the CA's cyclic event log. See "Annex A10" of InfiniBand* Architecture Specification Release 1.2.1, *Volume 1.*
Ca.Min	0	This is the lowest value that CCTI can be reduced to; the default value is zero. See "Annex A10" of InfiniBand* Architecture Specification Release 1.2.1, *Volume 1.*
Ca.Limit	128	CCTI_Limit is the bounding value for a CCTI; CCTI cannot be greater than CCTI_Limit. See "Annex A10" of *InfiniBand* Architecture Specification Release 1.2.1, Volume 1*.
Ca.DesiredMaxDelay	21	Maximum injection rate delay in us, used to determine how the congestion control table entries are generated.

4.1.4.8.5 Adaptive Routing

The parameters in Table 18 control Adaptive Routing configuration for the Intel® QDR Switches in the fabric. The following is a sample of the Adaptive Routing section of the FM configuration file.

```
<AdaptiveRouting>

      <!-- 1 = Enable, 0 = Disable -->

      <Enable>0</Enable>

      <!-- When set, only adjust routes when they are lost. -->

      <!-- If not set, adjust routes when they are lost and -->

      <!-- when congestion is indicated. -->

      <LostRouteOnly>0</LostRouteOnly>

      <!-- The topology is a pure fat tree.  Do maximum amount of -->

      <!-- adaptive routing based on this topology. -->

      <Tier1FatTree>0</Tier1FatTree>

</AdaptiveRouting>
```

Table 18. Sm Adaptive Routing Parameters

Parameter	Default Value	Description
Enable	0	Enables (1) or disables (0) Adaptive Routing.
LostRouteOnly	0	When set (1), only adjust routes when they are lost. If not set (0), adjust routes when they are lost or when congestion is detected
Tier1FatTree	0	When set (1), the topology is a pure fat tree. Do maximum amount of adaptive routing based on this topology.

4.1.4.8.6 Dynamic Packet Lifetime

`DynamicPacketLifetime` and `PacketLifetime` controls the `PktLifetime` reported in PathRecord queries. `PktLifetime` is used by IBTA compliant applications to set the Queue Pair timeout and retry intervals. Per the IBTA algorithm the Queue Pair timeout will typically be two-times or four-times these values. When `DynamicPacketLifetime` is enabled, the `PktLifetime` reported will depend on number of switch hops. `Hops01` is `PktLifetime` for one-hop paths, `Hops02` is two-hop paths, etc. When `DynamicPacketLifetime` is disabled, the `PacketLifetime` is reported as the `PKtLifetime` for all paths. The following is a sample of the Dynamic Packet Lifetime section of the FM configuration file.

```
<PacketLifetime>1s</PacketLifetime>

<DynamicPacketLifetime>

      <Enable>1</Enable>

      <Hops01>67ms</Hops01>

      <Hops02>134ms</Hops02>

      <Hops03>134ms</Hops03>

      <Hops04>268ms</Hops04>

      <Hops05>268ms</Hops05>

      <Hops06>268ms</Hops06>

      <Hops07>268ms</Hops07>

      <Hops08>536ms</Hops08>

      <Hops09>536ms</Hops09>

</DynamicPacketLifetime>
```

The parameters in Table 19 define Packet Lifetime values based on HopCount of the path.

Table 19. Sm DynamicPacketLifetime Parameters

Parameter	Default Value	Description
Enable	1	Enables or disables the use of dynamic packet lifetime. This provides a convenient way to disable this section without needing to delete it from the configuration file
Hops01	67ms	Packet Lifetime for one-hop paths
Hops02	134ms	Packet Lifetime for two- hop paths
Hops03	134ms	Packet Lifetime for three-hop paths
Hops04	268ms	Packet Lifetime for four-hop paths
Hops05	268ms	Packet Lifetime for five-hop paths
Hops06	268ms	Packet Lifetime for six-hop paths
Hops07	268ms	Packet Lifetime for seven-hop paths
Hops08	536ms	Packet Lifetime for eight-hop paths
Hops09	536ms	Packet Lifetime for nine or more hop paths

4.1.4.9 Fabric Sweep

The Fabric sweep is the process by which the SM discovers fabric changes and then reprograms the parts of the fabric affected by the changes. The SM sweeps immediately upon fabric changes based on traps from the switches. Since traps can be lost, the SM also has a slow periodic sweep at `SweepInterval` to verify fabric configuration. The following is a sample of the Fabric Sweep section of the FM configuration file.

```
<!-- ***************** Fabric Sweep ***************************** -->

<!-- The SM sweeps immediately upon fabric changes based on traps from -->

<!-- the switches.  Since traps can be lost, the SM also has a slow -->

<!-- periodic sweep at SweepInterval to verify fabric config. -->

<SweepInterval>300</SweepInterval> <!-- max seconds between sweeps -->

<IgnoreTraps>0</IgnoreTraps> <!-- don't sweep nor log when traps occur -->

<!-- The SM waits up to RespTimeout milliseconds for responses. -->

<!-- Upon a timeout, up to MaxAttempts are attempted for a given request -->

<MaxAttempts>3</MaxAttempts>

<RespTimeout>250</RespTimeout> <!-- in milliseconds -->

<!-- SM will start with MinRespTimeout as the timeout value for requests -->

<!-- and use multiples of this value for subsequent attempts if there is -->

<!-- a timeout in the previous attempt. SM will keep retrying till -->

<!-- cumulative sum of timeouts for retries is less than -->

<!-- RespTimeout multiplied by MaxAttempts. -->

<!-- If MinRespTimeout is set to 0, upon timeout, up to MaxAttempts -->

<!-- are attempted with each attempt having a timeout of RespTimeout -->
```

```
<MinRespTimeout>35</MinRespTimeout> <!-- in milliseconds -->

<!-- When there are a large number of fabric changes at once, the SM -->
<!-- could have lots of errors while attempting to access/program -->
<!-- devices which disappeared mid-sweep.  If the SM has more than -->
<!-- SweepErrorsThreshold in a given sweep, it will give up and start -->
<!-- the sweep over.  Should SweepAbandonThreshold sweeps fail in a row -->
<!-- the SM will will do its best to complete the sweep as is. -->
<SweepErrorsThreshold>0</SweepErrorsThreshold>
<SweepAbandonThreshold>3</SweepAbandonThreshold>
<!-- If a given port issues more than TrapThreshold traps/minute -->
<!-- it will be disabled as an unstable port.  0 disables this feature. -->
<!-- The traps managed by this threshold are Traps 129-131. -->
<!-- These include traps for port Local Link Integrity threshold, -->
<!-- Excessive Buffer Overrun, and Flow Control Update watchdog timer. -->
<!-- Valid values to enable this feature are 10-100. -->
<TrapThreshold>0</TrapThreshold>
<!-- TrapThresholdMinCount is minimum number of traps required -->
<!-- to consider that TrapThreshold rate has been been reached. -->
<!-- For example if TrapThreshold is set to 10 traps/minute and -->
<!-- TrapThresholdMinCount is set to 5, the port will be disabled  -->
<!-- after 5 traps are received at the rate of 10 traps per minute -->
<!-- i.e. 5 traps in 30 seconds. -->
<!-- This value must be greater than 2. Default is 10. Ignored if -->
<!-- TrapThreshold is set to 0.-->
<!-- Larger values will increase accuracy of detecting trap rate -->
<!-- but also increase the time between a trap surge and the SM -->
<!-- disabling a port. Very small values can lead to a port being -->
<!-- disabled just after a few traps. -->
<TrapThresholdMinCount>10</TrapThresholdMinCount>
<!-- When Suppress1x is enabled, the SM will not Activate 1x links and -->
<!-- will take Down any Active 1x links found -->
<Suppress1x>0</Suppress1x>
```

```
<!-- Multicast join/create/deletes result in a SM sweeps. -->
<!-- A pathological node can cause a denial of service attack by  -->
<!-- execessive creates and deletes of multicast groups.  The -->
<!-- following limits the number of multicast Set/Delete sequences -->
<!-- that a node can issue before the SM takes action to quiet  -->
<!-- the node.-->
<!-- The number of deletes allowed within a given interval is limited -->
<!-- by the following parameters.  -->
<!-- When McDosThreshold is zero, monitoring of MC DOS is disabled. -->
<McDosThreshold>0</McDosThreshold>
<!-- Default interval is 60 seconds.  -->
<McDosInterval>60</McDosInterval>
<!-- McDosAction, the action to take if MC DOS is suspected. -->
<!--    0 = Port will be disabled.  -->
<!--    1 = Port will be bounced.  -->
<McDosAction>0</McDosAction>

<!-- Sometimes when a node is under heavy load, it may fail to respond -->
<!-- to SMA queries for a while.  In order to prevent the SM from -->
<!-- dropping such nodes from the fabric, the SM will allow a node to be -->
<!-- non responsive for up to NonRespMaxCount sweeps and NonRespTimeout -->
<!-- seconds before dropping it from the fabric. -->
<NonRespTimeout>600</NonRespTimeout> <!-- in seconds -->
<NonRespMaxCount>3</NonRespMaxCount>
```

The parameters in Table 20 control Fabric Sweep.

Table 20. Sm Fabric Sweep Parameters

Parameter	Default Value	Description
SweepInterval	300	The SM sweeps immediately upon fabric changes based on traps from the switches. Since traps can be lost, the SM also has a slow periodic sweep at a maximum of SweepInterval seconds to verify fabric configuration.
IgnoreTraps	0	Do not sweep or log when traps occur when enabled (1)
MaxAttempts RespTimeout MinRespTimeout	3 250 35	The SM will spend up to `RespTimeout` multiplied by `MaxAttempts` per packet. These allow two modes of operation. When `MinRespTimeout` is non-zero: the SM will start with `MinRespTimeout` as the time-out value for requests and use multiples of this value for subsequent attempts if there is a time-out in the previous attempt. SM will keep retrying until the cumulative sum of time-outs for retries is less than `RespTimeout` multiplied by `MaxAttempts`. This approach is recommended and will react quickly to lost packets while still allowing adequate time for slower SMAs to respond. When `MinRespTimeout` is zero: upon a time-out, up to `MaxAttempts` are attempted with each attempt having a time-out of `RespTimeout`. This approach is provided for backward compatibility with previous SM versions.
SweepErrorsThreshold SweepAbandonThreshold	0 3	When there are a large number of fabric changes at once, the SM could have lots of errors while attempting to access/ program devices which disappeared mid-sweep. If the SM receives more than the set SweepErrorsThreshold parameter of errors in a given sweep, it will give up and start the sweep over. When the SM abandons more than the set `SweepAbandonThreshold` parameter of sweeps in a row the SM will do its best to complete the sweep.
TrapThreshold	0	If a given port issues more than the `TrapThreshold` traps/ minute value it will be disabled as an unstable port. 0 disables this feature.
TrapThresholdMinCount	10	Minimum number of traps required to reach the the TrapThreshold rate. For example, if TrapThreshold is set to 10 traps per minute and TrapThresholdMinCount is set to 5, the port will be disabled after 5 traps are received at the rate of 10 traps per minute (for example: 5 traps in 30 seconds). This parameter value must be greater than 2, and the default parameter value is 10. This parameter is ignored if TrapThreshold is set to 0. Larger values increase the accuracy of detecting the trap rate, but also increase the time between a trap surge and the SM disabling a port. Very small values can result in a port being disabled after a few traps.
Suppress1x	0	When `Suppress1x` is enabled, the SM will not activate 1x links and will take down any active 1x links found
McDosThreshold	0	Multicast Denial of Service Threshold (McDos) A pathological node can cause a denial of service attack by excessive creates and deletes of multicast groups. The following limits the number of multicast Set/Delete sequences that a node can issue before the SM takes action to quiet the node. The number of deletes allowed within a given interval is limited by the following parameters. When McDosThreshold is zero, monitoring of MC DOS is disabled.

Table 20. Sm Fabric Sweep Parameters (Continued)

Parameter	Default Value	Description
McDosInterval	60	McDos Interval The interval in seconds for the number of deletes allowed. Default is 60.
McDosAction	0	McDos Action The action to take if McDos is suspected. 0 = Port will be disabled. 1 = Port will be bounced.
NonRespTimeout NonRespMaxCount	600 3	When a node is under heavy load, it may fail to respond to SMA queries for a while. In order to prevent the SM from dropping such nodes from the fabric, the SM will allow a node to be non responsive for up to `NonRespMaxCount` sweeps and `NonRespTimeout` seconds before dropping it from the fabric.

4.1.4.10 SM Logging and Debug

When nodes appear or disappear from the fabric, a message is logged. The SM Logging/Debug section defines how many messages are logged, and if additional SM sweep and SA query debug information is logged. The following is a sample of the SM Logging/Debug section of the FM configuration file.

```
<!-- ***************** SM Logging/Debug ************************** -->

<!-- When nodes appear or disappear from the fabric, a message is logged -->

<!-- This can set a threshold on how many such messages to output per -->

<!-- sweep.  Once NodeAppearanceMsgThreshold messages are logged in a -->

<!-- given sweep, the remainder are output at a lower log level (INFO) -->

<!-- Hence avoiding excessive log messages when significant -->

<!-- fabric changes occur.  0 means no limit. -->

<NodeAppearanceMsgThreshold>100</NodeAppearanceMsgThreshold>

<SmPerfDebug>0</SmPerfDebug> <!-- log additional SM sweep info -->

<SaPerfDebug>0</SaPerfDebug> <!-- log additional SA query info -->
```

The parameters in Table 21 control SM Logging.

Table 21. Sm Logging and Debug Parameters

Parameter	Default Value	Description
NodeAppearanceMsgThreshold	100	This can set a threshold on how many log messages to output per sweep. Once `NodeAppearanceMsgThreshold` messages are logged in a given sweep, the remainder are output at a lower log level (INFO) therefore avoiding excessive log messages when significant fabric changes occur. `0` means no limit.
SmPerfDebug	0	If `1`, then log additional SM sweep info.
SaPerfDebug	0	If `1`, then log additional SA query info. Log additional SA query info `SM_0_sa_debug_perf:dec` • `<Debug>0</Debug>` #SM_0_debug:dec • `<RmppDebug>0</RmppDebug>` #SM_0_sa_debug_rmpp:dec

4.1.4.11 Miscellaneous

In addition the SM supports the parameters described in the following sections.

4.1.4.11.1 LID

LID can be set for this SM as shown in Table 22

Table 22. Additional Sm Parameters

Parameter	Default Value	Description
LID	0	LID for this SM, 0=pick any available. 0 is recommended.

4.1.4.11.2 Overrides of the Common.Shared parameters

The Common.Shared parameters can be overridden in the SM using the parameters described in Table 23

Table 23. Additional Sm Parameters

Parameter	Default Value	Description
Priority	0	0 to 15, higher wins.
ElevatedPriority		0 to 15, higher wins.
LogLevel	2	***Note:*** Overrides the Common.Shared LogLevel Settings Sets log level option for SM: • 0 = disable vast majority of logging output. • 1 = fatal, error, warn (syslog CRIT, ERR, WARN). • 2 = +notice, INFIIINFO (progress messages) (syslog NOTICE, INFO). • 3 = +INFO (syslog DEBUG). • 4 = +VERBOSE and some packet data (syslog DEBUG). • 5 = +debug trace info (syslog DEBUG) This parameter is ignored for the Embedded FM. Refer to the *Intel® True Scale Fabric Switches 12000 Series User Guide* for information on configuring chassis logging options.

Table 23. Additional Sm Parameters (Continued)

Parameter	Default Value	Description
LogFile		***Note:*** Overrides Common.Shared setting. Sets log output location for SM. By default (or if this parameter is empty) log output is accomplished using syslog. However, if a LogFile is specified, logging will be done to the given file. LogMode further controls logging. This parameter is ignored for the Intel® Embedded FM. Refer to the *Intel® True Scale Fabric Switches 12000 Series User Guide* for information on configuring chassis logging options.
SyslogFacility	Local6	***Note:*** **O**verrides Common.Shared setting. For the Host FM, controls what syslog facility code is used for log messages, Allowed values are: auth, authpriv, cron, daemon, ftp, kern, local0-local7, lpr, mail, news, syslog, user, or uucp. For the Embedded FM, this parameter is ignored
ConfigConsistencyCheckLevel	2	Controls the Configuration Consistency Check for SM. If specified for an individual instance of SM, will override Shared settings. Checking can be completely disabled, or can be set to take action by deactivating Standby SM if configuration does not pass the consistency check criteria. • 0 = disable Configuration Consistency Checking • 1 = enable Configuration Consistency Checking without taking action (only log a message) • 2= enable Configuration Consistency Checking and take action (log message and move standby to inactive state)
ConfigConsistencyCheckMethod	0	Controls the checksum generation method for Configuration Consistency Checking between redundant FMs. Will override Shared settings if specified per instance. Checking MD5 cannot be used when a Host FM and Embedded FM are being used as a redundant pair. In which case the simple additive checksum must be used. The simple additive checksum mechanism may fail to detect some configuration differences. • 0 = use MD5 checksum method • 1 = use simple additive checksum method

4.1.4.11.3 Additional Parameters for Debug and Development

The SM supports the parameters in Table 24 to aid diagnosis and debug. Only use these parameters under the direction of your support representative.

Table 24. Sm Debug Parameters

Parameter	Default Value	Description
LoopTestOn	0	Cable loop test enable. When SM starts, if LoopTestOn is set to 1 a loop test will be started in normal mode. The LoopTestPackets setting will specify how many packets will be injected into the loop test when started. LoopTest is a good way to validate ISL's between switches in a cluster. By default Loop test runs in default mode. In the default mode, the SM uses an exhaustive approach to setup loop routes and will include each ISL in as many loops as possible. This ensures that each ISL is exactly in the same number of loops and hence will see the same amount of utilization. But finding all possible loops is computationally intensive and can take a long amount of time.
LoopTestFastMode	0	Puts LoopTest in fast mode when set to 1, when started based on LoopTestOn setting. Under this mode, loop test doesn't attempt to include each ISL in all possible loops, but includes it in at least the specified number of loops (this value is controlled via the `MinISLRedundancy` parameter). In typical fast mode operations (with the default `MinISLRedundancy` of 4), injecting four packets into each loop is sufficient to get a high utilization on the ISLs.
LoopTestPackets	0	Number of packets to inject. If the XML tag is not set or commented out in the XML configuration file but loop test has been enabled in the configuration file, then a loop test will start but since the packet count is zero, no packets will be injected into the loops for testing. Using one of the inject CLI commands the user can manually inject packets.
LIDSpacing	0	Spacing of LIDs to test LFT
SaRmppChecksum	0	RMPP internal checksum

Table 24. Sm Debug Parameters (Continued)

Parameter	Default Value	Description
DynamicPortAlloc	1	This parameter is for development use only.
TrapLogSuppressTriggerInterval	30	If traps are received from the same port within the interval (in seconds), logging of traps from that port will be suppressed. Log suppression is disabled if set to 0.
CS_LogMask	0x00000000	Alternative to use of LogLevel. For advanced users, these parameters can provide more precise control over per subsystem logging. For typical configurations these should be omitted and the `LogLevel` parameter should be used instead. For each subsystem there can be a `LogMask`. The mask selects severities of log messages to enable and is a sum of the following values: 0x1=fatal 0x2=actionable error 0x4=actionable warning 0x8=actionable notice 0x10=actionable info 0x20=error 0x40=warn 0x80=notice 0x100=progress 0x200=info 0x400=verbose 0x800=data 0x1000=debug1 0x2000=debug2 0x4000=debug3 0x8000=debug4 0x10000=func call 0x20000=func args 0x40000=func exit For embedded FM corresponding Chassis Logging must also be enabled and Sm configuration applies to all managers For Host FM, the linux syslog service will need to have an appropriate level of logging enabled.
MAI_LogMask	0x000001ff	
CAL_LogMask	0x000001ff	
DVR_LogMask	0x000001ff	
IF3_LogMask	0x000001ff	
SM_LogMask	0x000001ff	
SA_LogMask	0x000001ff	
PM_LogMask	0x000001ff	
PA_LogMask	0x000001ff	
BM_LogMask	0x000001ff	
FE_LogMask	0x000001ff	
APP_LogMask	0x000001ff	

4.1.5 Fe Parameters

The following tables describe parameters that can be used in the `Sm` subsection of either the `Common` or `Fm` sections.

Any parameter that can be used in the `Common.Shared` section can also be used in the `Common.Fe` or `Fm.Fe` sections.

4.1.5.1 Overrides of the Common.Shared parameters

The Common.Shared parameters can be overridden in the Fe using the parameters described in Table 25

Table 25. Additional Fe Parameters

Parameter	Default Value	Description
LogLevel	2	***Note:*** Overrides the Common LogLevel Settings Sets log level option for FE: • 0 = disable vast majority of logging output • 1 = fatal, error, warn (syslog CRIT, ERR, WARN) • 2 = +notice, INFIIINFO (progress messages) (syslog NOTICE, INFO) • 3 = +INFO (syslog DEBUG) • 4 = +VERBOSE and some packet data (syslog DEBUG) • 5 = +debug trace info (syslog DEBUG) This parameter is ignored for the Embedded FM. Refer to the *Intel® True Scale Fabric Switches 12000 Series User Guide* for information on configuring chassis logging options.
LogFile		***Note:*** Overrides Common.Shared setting. Sets log output location for FE. By default (or if this parameter is empty) log output is accomplished using syslog. However, if a LogFile is specified, logging will be done to the given file. LogMode further controls logging. This parameter is ignored for the Intel® Embedded FM. Refer to the *Intel® True Scale Fabric Switches 12000 Series User Guide* for information on configuring chassis logging options.
SyslogFacility	Local6	***Note:*** Overrides Common.Shared setting. For the Host FM, controls what syslog facility code is used for log messages, Allowed values are: auth, authpriv, cron, daemon, ftp, kern, local0-local7, lpr, mail, news, syslog, user, or uucp. For the Embedded FM, this parameter is ignored
ConfigConsistencyCheckMethod	0	Only the ConfigConsistencyCheckMethod can be changed for Fe.
DefaultPKey	0xffff	Overrides setting from Common.Shared for FE. The PKey for FE. This should be the Default PKey so they can manage all nodes.

4.1.5.2 Additional Parameters for Debug and Development

The Fe supports the parameters in Table 26 to aid diagnosis and debug. Only use these parameters under the direction of your support representative.

Table 26. Fe Debug Parameters

Parameter	Default Value	Description
Debug	0	***Note:*** Overrides Debug setting from `Common.Shared` Additional parameters for debug/development use - This enables debugging modes for FE.
RmppDebug	0	***Note:*** Overrides RmppDebug setting from `Common.Shared` If 1, then log additional FE info with regards to RMPP or the Reliable Message Passing Protocol.
CS_LogMask MAI_LogMask CAL_LogMask DVR_LogMask IF3_LogMask SM_LogMask SA_LogMask PM_LogMask PA_LogMask BM_LogMask FE_LogMask APP_LogMask	0x00000000 0x000001ff 0x000001ff 0x000001ff 0x000001ff 0x000001ff 0x000001ff 0x000001ff 0x000001ff 0x000001ff 0x000001ff 0x000001ff	Alternative to use of LogLevel. For advanced users, these parameters can provide more precise control over per subsystem logging. For typical configurations these should be omitted and the `LogLevel` parameter should be used instead. For each subsystem there can be a `LogMask`. The mask selects severities of log messages to enable and is a sum of the following values: 0x1=fatal 0x2=actionable error 0x4=actionable warning 0x8=actionable notice 0x10=actionable info 0x20=error 0x40=warn 0x80=notice 0x100=progress 0x200=info 0x400=verbose 0x800=data 0x1000=debug1 0x2000=debug2 0x4000=debug3 0x8000=debug4 0x10000=func call 0x20000=func args 0x40000=func exit For embedded FM corresponding Chassis Logging must also be enabled and Sm configuration applies to all managers For Host FM, the linux syslog service will need to have an appropriate level of logging enabled.

4.1.5.3 Fe Instance Specific Parameter

The parameter shown in Table 27 can be used only in the `Fm.Fe` section.

Table 27. Fe Instance Specific Parameters

Parameter	Default Value	Description
TcpPort	3245	TCP socket to listen for FV on

4.1.6 Pm Parameters

The following tables describe parameters that can be used in the `Pm` subsection of either the `Common` or `Fm` sections.

Any parameter that can be used in the `Common.Shared` section can also be used in the `Common.Pm` or `Fm.Pm` sections.

4.1.6.1 Pm Controls

The parameters shown in Table 28 set up the when and how the Pm monitors the fabric.

Table 28. Pm Parameters

Parameter	Default Value	Description
ServiceLease	60	ServiceRecord lease with SA in seconds
SweepInterval	10	The PM constantly sweeps and computes fabric statistics If the `SweepInterval` is set to `0`, the PM will not perform sweeps. But instead if queried it will do an immediate PMA operation Tools such as `iba_top` require PM `SweepInterval` be non-zero The default in the sample FM configuration file is ten seconds. However when upgrading from previous FM releases, if a FM configuration file is used without this value specified, a default of `0` is used. This permits upgrades to operate in a mode comparable to the existing configuration and requires specific user action to enable the new 6.0 and above PM features.
MaxClients	3	The maximum number of concurrent PA client applications (For example, `iba_report`, `iba_top`, `iba_rfm`) running against the same PM/PA.
TotalImages FreezeFrameImages	10 5	The PM can retain recent fabric topology and performance data. Each such dataset is referred to as an Image. Images allow for access to recent history and/or Freeze Frame by clients. Each image consumes memory, so care must be taken not to take an excessive amount of memory, especially for larger fabrics `TotalImages` – total images for history and freeze `FreezeFrameImages` – max unique frozen images
FreezeFrameLease	60	IF a PA client application hangs or dies, after this set time, all its frozen images will be released Specified in seconds

4.1.6.2 Threshold Exceeded Message Limit

The parameters shown in Table 29 limit how many ports which exceed their Pm Thresholds will be logged per sweep. These can avoid excessive log messages when extreme fabric problems occur. These parameters can be used in the `ThresholdsExceededMsgLimit` section.

Table 29. Pm ThresholdsExceededMsgLimit Parameters

Parameter	Default Value	Description
Integrity	10	Maximum ports per PM Sweep to log if exceeds configured threshold. A value of `0` will suppress logging of this class of threshold exceeded errors.
Congestion	0	Maximum ports per PM Sweep to log if exceeds configured threshold. A value of `0` will suppress logging of this class of threshold exceeded errors.

Table 29. Pm ThresholdsExceededMsgLimit Parameters (Continued)

Parameter	Default Value	Description
SmaCongestion	0	Maximum ports per PM Sweep to log if exceeds configured threshold. A value of `0` will suppress logging of this class of threshold exceeded errors.
Security	10	Maximum ports per PM Sweep to log if exceeds configured threshold. A value of `0` will suppress logging of this class of threshold exceeded errors.
Routing	10	Maximum ports per PM Sweep to log if exceeds configured threshold. A value of `0` will suppress logging of this class of threshold exceeded errors.

4.1.6.3 Integrity Weights

The parameters shown in Table 30 control the weights for the individual counters which are combined to form the `Integrity` count. These parameters can be used in the `IntegrityWeights` section.

Table 30. Pm IntegrityWeights Parameters

Parameter	Default Value	Description
SymbolErrors	1	Weight for `SymbolError` counter,. `0` causes counter to be ignored
LinkErrorRecovery	30	Weight for `linkErrorRecovery` counter. `0` causes counter to be ignored
LinkDowned	30	Weight for `LinkDowned` counter. `0` causes counter to be ignored
PortRcvErrors	1	Weight for `PortRcvErrors` counter. `0` causes counter to be ignored
LocalLinkIntegrityErrors	30	Weight for `LocalLinkIntegrity` counter. `0` causes counter to be ignored
ExcessiveBufferOverrunErrors	30	Weight for `ExcessiveBufferOverrunErrors` counter. `0` causes counter to be ignored

4.1.6.4 Congestion Weights

The parameters shown in Table 31 control the weight to use for each individual counter when computing congestion, which are combined to form the `Congestion` count. Integrity errors can also cause congestion.

- `XmitDiscard` is available for all devices.
- `XmitCongestionPct10` and `XmitInefficiencyPct10` are available for Intel® QDR switches only. This is based on the deepest Virtual Lanes queue depth.
- `XmitWaitCongestionPct10` and `XmitWaitInefficiencyPct10` available for Intel® QDR and DDR HCAs only. This is based on the time the link was idle because none of the Virtual Lanes that had data to transmit had any credits.

Table 31. Pm CongestionWeights Parameters

Parameter	Default Value	Description
PortXmitDiscard	1000	Weight for `PortXmitDiscard` count of attempted packets which were discarded due to timeout or port down. `0` causes counter to be ignored
PortXmitCongestionPct10	0	Weight for `PortXmitCongestion` on Intel® QDR switches as a percentage of time that the port was congested. The percentage is multiplied by 10 so it has a range from 0 to 1000. `0` causes counter to be ignored
PortXmitInefficiencyPct10	1	Weight for `PortXmitfInefficiency` on Intel® QDR switches as a percentage of time that the port was congestion when it was busy. The percentage is multiplied by 10 so it has a range from 0 to 1000. `0` causes counter to be ignored
PortXmitWaitCongestionPct10	0	Weight for `PortXmitCongestion` on Intel® HCAs as a percentage of time that the port was out of transmit credits. The percentage is multiplied by 10 so it has a range from 0 to 1000. `0` causes counter to be ignored.
PortXmitWaitInefficiencyPct10	1	Weight for `PortXmitInefficiency` on Intel® HCAs as a percentage of time that the port was out of transmit credits when it was busy. The percentage is multiplied by 10 so it has a range from 0 to 1000. `0` causes counter to be ignored.

4.1.6.5 Pm Sweep Operation Control

The parameters shown in Table 32 control the operation of the PM during each sweep.

Table 32. Pm Sweep Parameters

Parameter	Default Value	Description
ErrorClear	2	This controls when the PM clears PMA Error counters. Clearing the counters sooner can result in more accurate running totals and information in iba_report. Clearing the counters later can allow other tools and switch captures to have more information when counters are modest in value. The values are as follows: 0 = clear when non-zero. 1 = clear when 1/4 of individual counters max. 2 = clear when 2/4 of individual counters max. 3 = clear when 3/4 of individual counters max.
EhcaPmaAvoid	1	Provides control over how the PM accesses the PMAs on the IBM eHCA Logical Channel Adapter. The IBM eHCA may be avoided if they are known not to support a PMA. The Intel® Vendor PMA provides increased efficiency in the PM and additional statistics when used with Intel® QDR switches at level 6.0 or later, and/or Intel® HCAs with Intel® OFED+ level 6.0 or later. If 1, avoid PMA on eHCA Logical Channel Adapter.
CaPmaAvoid	0	Provides control over how the PM accesses the PMAs on Channel Adapters. Channel Adapters may be avoided if they are known not to support a PMA. The Intel® Vendor PMA provides increased efficiency in the PM and additional statistics when used with Intel® HCAs with Intel® OFED+ level 6.0 or later. If 1, avoid PMA on all Channel Adapters.
Pma64Enable	1	Enable 64-bit PMA queries to be used for devices which support them. 1 = Enable.
PmaSwVendorEnable	1	Enable use of Intel® vendor specific PMA query for Intel® 12000 series switches. This query allows for additional statistics about congestion and adaptive routing. This query is more efficient and requires fewer PMA packets than IBTA defined queries. 1 = Enable.
PmaSwVendor2Enable	1	Enable use of Intel® Vendor PortCounters2 specific PMA query for Intel® 12000 series switches. This query allows for additional statistics about congestion and adaptive routing. This query is more efficient and requires fewer PMA packets than IBTA defined queries. This flag is ignored if PmaSwVendorEnable is not enabled. 1 = Enable.
PmaCaVendorEnable	0	Enable use of Intel® vendor specific PMA query for Intel® HCAs. This query allows for additional statistics about congestion and adaptive routing. This query is more efficient and requires fewer PMA packets than IBTA defined queries. 1 = Enable.
PmaCaVendor2Enable	0	Enable use of Intel® Vendor PortCounters2 specific PMA query for Intel® HCAs. This query allows for additional statistics about congestion and adaptive routing. This query is more efficient and requires fewer PMA packets than IBTA defined queries. This flag is ignored if PmaCaVendorEnable is not enabled. 1 = Enable.

Table 32. Pm Sweep Parameters (Continued)

Parameter	Default Value	Description
PmaBatchSize	2	Maximum concurrent PMA requests the PM can have in flight while querying the PMAs in the fabric.
MaxParallelNodes	10	Maximum nodes to concurrently issue parallel requests to a given PMA.
MaxAttempts RespTimeout MinRespTimeout	3 250 35	The PM will spend up to `RespTimeout` * `MaxAttempts` per packet. These allow two modes of operation. When `MinRespTimeout` is non-zero, the PM will start with `MinRespTimeout` as the time-out value for requests and use multiples of this value for subsequent attempts if there is a time-out in the previous attempt. PM will keep retrying until the cumulative sum of time-outs for retries is less than `RespTimeout` multiplied by `MaxAttempts`. This approach is recommended and will react quickly to lost packets while still allowing adequate time for slower PMAs to respond. When `MinRespTimeout` is zero, upon a time-out, up to `MaxAttempts` are attempted with each attempt having a time-out of `RespTimeout`. This approach is provided for backward compatibility with previous PM versions.
SweepErrorsLogThreshold	10	Maximum number of PMA node or Port warning messages to output per sweep with regard to nodes which cannot be properly queried.

4.1.6.6 Overrides of the Common.Shared parameters

The Common.Shared parameters can be overridden in the Pm using the parameters described in Table 33

Table 33. Additional Pm Parameters

Parameter	Default Value	Description
LogLevel	2	***Note:*** Overrides the Common.Shared LogLevel Settings Sets log level option for PM: • 0 = disable vast majority of logging output • 1 = fatal, error, warn (syslog CRIT, ERR, WARN) • 2 = +notice, INFIINFO (progress messages) (syslog NOTICE, INFO) • 3 = +INFO (syslog DEBUG) • 4 = +VERBOSE and some packet data (syslog DEBUG) • 5 = +debug trace info (syslog DEBUG) This parameter is ignored for the Embedded FM. Refer to the *Intel® True Scale Fabric Switches 12000 Series User Guide* for information on configuring chassis logging options.
LogFile		***Note:*** Overrides Common.Shared setting. Sets log output location for PM. By default (or if this parameter is empty) log output is accomplished using syslog. However, if a LogFile is specified, logging will be done to the given file. LogMode further controls logging. This parameter is ignored for the Intel® Embedded FM. Refer to the *Intel® True Scale Fabric Switches 12000 Series User Guide* for information on configuring chassis logging options.
SyslogFacility	Local6	***Note:*** Overrides Common.Shared setting. For the Host FM, controls what syslog facility code is used for log messages, Allowed values are: auth, authpriv, cron, daemon, ftp, kern, local0-local7, lpr, mail, news, syslog, user, or uucp. For the Embedded FM, this parameter is ignored
ConfigConsistencyCheckLevel	2	Controls the Configuration Consistency Check for PM. If specified for an individual instance of PM, will override Shared settings. Checking can be completely disabled, or can be set to take action by deactivating Secondary PM if configuration does not pass the consistency check criteria. • 0 = disable Configuration Consistency Checking • 1 = enable Configuration Consistency Checking without taking action (only log a message) • 2= enable Configuration Consistency Checking and take action (log message and shutdown Secondary PM)
ConfigConsistencyCheckMethod	0	Controls the checksum generation method for Configuration Consistency Checking between redundant PMs. Will override Common.Shared settings if specified per instance. Checking MD5 cannot be used when a Host FM and Embedded FM are being used as a redundant pair. In which case the simple additive checksum must be used. The simple additive checksum mechanism may fail to detect some configuration differences. • 0 = use MD5 checksum method • 1 = use simple additive checksum method
Priority	0	0 to 15, higher wins.
ElevatedPriority		0 to 15, higher wins.
DefaultPKey	0xffff	Overrides setting from Common.Shared for PM The PKey for PM. This should be the Default PKey so they can manage all nodes.

4.1.6.7 Additional Parameters for Debug and Development

The Pm supports the parameters in Table 34 to aid diagnosis and debug. Only use these parameters under the direction of your support representative.

Table 34. Pm Debug Parameters

Parameter	Default Value	Description
Debug	0	***Note:*** Overrides Debug setting from Common.Shared Additional parameters for debug/development use - This enables debugging modes for PM.
RmppDebug	0	***Note:*** Overrides RmppDebug setting from Common.Shared If 1, then log additional PM info with regards to RMPP or the Reliable Message Passing Protocol.
CS_LogMask MAI_LogMask CAL_LogMask DVR_LogMask IF3_LogMask SM_LogMask SA_LogMask PM_LogMask PA_LogMask BM_LogMask FE_LogMask APP_LogMask	0x00000000 0x000001ff 0x000001ff 0x000001ff 0x000001ff 0x000001ff 0x000001ff 0x000001ff 0x000001ff 0x000001ff 0x000001ff 0x000001ff	Alternative to use of LogLevel. For advanced users, these parameters can provide more precise control over per subsystem logging. For typical configurations these should be omitted and the `LogLevel` parameter should be used instead. For each subsystem there can be a `LogMask`. The mask selects severities of log messages to enable and is a sum of the following values: 0x1=fatal 0x2=actionable error 0x4=actionable warning 0x8=actionable notice 0x10=actionable info 0x20=error 0x40=warn 0x80=notice 0x100=progress 0x200=info 0x400=verbose 0x800=data 0x1000=debug1 0x2000=debug2 0x4000=debug3 0x8000=debug4 0x10000=func call 0x20000=func args 0x40000=func exit For embedded FM corresponding Chassis Logging must also be enabled and Sm configuration applies to all managers For Host FM, the linux syslog service will need to have an appropriate level of logging enabled.

4.1.7 Bm Parameters

The following tables describe parameters that can be used in the `Pm` subsection of either the `Common` or `Fm` sections.

Any parameter that can be used in the `Common.Shared` section may also be used in the `Common.Bm` or `Fm.Bm` sections.

4.1.7.1 Bm Controls

The parameters shown in Table 35 may be used in the `Bm` subsection of either the `Common` or `Fm` sections.

Table 35. Bm Parameters

Parameter	Default Value	Description
BKey	0	64-bit security Key to assign to BMA agents.
BKeyLease	0	`ServiceRecord` lease with SA in seconds, 0=infinite

4.1.7.2 Overrides of the Common.Shared parameters

The Common.Shared parameters can be overridden in the Bm using the parameters described in Table 36

Table 36. Additional Bm Parameters

Parameter	Default Value	Description
LogLevel	2	***Note:*** Overrides the Common.Shared LogLevel Settings Sets log level option for BM: • 0 = disable vast majority of logging output • 1 = fatal, error, warn (syslog CRIT, ERR, WARN) • 2 = +notice, INFIIINFO (progress messages) (syslog NOTICE, INFO) • 3 = +INFO (syslog DEBUG) • 4 = +VERBOSE and some packet data (syslog DEBUG) • 5 = +debug trace info (syslog DEBUG) This parameter is ignored for the Embedded FM. Refer to the *Intel® True Scale Fabric Switches 12000 Series User Guide* for information on configuring chassis logging options.
LogFile		***Note:*** Overrides Common.Shared setting. Sets log output location for PM. By default (or if this parameter is empty) log output is accomplished using syslog. However, if a LogFile is specified, logging will be done to the given file. LogMode further controls logging. This parameter is ignored for the Intel® Embedded FM. Refer to the *Intel® True Scale Fabric Switches 12000 Series User Guide* for information on configuring chassis logging options.
SyslogFacility	Local6	***Note:*** Overrides Common.Shared setting. For the Host FM, controls what syslog facility code is used for log messages, Allowed values are: auth, authpriv, cron, daemon, ftp, kern, local0-local7, lpr, mail, news, syslog, user, or uucp. For the Embedded FM, this parameter is ignored
ConfigConsistencyCheckLevel	2	Controls the Configuration Consistency Check for BM. If specified for an individual instance of BM, will override Common.Shared settings. Checking can be completely disabled, or can be set to take action by deactivating Secondary BM if configuration does not pass the consistency check criteria. • 0 = disable Configuration Consistency Checking • 1 = enable Configuration Consistency Checking without taking action (only log a message) • 2= enable Configuration Consistency Checking and take action (log message and shutdown Secondary BM)

Table 36. Additional Bm Parameters (Continued)

Parameter	Default Value	Description
ConfigConsistencyCheckMethod	0	Controls the checksum generation method for Configuration Consistency Checking between redundant BMs. Will override Common.Shared settings if specified per instance. Checking MD5 cannot be used when a Host FM and Embedded FM are being used as a redundant pair. In which case the simple additive checksum must be used. The simple additive checksum mechanism may fail to detect some configuration differences. • 0 = use MD5 checksum method • 1 = use simple additive checksum method
Priority	0	0 to 15, higher wins.
ElevatedPriority		0 to 15, higher wins.
DefaultPKey	0xffff	Overrides setting from Common.Shared for BM The PKey for BM. This should be the Default PKey so they can manage all nodes.

4.1.7.3 Additional Parameters for Debug and Development

The Bm supports the parameters in Table 37 to aid diagnosis and debug. Only use these parameters under the direction of your support representative.

Table 37. Bm Debug Parameters

Parameter	Default Value	Description
Debug	0	***Note:*** Overrides Debug setting from Common.Shared Additional parameters for debug/development use - This enables debugging modes for BM.

Table 37. Bm Debug Parameters (Continued)

Parameter	Default Value	Description
RmppDebug	0	***Note:*** Overrides RmppDebug setting from Common.Shared If 1, then log additional BM info with regards to RMPP or the Reliable Message Passing Protocol.
DebugFlag	0	Additional Debug messages unique to BM. Likely engineering specific debug info.
CS_LogMask MAI_LogMask CAL_LogMask DVR_LogMask IF3_LogMask SM_LogMask SA_LogMask PM_LogMask PA_LogMask BM_LogMask FE_LogMask APP_LogMask	0x00000000 0x000001ff 0x000001ff 0x000001ff 0x000001ff 0x000001ff 0x000001ff 0x000001ff 0x000001ff 0x000001ff 0x000001ff 0x000001ff	Alternative to use of LogLevel. For advanced users, these parameters can provide more precise control over per subsystem logging. For typical configurations these should be omitted and the `LogLevel` parameter should be used instead. For each subsystem there can be a `LogMask`. The mask selects severities of log messages to enable and is a sum of the following values: 0x1=fatal 0x2=actionable error 0x4=actionable warning 0x8=actionable notice 0x10=actionable info 0x20=error 0x40=warn 0x80=notice 0x100=progress 0x200=info 0x400=verbose 0x800=data 0x1000=debug1 0x2000=debug2 0x4000=debug3 0x8000=debug4 0x10000=func call 0x20000=func args 0x40000=func exit For embedded FM corresponding Chassis Logging must also be enabled and Sm configuration applies to all managers For Host FM, the linux syslog service will need to have an appropriate level of logging enabled.

4.1.8 Fm Instance Shared Parameters

Any parameter that can be used in the `Common.Shared` section may also be used in the `Fm.Shared` section.

Table 38 describes parameters that are used in the `Shared` subsection of `Fm` section.

Table 38. Fm Instance Shared Parameters

Parameter	Default Value	Description
Name	fm0	Name for the given FM instance. Also used to mark log messages with `_sm`, `_fe`, `_pm`, `_bm` appended This parameters is ignored for the Embedded FM
Hca Port PortGUID	1 1 0	Each FM runs on a single local HCA port, The port may be specified by `Hca` and `Port` or by `PortGuid` `Hca` = local HCA to use for Fm Instance, 1=1st HCA `Port` = local HCA port to use for Fm instance, 1=1st Port `PortGUID` = local HCA port to use for Fm instance If `PortGUID` is `0`, it is ignored and `Hca/Port` is used. When `PortGUID` is non-zero, it will be used and `Hca/Port` will be ignored. These parameters are ignored for the Embedded FM.
SubnetPrefix	0xfe8000000000000	Unique subnet prefix to assign to this Fabric. The same subnet prefix must be used by all redundant SMs on a given fabric.

§ §

5.0 Virtual Fabrics

5.1 Overview

Virtual Fabrics (vFabrics) bring to the True Scale Fabric many of the capabilities of Ethernet VLANs and Fibre Channel Zoning.

Using vFabrics, the administrator may slice up the physical fabric into many overlapping virtual fabrics. The administrator's selections determine how the InfiniBand* Technology-specific configuration of the fabric is performed.

The goal of vFabrics is to permit multiple applications to be run on the same fabric at the same time with limited interference. The administrator can control the degree of isolation.

As in the Ethernet VLANs, a given node may be in one or more vFabrics. vFabrics may have overlapping or completely independent membership. When IPoIB is in a fabric, typically each vFabric represents a unique IP subnet, in the same way a unique subnet is assigned to different virtual LANs (VLANs).

Each vFabric can be assigned Quality of Service (QoS) policies to control how common resources in the fabric are shared among vFabrics.

Additionally, as in the Fibre Channel Zoning, vFabrics can be used in an True Scale fabric with shared I/O and InfiniBand* Technology-compliant storage. vFabrics help control and reduce the number of storage targets visible to each host, making Plug and Play host software and storage-management tools easier to use. Some typical usage models for vFabrics include the following:

- Separating a cluster into multiple vFabrics so that independent applications can run with minimal or no effect on each other
- Separating classes of traffic. For example, putting a storage controller in one vFabric and a network controller in another, using one wire for all networking becomes secure and predictable.

A vFabric consists of a group of applications that run on a group of devices. For each vFabric the operational parameters of the vFabric can be selected.

5.1.1 Quality of Service

Quality of Service (QoS) allows the priority of each vFabric to be configured. QoS policies allow multiple applications to run on the same fabric without interference, through controlling the use of network resources. These policies allow the following:

- Setting minimum bandwidth percentages for high-volume traffic.
- Addressing needs of high-priority, low-volume traffic.

QoS policies allow the user to assign a bandwidth minimum to classes of traffic to define how network resources are allocated. For instance, networking traffic can be assigned to one vFabric, storage to a second vFabric, and compute to a third. These classes of traffic can then be assigned a bandwidth percentage to which they will be limited when a link is saturated. Low-volume traffic requires low latency, like administrative or control traffic, the user can specify this as high priority.

5.1.1.1 QoS Operation

Virtual Lanes (VLs) permit multiple logical flows over a single physical link. Each physical link has a number of VLs. Each class of service has a Service Level (SL). The local route header of the packets carries the SL, which identifies the different flows

within the subnets. The amount of resources allocated to the VL is based on the bandwidth configured for the vFabric containing the SL. The Subnet Manager programs the SL-to-VL mappings and VL Arbitration tables to achieve the QoS policies.

Applications can query for path records by Service ID to obtain path records associated with the given application. These path records contain the SL associated with the traffic class/QoS vFabric. In turn, this SL is used for packets within this class of traffic.

5.2 Configuration

The configuration of vFabrics consists of the following sections:

- Applications — describes applications that can run on one or more end nodes
- DeviceGroups — describes a set of end nodes in the fabric
- VirtualFabrics — defines a vFabric consisting of a group of applications, a set of devices, and the operating parameters for the vFabric

Each Application, DeviceGroup, and VirtualFabric must be given a unique name that is used to reference it. Applications, DeviceGroups, and VirtualFabrics may be defined in the `Common` or `Fm` sections. Those defined in the `Common` section apply to all `Fm` Instances. Those defined in an `Fm` section describe additional Applications, DeviceGroups, or VirtualFabrics that are specific to the given `Fm` instance.

5.2.1 Application Parameters

Applications are defined within the `Applications` section. This section contains zero or more `Application` sections.

The layout is as follows:

```
<Applications>

    <Application>

      <!-- application parameters -->

    </Application>

</Applications>
```

Each `Application` section has zero or more `ServiceIDs` subsections and/or `MGIDs` subsections. These are matched against PathRecord and Multicast SA queries so that the returned SLID/DLID, PKey, SL, and so on are appropriate for the vFabric that contains the application(s).

`ServiceIDs` subsections are 64-bit values that identify applications within a PathRecord query. In many ways ServiceIDs are the InfiniBand* Architecture's equivalent of TCP socket ports. ServiceIDs are typically used within the InfiniBand* Technology-compliant Communication Manager protocol to identify the application making a connection request. When an application issues a PathRecord query to the SA, the ServiceID in the query is compared against the `ServiceID` subsections in the various `VirtualFabrics` sections.

ServiceIDs are assigned by application writers and standards bodies such as IBTA and IEEE. Consult with the application supplier to determine the Service IDs used by the application.

Multicast GIDs (MGIDs) are 128-bit values that identify multicast groups for Unreliable Datagram applications such as IPoIB. MGIDs are represented as two 64-bit values separated by a colon (:). For example: `0xabc:0x123567`

This way of representing 128-bit values is the same as used in other FastFabric Toolset commands such as `iba_saquery`, `iba_showmc`, and so on. Applications not used in any vFabric have no effect.

Note: The `ifs_fm.xml-sample` configuration file contains many standard preconfigured applications that are referenced in VirtualFabrics, which can be created by the administrator.

Note: InfiniBand* Architecture Standard mechanisms establish connections for the association of applications to VirtualFabrics, by using the ServiceID in PathRecord SA queries (for unicast applications) and the use of MGIDs in McMemberRecord SA queries (for multicast applications).

Note: Some unicast applications, notably `openmpi`, `mvapich` and `mvapich2`, use nonstandard mechanisms to establish connections. Therefore, the unicast applications will need to have their PKey and SL manually configured consistent with the vFabric used.

Table 39. Application Parameters

Parameter	Description
Name	Name for Application. Every Application must have a unique name. The name must be unique among all Application names within an FM instance. When defined at Common level, the name must be unique within all instances. The name is limited to 64 characters and is case sensitive.
ServiceID	A single 64-bit service ID to match against.
ServiceIDRange	A range of service IDs to match against. Any service ID within the range (inclusive) is considered a match. The range is two 64-bit values separated by a dash such as: 0-0xffffffffffffffff
ServiceIDMasked	A masked compare of service ID to match against. Matches service IDs that when ANDed with second value (the mask) match the first value. The mask is two 64-bit values separated by a * such as: 0x120003567*0xff000ffff
MGID	A single 128-bit MGID to match against.
MGIDRange	A range of MGIDs to match against. Any MGID within the range (inclusive) is considered a match. The range is two 128-bit values separated by a dash such as: 0:0-0xffffffffffffffff:0xffffffffffffffff

Table 39. Application Parameters (Continued)

Parameter	Description
MGIDMasked	A masked compare of MGID to match against. Matches MGIDs that when ANDed with second value (the mask) match the first value. The mask is two 128-bit values separated by a * such as: 0xabc:0x120003567*0xfff:0xff000ffff
Select	Special selection cases. The following selection cases can be used as an easy catch-all: • `UnmatchedServiceID` – matches all applications' service IDs that match none of the vFabrics after filtering by `src/dest/requestor`. • `UnmatchedMGID` – matches all applications' MGIDs that match none of the vFabrics after filtering by `src/dest/requestor`. • `SA` – allows an application to be specified that includes SA queries. This allows SA query operations to be assigned to an appropriate vFabric. Per IBTA, SA queries must use the default Partition Key (0x7fff or 0xffff). However, other aspects of SA access can be controlled (SL, and so on). The listed specifiers are case insensitive By having multiple select parameters, multiple special cases can be combined as needed in the same application section.
IncludeApplication	This includes all of the service IDs, MGIDs, and special selections in the given application. Loops (including the parent application) are not allowed. There is a nesting limit of 32.

The following is the `Applications` section from the `ifs_fm.xml-sample` file (see the `ifs_fm.xml-sample` file for a complete example of many standard applications):

```
<Applications>

  <!--  Each Application can have one or more ServiceIDs and/or MGIDs. -->

  <!--  These will be matched against PathRecord and Multicast SA queries -->

  <!--  so that the returned SLID/DLID, PKey, SL, etc are appropriate for -->

  <!--  the Virtual Fabric(VF) which contains the application(s). -->

  <!--  Every Application must have a unique <Name> -->

  <!--  The name must be unique among all Application names within an -->

  <!--  FM instance. -->

  <!--  When defined at Common level must be unique within all instances. -->

  <!--  The name is limited to 64 characters and is case sensitive. -->

  <!--  ServiceIDs are 64 bit values which identify applications within the -->

  <!--  PathRecord query and are typically used within the InfiniBand -->

  <!--  Communication Manager (CM) protocol to identify the application for -->

  <!--  a connection request. -->

  <!--  In many ways ServiceIDs are IBs equivalent of TCP "socket ports". -->

  <!--  ServiceIDs can be specified in the following ways in the -->

  <!--  Application section: -->
```

```
  <!--  a single ServiceID: <ServiceID>0x1234567812345678</ServiceID> -->

  <!--  a range of ServiceIDs:         <ServiceIDRange>0-0xffffffffffffffff</
ServiceIDRange> -->

  <!--  a masked compare of ServiceID:
        <ServiceIDMasked>0x120003567*0xff000ffff</ServiceIDMasked> -->

  <!--         Matches ServiceIDs which when ANDed with 2nd value (the -->

  <!--         mask) match the 1st value -->

  <!--  all others not in any VF: <Select>UnmatchedServiceID</Select> -->

  <!--         UnmatchedServiceID matches all applications' ServiceIDs -->

  <!--         which match none of the Virtual Fabrics after filtering -->

  <!--         by src/dest/requestor. -->

  <!--         Can be used as an easy catch-all. -->

  <!--  MGIDs (IB Multicast GIDs) are 128 bit values which identify -->

  <!--  multicast groups for Unreliable Datagram applications such as IPoIB -->

  <!--  MGIDs are represented as two 64 bit values separated by a colon (:) -->

  <!--  This way of representing 128 bit values is the same as used in -->

  <!--  other FastFabric Tools such as iba_saquery, iba_showmc, etc. -->

  <!--  MGIDs can be specified in the following ways in the Application -->

  <!--  section: -->

  <!--  a single MGID: <MGID>0xabc:0x123567</MGID> -->

  <!--  a range of MGIDs:         <MGIDRange>0:0-
0xffffffffffffffff:0xffffffffffffffff</MGIDRange> -->

  <!--  a masked compare of MGID:
        <MGIDMasked>0xabc:0x120003567*0xfff:0xff000ffff</MGIDMasked> -->

  <!--         Matches MGIDs which when ANDed with 2nd value (the -->

  <!--         mask) match the 1st value. -->

  <!--  all others not in any VF: <Select>UnmatchedMGID</Select> -->

  <!--         UnmatchedMGID matches all applications' MGIDs which match -->

  <!--         none of the Virtual Fabrics after filtering by -->

  <!--         src/dest/requestor. -->

  <!--         Can be used as an easy catch-all -->

  <!--  A special case is SA queries.  This special "application" allows -->

  <!--  SA query operations to be assigned to an appropriate VF.  Per IBTA -->

  <!--  SA queries must use the default Partition Key (0x7fff or 0xffff). -->

  <!--  However other aspects of SA access can be controlled (SL, etc). -->

  <!--  This syntax allows an Application to be specified which includes -->
```

```
  <!--  SA queries: <Select>SA</Select> -->
  <!--  In addition, an Application can include another Application via: -->
  <!--  <IncludeApplication>myapplication</IncludeApplication> -->
  <!--  This will include all the ServiceIDs and MGIDs in the given -->
  <!--  Application. -->
  <!--  Loops (including the parent Application) are not allowed. -->
  <!--  There is a nesting limit of 32. -->
  <!--  Some predefined Applications follow.  These are also a good set -->
  <!--  of examples to cut/paste to make other site specific applications: -->
  <!--  All Applications, can be used when per Application VFs not needed -->
  <Application>
    <Name>All</Name>
    <!--  Selects all service ID's -->
    <ServiceIDRange>0-0xffffffffffffffff</ServiceIDRange>
    <!--  selects all MGID's -->
    <MGIDRange>0:0-0xffffffffffffffff:0xffffffffffffffff</MGIDRange>
    <!--  these are redundant in this case, but don't hurt -->
    <Select>UnmatchedServiceID</Select>
    <Select>UnmatchedMGID</Select>
  </Application>
  <!--  An application just consisting of SA queries -->
  <Application>
    <Name>SA</Name>
    <Select>SA</Select>
  </Application>
  <!--  IPv4 over IB (IETF UD mode) for all partitions -->
  <Application>
    <Name>IPv4</Name>
    <!--  MGID = 0xffFS401bPPPP0000:00000000GGGGGGGG -->
    <!--  where F=flags, S=scope, P=PKey and G=IP Multicast Group -->
    <MGIDMasked>0xff00401b00000000:0*0xff00ffff0000ffff:0xffffffff00000000</
MGIDMa    sked>
  </Application>
  <!--  IPv4 over IB (IETF UD mode) for PKey 0x9001/0x1001 -->
```

```
    <Application>
      <Name>IPv4_1001</Name>
      <!--  MGID = 0xffFS401bPPPP0000:00000000GGGGGGGG -->
      <!--  where F=flags, S=scope, P=PKey and G=IP Multicast Group -->
      <MGIDMasked>0xff00401b90010000:0*0xff00ffffffffffff:0xffffffff00000000</
MGIDMa    sked>
    </Application>
    <!--  IPv6 over IB (IETF UD mode) for all partitions -->
    <Application>
      <Name>IPv6</Name>
      <!--  MGID = 0xffFS601bPPPPGGGG:GGGGGGGGGGGGGGGG -->
      <!--  where F=flags, S=scope, P=PKey and G=IP Multicast Group -->
      <MGIDMasked>0xff00601b00000000:0*0xff00ffff00000000:0</MGIDMasked>
    </Application>
    <!--  IPv6 over IB (IETF UD mode) for PKey 0x9001/0x1001 -->
    <Application>
      <Name>IPv6_1001</Name>
      <!--  MGID = 0xffFS601bPPPPGGGG:GGGGGGGGGGGGGGGG -->
      <!--  where F=flags, S=scope, P=PKey and G=IP Multicast Group -->
      <MGIDMasked>0xff00601b90010000:0*0xff00ffffffff0000:0</MGIDMasked>
    </Application>
    <!--  IPoIB Connected Mode (IPv4 and IPv6) -->
    <Application>
      <Name>IPoIBCM</Name>
      <!--  Unlike UD mode the ServiceID does not include PKey -->
      <!--  ServiceID = 0x1000000000XXXXXX where XXXXXX is QPN -->
      <ServiceIDMasked>0x1000000000000000*0xfffffffffff000000</ServiceIDMasked>
    </Application>
    <!--  Sockets Direct Protocol (SDP) -->
    <Application>
      <Name>SDP</Name>
      <!--  ServiceID = 0x000000000001XXXX where XXXX is socket "port number" -->
      <ServiceIDRange>0x0000000000010000-0x000000000001ffff</ServiceIDRange>
    </Application>
```

```
<!--  Reliable Datagram Service (RDS) (for Oracle) -->
<Application>
  <Name>RDS</Name>
  <!--  ServiceID = 0x0000000000106XXXX where XXXX is socket "port number" -->
  <ServiceIDRange>0x0000000001060000-0x000000000106ffff</ServiceIDRange>
</Application>
<!--  Unfortunately, none of the OFED Verbs MPIs use the SA nor CM. -->
<!--  Hence to put an MPI job in its own Virtual Fabric, the PKey/SL of -->
<!--  the desired Virtual fabric must be manually provided to the MPI run -->
<Application>
  <Name>VerbsMPI</Name>
  <!--  <ServiceID>TBD</ServiceID> -->
</Application>
<!--  HPC Libraries (MPI, SHMEM, etc) which use the Intel True Scale HCAs -->
<!--  and PSM technology can query the SA and hence can use service IDs -->
<!--  to select the desired Virtual Fabric. -->
<!--  when separating PSM Control from Data Traffic: -->
<!--  1st Service ID in range is used for Control traffic. -->
<!--  This will be the service ID used on the command line. -->
<Application>
  <Name>PSM_Control</Name>
  <ServiceIDRange>0x1000117500000000-0x1000117500000007</ServiceIDRange>
  <!--  for simplicity, when serviceId is needed on mpirun command line -->
  <!--  we also allow serviceId 0x1 to 0x7. -->
  <ServiceIDRange>0x1-0x7</ServiceIDRange>
</Application>
<!--  when separating PSM Control from Data Traffic: -->
<!--  Control Service ID +8 is used for Eager/Bulk traffic. -->
<!--  Do NOT use this service ID on the command line. -->
<Application>
  <Name>PSM_Data</Name>
  <ServiceIDRange>0x1000117500000008-0x100011750000000f</ServiceIDRange>
  <!--  for simplicity, when serviceId is needed on mpirun command line -->
  <!--  we also allow serviceId 0x9 to 0xf. -->
```

```
    <ServiceIDRange>0x9-0xf</ServiceIDRange>
  </Application>
  <!--  When not separating PSM Control from Data Traffic, map all -->
  <!--  PSM Service IDs as one application -->
  <Application>
    <Name>PSM</Name>
    <ServiceIDRange>0x1000117500000000-0x100011750000000f</ServiceIDRange>
    <!--  for simplicity, when serviceId is needed on mpirun command line -->
    <!--  we also allow serviceId 0x1 to 0xf -->
    <ServiceIDRange>0x1-0xf</ServiceIDRange>
  </Application>
  <!--  all the common networking protocols/applications -->
  <Application>
    <Name>Networking</Name>
    <IncludeApplication>IPv4</IncludeApplication>
    <IncludeApplication>IPv6</IncludeApplication>
    <IncludeApplication>IPoIBCM</IncludeApplication>
    <IncludeApplication>SDP</IncludeApplication>
    <IncludeApplication>RDS</IncludeApplication>
</Application>
  <!--  all the common storage protocols/applications -->
  <Application>
    <Name>Storage</Name>
</Application>
  <!--  all the Intel Virtual IO Controller protocols/applications -->
  <Application>
  <  Name>VirtualIO</Name>
</Application>
  <!--  all the common compute protocols/applications -->
  <Application>
    <Name>Compute</Name>
    <IncludeApplication>VerbsMPI</IncludeApplication>
    <IncludeApplication>PSM</IncludeApplication>
  </Application>
```

```
  <!--  This is a catchall which can be used to identify all applications -->
  <!--  which are not part of some other vFabric. -->
  <Application>
    <Name>AllOthers</Name>
    <Select>UnmatchedServiceID</Select>
    <Select>UnmatchedMGID</Select>
  </Application>
</Applications>
```

5.2.2 DeviceGroup Parameters

Device groups are defined within the `DeviceGroups` section. This section contains zero or more `DeviceGroup` sections.

The layout is as follows:

```
<DeviceGroups>
    <DeviceGroup>
      <!-- device group parameters -->
    </DeviceGroup>
</DeviceGroups>
```

Each `DeviceGroup` section can have one or more devices (nodes and ports). Devices are matched against PathRecord and Multicast SA queries so that the returned SLID/DLID, PKey, SL, and so on are appropriate for the vFabric that contains the device(s).

Devices in a `DeviceGroup` but not found in the fabric are ignored.

When security is enabled for a vFabric, further measures will be taken, using PKeys, to secure the vFabric and ensure devices in other vFabrics cannot talk to devices in the given vFabric. Such security includes hardware enforced per-packet PKey checking and enforcement.

DeviceGroups not used in any vFabric will have no effect.

Note: To effectively use the `DeviceGroups` section, the administrator must carefully list the required devices in each `DeviceGroup` section. FastFabric Toolset commands, such as `iba_saquery` and `iba_report`, can help generate the lists.

Note: Portions of `iba_report` XML output, such as `iba_report -o brnodes -x`, can be cut and pasted into the appropriate `DeviceGroup` sections.

When assigning devices to a vFabric, the capabilities of the device must be considered. There is a limit of the number of Secure VirtualFabrics a given device's port can be in. That limit is dependent on the PKey capabilities of the hardware.

The PKey capabilities of the devices shipped by Intel® are as follows:

- Intel® SDR HCA: 4

- Intel® DDR HCA (True Scale): 4
- SDR PCI-X (Tavor) HCA: 64
- DDR PCIe dual port (Arbel) HCA: 64
- DDR PCIe single port (Sinai) HCA: 64
- DDR Connect-X HCA: 128
- I9K SWE0:8, Port 0: 8, Port 1-24: 32

Given the listed capabilities, users must be especially careful about how a Switch Port 0 is included in many vFabrics. To avoid specifying Switch Port 0 in too many vFabrics, check `<DeviceGroups>` which include the following:

- `<Select>SWE0</Select>`
- `<NodeType>SW</NodeType>`
- `<Select>All</Select>`
- Explicit specification of a switch by `PortGuid`, `NodeGuid` or `NodeDesc`.

In a typical customer configuration, Switch Port 0 only needs to be part of the admin or default vFabric, for example, PKey 0xffff.

When using Intel® HCAs, any given Channel Adapter port cannot be in more than four secure VirtualFabrics.

When using non-secure VirtualFabrics, the FM may consolidate PKeys so that a configuration is valid that contains more vFabrics than the PKey capability.

For secure VirtualFabrics, the PKey is programmed in both the HCA and its neighboring switch port, in this case the number of PKeys the HCA can be in is the lesser of the PKey capabilities of the HCA and the neighboring switch port.

Table 40 lists the parameters and their descriptions for the `DeviceGroup` subsection under the `DeviceGroups` section.

Table 40. DeviceGroup Parameters

Parameter	Description
Name	Name for DeviceGroup. Every `DeviceGroup` must have a unique Name. The name must be unique among all `DeviceGroup` names within an `FM` instance. When defined at `Common` level must be unique within all instances The name is limited to 64 characters and is case sensitive.
SystemImageGUID	Include all of the ports and nodes within the given system as identified by its 64-bit System Image GUID.
NodeGUID	Selects all of the ports in the node (a Channel Adapter or switch is a single node) as identified by its 64-bit node GUID.
PortGUID	Selects the given port as identified by its 64-bit port GUID.
NodeDesc	Selects all nodes exactly matching the given name. Typically, will match one node, but multiple nodes with the same NodeDesc are possible. While easier to configure, use of this mechanism is less secure than specification using GUIDs. It's very easy for a systems node description to be changed.

Table 40. DeviceGroup Parameters (Continued)

Parameter	Description
NodeType	Selects all ports on all nodes of the specified node type. The following types may be specified: • CA – All Channel Adapters • SW – All Switches • RT – All Routers The listed types are case insensitive By having multiple Select parameters, multiple node types can be combined as required in the same DeviceGroup section.
Select	Special selection cases. The following selection cases can be used as an easy catch-all: • All – All Devices • Self – This FM instance's port • AllSMs – All ports that report IsSM Capability Mask, including self • SWE0 – Every Switch Port 0 with Enhanced Port 0 in Capability Mask. By definition also includes all Embedded SMs and managed spines of Intel® Internally Managed Switches. The listed selection cases are case insensitive By having multiple Select parameters, multiple selections can be combined as required in the same DeviceGroup section.
IncludeGroup	This will include all the devices in the given DeviceGroup. Loops (including the parent DeviceGroup) are not allowed. It is valid to have more than one specification match the same device; in which case, the device is only included in the DeviceGroup once. There is a nesting limit of 32.

Note: The SystemImageGUID, NodeGUID, PortGUID, and NodeDesc for devices presently in the fabric can be identified by using iba_report or iba_report -o comps. If required iba_report -x or iba_report -o comps -x provides an XML output from which individual NodeGUID, PortGUID, SystemImageGUID, or NodeDesc lines can be cut/pasted into the required DeviceGroup sections.

The following is the DeviceGroups section from the ifs_fm.xml-sample file:

```
<DeviceGroups>

  <!--  Each DeviceGroup can have one or more Nodes/Ports (eg. Devices). -->

  <!--  These will be matched against PathRecord and Multicast SA queries -->

  <!--  so that the returned SLID/DLID, PKey, SL, etc are appropriate for -->

  <!--  the Virtual Fabric(VF) which contains the Device(s). -->

  <!--  Devices in a DeviceGroup but not found in the fabric are ignored. -->

  <!--  Every DeviceGroup must have a unique <Name> -->

  <!--  The name must be unique among all DeviceGroup names within an -->

  <!--  FM instance. -->

  <!--  When defined at Common level must be unique within all instances. -->

  <!--  The name is limited to 64 characters and is case sensitive. -->

  <!--  Devices can be explicitly specified in the following ways in the -->

  <!--  DeviceGroup section: -->

  <!--  A SystemImageGUID: <SystemImageGUID>0x123567</SystemImageGUID> -->
```

```
<!--         selects all nodes and ports in the system -->
<!--  A NodeGUID: <NodeGUID>0x123567</NodeGUID> -->
<!--         selects all ports in the node (a CA or SW is a single node) -->
<!--  A PortGUID: <PortGUID>0x123567</PortGUID> -->
<!--         selects a specific port -->
<!--  A NodeDesc: <NodeDesc>Some Name</NodeDesc> -->
<!--         selects all nodes exactly matching given name -->
<!--         typically will match exactly one, but multiple nodes with -->
<!--         same NodeDesc is possible. -->
<!--  Devices can be generically specified in the following ways in the -->
<!--  DeviceGroup section: -->
<!--  All CAs: <NodeType>CA</NodeType> -->
<!--  All Switches: <NodeType>SW</NodeType> -->
<!--  All Routers: <NodeType>RT</NodeType> -->
<!--  The values above are case insensitive: CA, SW, RT -->
<!--  Each of the above selects all ports on the selected nodes. -->
<!--  The above can be combined as desired in the same DeviceGroup. -->
<!--  Some special types of Devices can be generically specified in -->
<!--  the following ways in the DeviceGroup section: -->
<!--  All Devices: <Select>All</Select> -->
<!--  This FM Instance's Port: <Select>Self</Select> -->
<!--  All SMs: <Select>AllSMs</Select> -->
<!--         All ports which report IsSM Capability mask, including Self -->
<!--  All Switch Enhanced Port 0s: <Select>SWE0</Select> -->
<!--         All Switch Port 0 with Enhanced Port 0 in Capability Mask. -->
<!--         By definition will also include all Embedded SMs. -->
<!--  The values above are case insensitive: All, Self, AllSMs, SWE0 -->
<!--  The above can be combined as desired in the same DeviceGroup. -->
<!--  In addition, a DeviceGroup can include another DeviceGroup via: -->
<!--  <IncludeGroup>mygroup</IncludeGroup> -->
<!--  This will include all the Devices in the given DeviceGroup. -->
<!--  Loops (including the parent DeviceGroup) are not allowed. -->
<!--  It is valid to have more than 1 specification match the same device -->
<!--  in which case the device is only included in the DeviceGroup once. -->
```

```
<!--  There is a nesting limit of 32. -->
<!--  Some predefined DeviceGroups follow.  These are also a good set -->
<!--  of examples to cut/paste to make other site specific DeviceGroups: -->
<!--  All Nodes/Ports, can be used when Device specific VFs not needed -->
<DeviceGroup>
  <Name>All</Name>
  <Select>All</Select>
</DeviceGroup>
<!--  All Channel Adapters, includes HCAs and TCAs -->
<DeviceGroup>
  <Name>AllCAs</Name>
  <NodeType>CA</NodeType>
</DeviceGroup>
<!--  All ports on all Switches -->
<DeviceGroup>
  <Name>AllSWs</Name>
  <NodeType>SW</NodeType>
</DeviceGroup>
<!--  All Enhanced Port 0 ports on all Switches -->
<DeviceGroup>
  <Name>AllSWE0s</Name>
  <Select>SWE0</Select>
</DeviceGroup>
<!--  All End-Nodes including Enhanced Port 0 ports on all Switches -->
<DeviceGroup>
  <Name>AllEndNodes</Name>
  <IncludeGroup>AllCAs</IncludeGroup>
  <IncludeGroup>AllSWE0s</IncludeGroup>
</DeviceGroup>
<!--  All ports with IsSM Capability (eg. running an SM) -->
<DeviceGroup>
  <Name>AllSMs</Name>
  <Select>AllSMs</Select>
</DeviceGroup>
```

```
  <!--  Just the port running this FM Instance -->
  <DeviceGroup>
    <Name>Self</Name>
    <Select>Self</Select>
  </DeviceGroup>
  <!--  Example using specific device selection -->
  <DeviceGroup>
    <Name>Example</Name>
    <SystemImageGUID>0x123567</SystemImageGUID>
    <NodeGUID>0x123567</NodeGUID>
    <PortGUID>0x123567</PortGUID>
    <NodeDesc>Some Name</NodeDesc>
    <IncludeGroup>AllSWE0s</IncludeGroup>
  </DeviceGroup>
</DeviceGroups>
```

5.2.3 VirtualFabric Parameters

vFabrics are defined within the `VirtualFabrics` section. This section contains zero or more `VirtualFabric` sections.

The layout is as follows:

```
<VirtualFabrics>
    <VirtualFabric>
      <!-- virtual fabric parameters -->
    </VirtualFabric>
</VirtualFabrics>
```

Each `VirtualFabric` contain the following:

- One or more groups of devices
- One or more sets of Applications
- Administrator policies

The vFabrics control the security configuration for the given set of devices and applications in the overall fabric.

Each `VirtualFabric` section can have one or more `Applications` and `DeviceGroups` sections. Both the devices and the applications are matched against PathRecord and Multicast SA queries. The returned SLID/DLID, PKey, SL, and so on, are appropriate for the vFabric that contain the involved devices and applications.

5.2.3.1 Device Membership and Security

Devices are specified by the devicegroup name in the following ways in the `VirtualFabric` section:

- As Full Members:

```
<Member>group_name</Member>
```

  Such devices may talk to any other `Member` or `LimitedMember`.

- As Limited Members:

```
<LimitedMember>group_name</LimitedMember>
```

When `Security` is `1` (On), `LimitedMembers` are not permitted to talk to other `LimitedMembers`. However, `LimitedMembers` can always talk to `Members`. `LimitedMembers` cannot join multicast groups in the vFabric.

When security is on for a vFabric, PKeys and switch hardware enforcement is used to secure the vFabric and enforce the `Members` and `LimitedMembers` rule. Security also ensures that devices in other vFabrics cannot talk to devices in the given vFabric. This security includes, hardware enforced per-packet PKey checking, and enforcement by switches and end nodes.

If `Security` is `0` (Off), `LimitedMembers` are treated the same as `Members`. This allows the user to easily turn off Security for a vFabric without changing the rest of the definition.

`Member` and `LimitedMember` can each be specified more than once per VirtualFabric if required. If a device is in both the `Members` and `LimitedMembers DeviceGroups` subsection, it is treated as a `Member`. This allows `All` to be specified as a `LimitedMember`; then selected `Members` can be specified, ensuring the VirtualFabric includes all devices while allowing a limited set of `Members`.

Devices in a `DeviceGroup` but not found in the fabric are ignored.

By default the FM picks an available PKey for the vFabric. When Security is off, the SM may share the same PKey among multiple vFabrics.

If required a user-selected PKey can be specified.

PKey must be specified for applications that do not use SA PathRecord queries, including MPIs that use non-IBTA compliant mechanisms for job startup.

Note: When secure vFabrics are used, every host port must be a member of at least one vFabric for proper operation of host tools such as `p1stats` and `iba_mon` it. If a host port is not a member of any vFabric these tools will be unable to access the local port.

5.2.3.2 Application Membership

Applications are specified by the application name in the `VirtualFabric` section.

5.2.3.3 Policies

The QoS Policy, MaxMTU, and MaxRate can be specified for a vFabric. This is one way to restrict the capabilities of the vFabric and influence the performance available to applications and devices within the vFabric.

5.2.3.4 Quality of Service (QoS)

When setting up QoS within vFabrics, the user should identify the maximum bandwidth for each vFabric when the link is saturated. High-priority, low-volume groups can be configured with the `HighPriority` setting. The user can also configure the SL used to enforce this bandwidth. If not configured, the SL is assigned. The amount of configured bandwidth cannot exceed 100 percent. If the configured bandwidth does exceed 100 percent, a parser error is given. If there is a mixture of QoS and non-QoS vFabrics configured, all non-QoS vFabrics are assigned the same SL. Any unconfigured bandwidth is assigned to that SL.

5.2.3.5 The Default Partition

The InfiniBand* Architecture Standard requires every fabric to have a default partition. At a minimum this vFabric is used by all end nodes to interact with the SM/SA.

To meet this requirement there must be an enabled vFabric with the following items:

- A PKey of 0x7fff (or 0xffff)
- The SA application (for example, an Application with `<Select>SA</Select>`)
- The only enabled vFabric that includes the SA application
- Have `AllSMs` (for example, a DeviceGroup with `<Select>AllSMs</Select>`) as a `Member`
- Have `All` (for example, a DeviceGroup with `<Select>All</Select>`) as a `Member` or `LimitedMember`
- Additional Application as needed
- Additional DeviceGroups as `Members` or `LimitedMembers` as needed
- All other vFabric policies (QoS, Security, MTU...) may be set as needed

Note: It is a requirement that all chassis-managed spines (for example, SWE0) be a member of the default partition so that the chassis can access and manage its own leaf switch chips.

5.2.3.6 IPoIB and vFabrics

vFabrics are configured within the hardware in the order in which they appear in the configuration file. When IPoIB runs, it uses the first PKey on the given port for the default (ib0...) network device. Therefore, it is best to place the Networking/IPoIB vFabric first.

IPoIB starts with the PKey for the IPoIB interface and uses that to define the MGID of the VLAN's broadcast multicast group. Many aspects of the IPoIB VLAN are defined by the multicast group itself. Among them are the MTU for the VLAN.

A given port or node can participate in more than one IPoIB subnet. Each such subnet must have its own unique PKey. For vFabrics other than the first, the PKey should be manually specified in the `VirtualFabric` section and the PKey must be supplied to IPoIB. On some Linux systems with the Intel® True Scale Fabric or OFED stack, additional IPoIB virtual interfaces can be created by a command such as:

```
echo 0x1234 > /sys/class/net/ib0/create_child
```

The PKey given is ORed with 0x8000 to define the PKey for the multicast group. This creates an ib0.9234 interface that can be assigned the appropriate IP address and IP parameters.

The operation of Linux with multiple IPoIB subnets is very similar to the use of IP over Ethernet when VLANs are being used. It is up to the adminstrator which network interfaces are actually used and assigned IP addresses. This is done using the standard `ifcfg` files.

5.2.3.7 MPI and vFabrics

Many MPI implementations do not follow the InfiniBand* Architecture Standard. MPIs such as openmpi, mvapich, and mvapich2 when ran using InfiniBand* Technology verbs, use nonstandard mechanisms to establish connections, do not make PathRecord requests, and do not use ServiceIDs to determine connection parameters.

To use vFabrics in conjunction with those MPIs, the PKey and BaseSL must be manually specified in the `VirtualFabric` section. The selected PKey and SL also need to be specified to MPI at job startup. Some examples of this are shown in the `/opt/iba/src/mpi_apps/ofed*.params` files that are provided with Intel® FastFabric Toolset.

When using MPI with the Intel® PSM API, path record queries can be enabled in conjunction with the Distributed SA. In which case there is no need to manually specify the PKey and SL at MPI job startup. Refer to the *Intel® True Scale Fabric OFED+ Host Software User Guide* for more information about enabling Path Record queries in PSM.

5.2.3.8 Pre-Created Multicast Groups

The `Multicast.Multicast Group` section of the `ifs_fm.xml` configuration file can specify multicast groups that should be pre-created by the SM. If neither a VirtualFabric nor PKey is specified for a given pre-created MulticastGroup, it is created for all vFabrics that contain the given MGID as an application.

When no MGIDs are explicitly specified, the necessary IPoIB multicast groups for IPv4 and IPv6 are pre-created against the selected VirtualFabric/PKeys (all applicable vFabrics if no specific VirtualFabric/PKey selected). When such automatic pre-creation occurs, the PKey assigned to the vFabric is inserted into the MGIDs per the IPoIB standard.

An example of this capability is provided in the sample configuration file.

5.2.3.9 Securing the Default Partition

When using a secured default partition with vFabrics and redundant FMs, it is recommended to explicitly specify the nodes/ports running the FMs as Members of the default partition's vFabric. This is preferred over using the `AllSMs` parameter because, upon failure of the SM, other parts of the FM on the given node may not be able to perform their functions.

Similarly, if using Intel® FastFabric Toolset in conjunction with a secure default partition, it will be necessary to specify the nodes/ports running Intel® FastFabric Toolset as Members of the default partitions. Failure to do so limits the operations that Intel® FastFabric Toolset can perform and the nodes that Intel® FastFabric Toolset can manage.

5.2.3.10 Multiple vFabrics with Same PKey

When multiple vFabrics are specified with the same PKey, they share a single PKey. When this occurs, the security for the vFabrics is the logical "OR" of the security for the two. If security is off in both, there are no limited members (only full members). If security is on for either (or both), security is imposed for both.

When two vFabrics share the same PKey, the list of members is the combined list from both vFabrics. `Members` is the sum of members in both, and `LimitedMembers` is the sum of limited members in both.

5.2.3.11 Multiple vFabrics with Same BaseSL

When multiple vFabrics are specified with the same BaseSL, they share a single SL. When this occurs, the QoS for the vFabrics is the logical "OR" of the QoS for the two.

5.2.3.12 Parameters

Table 41. VirtualFabric Parameters

Parameter	Description
Name	Name for VirtualFabric. Every VirtualFabric must have a unique name. The name must be unique among all VirtualFabric names within an `FM` instance. When defined at Common level must be unique within all instances. The name is limited to 64 characters and is case sensitive.
Enable	Enable (1) or Disable (0) a vFabric. When Disabled (0), the VirtualFabric is ignored. This allows the user to easily disable a VirtualFabric without deleting its definition.
QoS	When On (1), the Subnet Manager provides QoS for this vFabric and between other vFabrics. When Off (0), the Subnet Manager is free to manage SLs and VLs as it chooses, and there are no guarantees.
HighPriority	If set to 1, this indicates the vFabric is for high-priority traffic that does not require any bandwidth limiting. This would typically include management or control traffic, which is low bandwidth, but critical to process in a timely manner. An example is SA traffic where there is no reason to restrict bandwidth since it is low volume, but it needs to be serviced at a high priority. When priority is set to High, any bandwidth allocation is ignored for this vFabric.
Bandwidth	This is the minimum percentage (1–100%) of bandwidth that should be given to this vFabric relative to other low-priority vFabrics. When there is no contention, this vFabric could get more than this amount. If unspecified, the SM evenly distributes the remaining bandwidth among all the vFabrics with unspecified bandwidth. Total bandwidth for all enabled vFabrics with QoS enabled must not exceed 100%. If HighPriority is specified, this field is ignored.
PktLifeTimeMult	Amount to multiply PktLifeTime by when reported by SM for this vFabric. This can permit extra time in PathRecords (and therefore end-to-end timeouts) to account for delays in low-priority vFabrics that are given low-bandwidth allocations. The value is rounded up to the next power of two. 0 is invalid; default is 1.
SecondaryRouteOnly	When the dor-updown RoutingAlgorithm is in use, multiple SLs are required for DOR functionality. SecondaryRouteOnly can be enabled for a vFabric to limit the number of SLs required for the vFabric to one. When this is enabled, the vFabric members are limited to using the secondary route. For dor-updown, this is the updown route (which requires only one SL) and that allows more QoS vFabrics to be configured. This parameter is ignored if the RoutingAlgorithm is configured as shortestpath. The default is 0.

Table 41. VirtualFabric Parameters (Continued)

Parameter	Description
BaseSL	Allows a specific SL (0–15) to be used for the vFabric. When dor-updown routing is used, additional SLs may also be used for routing purposes. SM selects value if unspecified.
Security	When On (1), the Subnet Manager provides security within this vFabric and between other vFabrics. `LimitedMembers` cannot talk to each other in the vFabric. When Off (0), the Subnet Manager is free to manage routes and PKeys as it chooses, and there are no guarantees. `LimitedMembers` can talk to each other in the vFabric.
Members	A `DeviceGroup.Name` that should be in the VirtualFabric. Can be specified more than once per VirtualFabric if required.
LimitedMembers	A `DeviceGroup.Name` with limited membership in the VirtualFabric. When Security is off, this is functionally the same as a `DeviceGroup` specified using Members. Can be specified more than once per VirtualFabric if required.
Application	An `Application.Name` that should be in the VirtualFabric. Can be specified more than once per VirtualFabric if required.
PKey	Partition Pkey to use for vFabric. By default the Subnet Manager picks an available PKey. However, if required, a user-selected PKey can be specified. The PKey is a 16-bit value, and the high bit is ignored. The Subnet Manager uses the appropriate high-bit based on the Security and `Member`/`LimitedMember` status per device.
MaxMTU	Maximum MTU for SM to return in any PathRecord or Multicast group for the VirtualFabric. Actual values returned may be further reduced by hardware capabilities or if the `PathRecord` or `Multicast` group is requested to have a smaller MTU. However, SM considers it an error to create a `Multicast` group with MTU larger than that of the VirtualFabric. The value can also be stated as Unlimited. If not specified, the default MaxMTU is unlimited.
MaxRate	Maximum static rate for SM to return in any PathRecord or Multicast group for the VirtualFabric. Actual values returned may be further reduced by hardware capabilities or if the `PathRecord` or `Multicast` group is requested to have a smaller rate. However, SM considers it an error to create a `Multicast` group with rate larger than that of the VirtualFabric. The value can also be stated as Unlimited. If not specified, the default MaxRate is unlimited.
Index	Unique VirtualFabric 16-bit numeric index By default the FM picks an available VirtualFabric Index. However if required, a user-supplied Index can be specified. Specification may be necessary for applications that want to query VirtualFabrics by Index instead of by Name. The Index is returned as the record ID (RID) for VirtualFabric SA queries. The Index must be unique within the given FM instance.

The following is a `VirtualFabrics` section example:

```
<VirtualFabrics>

  <!--  Each VirtualFabric (VF) is comprised of: -->

  <!--  1. one or more Groups of devices -->
```

```
<!--  2. AND one or more sets of Applications -->
<!--  3. AND administrator policies -->
<!--  The Virtual Fabric will control the configuration of Security -->
<!--  for the given set of Devices and -->
<!--  Applications in the overall Fabric. -->
<!--  The Applications and DeviceGroups are matched against PathRecord and -->
<!--  Multicast SA queries so that the returned SLID/DLID, PKey, SL, etc -->
<!--  are consistent with the policies established for the VirtualFabric. -->
<!--  Devices in a DeviceGroup but not found in the fabric are ignored. -->
<!--  Every VirtualFabric must have a unique <Name> -->
<!--  The name must be unique among all VirtualFabric names within an -->
<!--  FM instance. -->
<!--  When defined at Common level must be unique within all instances. -->
<!--  The name is limited to 64 characters and is case sensitive. -->
<!--  Devices are specified by DeviceGroup Name in the following ways in -->
<!--  the VirtualFabric section: -->
<!--  As Full Members: <Member>group_name</Member> -->
<!--          such devices may talk to any other Member or LimitedMember -->
<!--  As Limited Members: <LimitedMember>group_name</LimitedMember> -->
<!--          When Security is 1 (On), LimitedMembers are not permitted -->
<!--          to talk to other LimitedMembers. -->
<!--          However LimitedMembers can always talk to Members. -->
<!--          If Security is 0 (Off) LimitedMembers are treated the same -->
<!--          as Members -->
<!--          This allows the user to easily turn off Security for a -->
<!--          Virtual Fabric without changing the rest of the definition. -->
<!--          When Security is on, LimitedMembers will not be able to -->
<!--          join multicast groups in the VirtualFabric. -->
<!--  Member and LimitedMember can each be specified more than once per -->
<!--  VirtualFabric if desired. -->
<!--  If a Device is in both the Members and LimitedMembers DeviceGroups, -->
<!--  they will be treated as Members. -->
<!--  This allows All to be specified as a LimitedMember, then selected -->
<!--  Members can be specified.  Hence insuring the VirtualFabric -->
```

```
<!--  includes All devices while allowing a limited set of Members. -->
<!--  Applications are specified by Application Name via: -->
<!--  <Application>application_name</Application> -->
<!--  Application can be specified more than once per VirtualFabric if -->
<!--  desired. -->
<!--  The following Administrator Policies and Controls can be specified: -->
<!--  Enable/Disable: <Enable>1</Enable> -->
<!--         When Disabled (0) the VirtualFabric is ignored. -->
<!--         This allows the user to easily disable a VirtualFabric -->
<!--         without deleting it's definition. -->
<!--  Security: <Security>1</Security> -->
<!--         When On (1) the FM will provide security within this VF -->
<!--         and between other VFs. -->
<!--         When On, LimitedMembers cannot talk to each other in the VF. -->
<!--         When Off (0) the FM is free to manage Routes and PKeys as -->
<!--         it chooses and there are no guarantees. -->
<!--         When Off (0) LimitedMembers can talk to each other in the VF. -->
<!--  Partition Key: <PKey>0x1234</PKey> -->
<!--         By default the FM will pick an available PKey. -->
<!--         However if desired a user selected PKey can be specified. -->
<!--         Specification may be necessary such that it is known apriori -->
<!--         for applications which do not use SA PathRecord queries, -->
<!--         such as MPIs which use non-IBTA compliant mechanisms for job -->
<!--         startup. -->
<!--         The PKey is a 16 bit value, the high bit will be ignored. -->
<!--         The FM will use the appropriate high bit based on the -->
<!--         Security and Member/LimitedMember status per device. -->
<!--  Max MTU: <MaxMTU>2048</MaxMTU> -->
<!--         Maximum MTU for SM to return in any PathRecord or -->
<!--         Multicast group for the VirtualFabric. -->
<!--         Actual values may be further reduced by Hardware -->
<!--         capabilities or if the PathRecord or Multicast group is -->
<!--         requested to have a smaller MTU. -->
<!--         However, SM will consider it an error to create a Multicast -->
```

```
<!--         group with MTU larger than that of the VirtualFabric. -->
<!--         The value can also be stated as Unlimited. -->
<!--         If not specified the default MaxMTU will be Unlimited. -->
<!--  Max Rate: <MaxRate>20g</MaxRate> -->
<!--         Maximum static rate for SM to return in any PathRecord or -->
<!--         Multicast group for the VirtualFabric. -->
<!--         Similar behaviors to MaxMTU. -->
<!--         The value can also be stated as Unlimited. -->
<!--         If not specified the default MaxRate will be Unlimited. -->
<!--  Unique VirtualFabric index: <Index>3</Index> -->
<!--         By default the FM will pick an available VirtualFabric Index. -->
<!--         However if desired a user supplied Index can be specified. -->
<!--         Specification may be necessary for applications which want -->
<!--         to query VirtualFabrics by Index instead of by Name. -->
<!--         The Index is a 16 bit value and is returned as the RID -->
<!--         for VirtualFabric SA queries. -->
<!--         The Index must be unique within the given FM instance. -->
<!--  IBTA requires a "default" partition. -->
<!--  To meet this requirement there must be an enabled VirtualFabric: -->
<!--  *  with a PKey of 0x7fff (or 0xffff) -->
<!--  *  it must include the SA Application -->
<!--  *  it must be the only enabled VF which includes the SA Application -->
<!--  *  it must have AllSMs as a Member -->
<!--  *  it must have All as a Member or LimitedMember -->
<!--  *  as desired it may have additional Applications -->
<!--  *  as desired it may have additional DeviceGroups as Member -->
<!--  *  all other VF policies (Security, MTU ...) may be set as desired -->
<!--  QOS settings -->
<!--  QOS Enable: <QOS>0</QOS> -->
<!--         0=disable, 1=enable, if 0 the QOS settings are ignored -->
<!--  High Priority: <HighPriority>0</HighPriority> -->
<!--         If set to one, this indicates high priority traffic which -->
<!--         does not require any bandwidth limiting. This would -->
<!--         typically include management or control traffic which is low -->
```

```
<!--         bandwidth, but critical to process in a timely manner. -->
<!--         An example is SA traffic where there is no reason to -->
<!--         restrict bandwidth since it is low volume, but it needs to -->
<!--         be serviced at a high priority. -->
<!--         When priority is set to High, any bandwidth allocation is -->
<!--         ignored for this Virtual Fabric. -->
<!--  Bandwidth Allocation: <Bandwidth>100%</Bandwidth> -->
<!--         1-100%   This is the minimum percentage of bandwidth which -->
<!--         should be given to this Virtual Fabric relative to other low -->
<!--         priority Virtual Fabrics.  When there is no contention, this -->
<!--         Virtual Fabric could get more than this amount. -->
<!--         If unspecified, the SM evenly distributes remaining -->
<!--         among all the Virtual Fabrics with unspecified Bandwidth. -->
<!--         Total Bandwidth for all enabled Virtual Fabrics with QOS -->
<!--         enabled must not exceed 100%. -->
<!--         If HighPriority is specified, this field is ignored. -->
<!--  Packet Lifetime Multiplier: <PktLifeTimeMult>2</PktLifeTimeMult> -->
<!--         Amount to multiply PktLifeTime by when reported by SM for -->
<!--         this Virtual Fabric. -->
<!--         This can permit extra time in PathRecords (and hence end to -->
<!--         end timeouts) to account for delays in Low Priority Virtual -->
<!--         Fabrics which are given low bandwidth allocations. -->
<!--         Value will be rounded up to the next power of 2. -->
<!--         0 is invalid, default is 1 -->
<!--  Secondary Route Only : <SecondaryRouteOnly>1</SecondaryRouteOnly> -->
<!--         When the dor-updown RoutingAlgorithm is in use, multiple SLs -->
<!--         are required for DOR functionality.  SecondaryRouteOnly can -->
<!--         be enabled for a VF to limit the number of SLs required for -->
<!--         the VF to one.  When this is enabled, the VF members -->
<!--         will be limited to using the secondary route. -->
<!--         For dor-updown, this is the updown route (which requires -->
<!--         only 1 SL) and will allow more QoS VFs to be -->
<!--         configured.  This parameter is ignored if the -->
<!--         RoutingAlgorithm is configured as shortestpath. -->
```

```
<!--         The default is 0 -->
<!--  Base Service Level: <BaseSL>0</BaseSL> -->
<!--         Allows a specific SL (0-15) to be used for the VF. -->
<!--         When dor-updown routing is used, additional SLs may also -->
<!--         be used for routing purposes. -->
<!--         SM selects value if unspecified -->
<!--  Some predefined VirtualFabrics follow.  These are also a good set -->
<!--  of examples to cut/paste to make other site specific VFs: -->
<!--  An example networking VF.  When IPoIB runs, it will use the -->
<!--  1st PKey on the given Port for the default (ib0 ...) network device -->
<!--  Hence its best to place the Networking/IPoIB VF 1st. -->
<!--  For additional IPoIB subnets which a node participates in, the PKey -->
<!--  needs to be manually specified to IPoIB as a "ipoib vlan" -->
<VirtualFabric>
  <Name>Networking</Name>
  <Enable>0</Enable>
  <PKey>0x1001</PKey>
  <Security>1</Security>
  <QOS>1</QOS>
  <Bandwidth>10%</Bandwidth>
  <Member>All</Member>
  <!--  can use a smaller DeviceGroup if desired -->
  <!--  Since IPoIB uses Multicast, LimitedMembers wouldn't make sense -->
  <Application>Networking</Application>
</VirtualFabric>
<!--  An IBTA default Partition with all Devices and Applications -->
<!--  This is the default Virtual Fabric config -->
<VirtualFabric>
  <Name>Default</Name>
  <Enable>1</Enable>
  <PKey>0xffff</PKey>
  <!--  must be IBTA default PKey -->
  <Security>0</Security>
  <QOS>0</QOS>
```

```
    <Member>All</Member>
    <Application>All</Application>
    <Application>SA</Application>
    <MaxMTU>Unlimited</MaxMTU>
    <MaxRate>Unlimited</MaxRate>
  </VirtualFabric>
  <!--  This vFabric can be used as a simple catchall.  When none of the -->
  <!--  other enabled vFabrics include the "All" Application, this will -->
  <!--  pick up the remaining applications.  This is also very useful when -->
  <!--  uncertain about how to identify applications of interest such as -->
  <!--  storage devices or file systems with undocumented Service IDs -->
  <VirtualFabric>
    <Name>AllOthers</Name>
    <Enable>0</Enable>
    <Security>1</Security>
    <QOS>1</QOS>
    <Bandwidth>20%</Bandwidth>
    <Member>All</Member>
    <!--  can be reduced in scope if desired -->
    <Application>AllOthers</Application>
  </VirtualFabric>
  <!--  An alternative more secure IBTA default Partition -->
  <!--  This gives non-SMs limited privileges -->
  <VirtualFabric>
    <Name>Admin</Name>
    <Enable>0</Enable>
    <PKey>0x7fff</PKey>
    <!--  must be IBTA default PKey -->
    <Security>1</Security>
    <QOS>1</QOS>
    <SecondaryRouteOnly>1</SecondaryRouteOnly>
    <HighPriority>1</HighPriority>
    <!--  admin traffic is High priority -->
    <Member>AllSMs</Member>
```

```
    <Member>AllSWE0s</Member>
    <!--  so managed spines can access leafs -->
    <!--  <Member>AdminNodes</Member> add more FF/admin nodes if desired -->
    <LimitedMember>All</LimitedMember>
    <Application>SA</Application>
    <!--  add other applications if desired -->
    <!--  <MaxMTU>256</MaxMTU> can reduce MTU, SA only uses MADs -->
  </VirtualFabric>
  <!--  This vFabric can be used as a simple catchall.  When none of the -->
  <!--  other enabled vFabrics include the "All" nor SA Application, this -->
  <!--  picks up the remaining applications.  This is also very useful when -->
  <!--  uncertain about how to identify applications of interest such as -->
  <!--  storage devices or file systems with undocumented Service IDs -->
  <VirtualFabric>
    <Name>AllOthersWithSA</Name>
    <Enable>0</Enable>
    <PKey>0x7fff</PKey>
    <!--  must be IBTA default PKey -->
    <Security>1</Security>
    <QOS>1</QOS>
    <Bandwidth>20%</Bandwidth>
    <!--  SecondaryRouteOnly>1</SecondaryRouteOnly> -->
    <Member>AllSMs</Member>
    <Member>AllSWE0s</Member>
    <!--  so managed spines can access leafs -->
    <!--  <Member>AdminNodes</Member> add more FF/admin nodes if desired -->
    <Member>All</Member>
    <!--  can be reduced in scope if desired -->
    <Application>AllOthers</Application>
    <Application>SA</Application>
  </VirtualFabric>
  <!--  a template for a storage Virtual Fabric -->
  <VirtualFabric>
    <Name>Storage</Name>
```

```
    <Enable>0</Enable>
    <Security>1</Security>
    <QOS>1</QOS>
    <Bandwidth>20%</Bandwidth>
    <!--  <Member>group_with_storage_targets</Member> -->
    <LimitedMember>All</LimitedMember>
    <Application>Storage</Application>
    <!--  <MaxRate>10g</MaxRate> individual QP only needs FC speed -->
  </VirtualFabric>
  <!--  a template for a compute Virtual Fabric -->
  <!--  QOS is configured for utilization of 70% of the bandwidth for MPI -->
  <VirtualFabric>
    <Name>Compute</Name>
    <Enable>0</Enable>
    <PKey>0x1002</PKey>
    <!--  manually specify so can supply to MPI -->
    <Security>1</Security>
    <QOS>1</QOS>
    <Bandwidth>70%</Bandwidth>
    <!--  <Member>compute_nodes</Member> -->
    <Member>All</Member>
    <!--  can be reduced in scope if desired -->
    <!--  if using the PSM vFabric examples below, use VerbsMPI not Compute -->
    <Application>Compute</Application>
    <!--  <Application>VerbsMPI</Application> -->
  </VirtualFabric>
  <!--  when using Intel True Scale HCAs with PSM technology, there is no -->
  <!--  need to configure PKey nor SL. FM can freely select since PSM -->
  <!--  will query the SA using the PSM ServiceIds when PSM_PATH_REC="opp" -->
  <VirtualFabric>
    <Name>PSM_Compute</Name>
    <Enable>0</Enable>
    <Security>1</Security>
    <QOS>1</QOS>
```

```
    <Bandwidth>70%</Bandwidth>
    <!--  <Member>compute_nodes</Member> -->
    <Member>All</Member>
    <!--  can be reduced in scope if desired -->
    <Application>PSM</Application>
  </VirtualFabric>
  <!--  when using Intel True Scale HCAs with PSM technology, Control and -->
  <!--  Data traffic can be separated into different vFabrics. -->
  <!--  To do this, PSM must be run with PSM_PATH_REC="opp" -->
  <!--  Then enable these two VFs instead of the MPI Compute ones above -->
  <VirtualFabric>
    <Name>PSM_Compute_Control</Name>
    <Enable>0</Enable>
    <Security>1</Security>
    <QOS>1</QOS>
    <HighPriority>1</HighPriority>
    <!--  admin traffic is High priority -->
    <!--  <Member>compute_nodes</Member> -->
    <Member>All</Member>
    <!--  can be reduced in scope if desired -->
    <Application>PSM_Control</Application>
  </VirtualFabric>
  <VirtualFabric>
    <Name>PSM_Compute_Data</Name>
    <Enable>0</Enable>
    <Security>1</Security>
    <QOS>1</QOS>
    <Bandwidth>70%</Bandwidth>
    <!--  <Member>compute_nodes</Member> -->
    <Member>All</Member>
    <!--  can be reduced in scope if desired -->
    <Application>PSM_Data</Application>
  </VirtualFabric>
</VirtualFabrics>
```

5.2.4 QoS Capabilities for Mesh/Torus Fabrics

When routing a Mesh or Torus fabric, multiple SLs and VLs are needed for routing for each vFabric. This is in contrast to Fat Tree and other topologies that are routed using the shortestpath algorithm. The shortestpath algorithm uses exactly 1 SL and VL per QoS-enabled vFabric.

The table in Appendix D, "QOS Options in a Mesh/Torus vFabric" shows various combinations of Mesh/Torus topologies and the QoS and MTU capabilities of the fabrics.

§ §

6.0 Embedded Fabric Manager Commands and Configuration

6.1 Viewing the Fabric

To determine if the embedded Fabric Manager can see the fabric, use the following CLI command:

```
smShowMasterLid
```

Alternatively more information about the fabric can be obtained using the following CLI command:

```
smShowLids
```

Similar information can also be obtained using the FastFabric commands:

```
fabric_info, iba_top, iba_saquery or iba_report.
```

Refer to the *Intel® True Scale Fabric Suite FastFabric Command Line Interface Reference Guide* for more information on these commands and their options.

6.1.1 Determining the Active Fabric Manager

In a configuration with multiple Intel® switches, use the following CLI command to determine the active Fabric Manager:

```
smShowMasterLid
```

To determine if this FM is the master, use the following CLI command:

```
smControl status
```

Similar information can also be obtained using the FastFabric CLI commands:

```
fabric_info, iba_top, iba_saquery or iba_report.
```

6.2 Subnet Management Group CLI Commands

The following pages define the group CLI commands and give the syntax, options and a sample output of each. With the exception of the `smConfig`, `smPmBmStart`, `smShowConfig` and `smResetConfig` commands, these commands only act against the Management Card presently logged into. In order to act on the FM running on the slave management card, it is necessary to login into that card.

6.2.1 Operational Commands

The following commands control the configuration and operation of the FM.

6.2.1.1 smControl

Starts and stops the embedded FM.

6.2.1.1.1 Synax

smControl [start | stop | restart | status]

6.2.1.1.2 Options

`start` – Start the FM.

`stop` - Stop the FM.

`restart` – Restarts the FM. (starts it if its not already running).

`status` – Prints out the FM Status.

6.2.1.1.3 Examples

```
-> smControl start


Starting the SM...
```

6.2.1.2 smConfig

Configure startup parameters of the embedded subnet manager.

6.2.1.2.1 Syntax

smConfig [query] [startAtBoot yes|no] [startOnSlaveCmu yes|no]

6.2.1.2.2 Options

`query` – Display present settings, no change.

`startAtBoot yes | no`
`yes` – Start the subnet manager at chassis boot.
`no` – Do not start the subnet manager at chassis boot.

`startOnSlaveCmu yes | no`
`yes` – Start the subnet manager on the slave CMU.
`no` – Do not start the subnet manager on the slave CMU.

6.2.1.2.3 Notes

Use this command to configure the subnet manager. Changes to these parameters will not take effect until the next reboot of the Chassis Management Card(s).

This command can only be run on the master Chassis Management Card.

6.2.1.2.4 Examples

```
-> smconfig query

startAtBoot: no

startOnSlaveCmu: no
```

```
->
```

-or-

```
-> smConfig
Start at boot? [Y]
Start on slave CMU? [N]
->
```

-or-

```
-> smconfig startAtBoot yes startOnSlaveCmu yes
Saving....
Saving complete...
```

6.2.1.3 smPmBmStart

Set/display whether the PM and BM will start with the SM.

6.2.1.3.1 Syntax

smPmBmStart [enable | disable | pm | bm | none]

6.2.1.3.2 Options

`enable` – Enable the start of the PM and BM at SM start-up.

`disable` – Disable the start of the PM and BM at SM start-up.

`pm` – Enable start of PM and FE, and disable BM at SM start-up.

`bm` – Enable start of BM and FE, and disable PM at SM start-up.

`none` – Disable start of PM, BM, and FE at SM start-up.

6.2.1.3.3 Notes

The configuration can only be changed from the master Chassis Management Card.

6.2.1.3.4 Examples

```
-> smPmBmStart
SM is enabled
PM is enabled
BM is enabled
FE is enabled

-> smPmBmStart disable
```

```
SM is enabled

PM is disabled

BM is disabled

FE is enabled
```

6.2.1.4 smResetConfig

Reset the XML configuration for the embedded subnet manager to factory defaults.

6.2.1.4.1 Syntax

smResetConfig [-noprompt]

6.2.1.4.2 Options

`-noprompt` – Proceed with configuration reset without a confirmation prompt.

6.2.1.4.3 Notes

This command is only available on the Master Chassis Management Card.

6.2.1.4.4 Examples

```
-> smResetConfig


Proceed with configuration reset? [N] Y


Default XML configuration file has been generated.
```

6.2.1.5 smShowConfig

Display the XML configuration file.

6.2.1.5.1 Syntax

smShowConfig [-infoOnly | -contentOnly] [-noprompt]

6.2.1.5.2 Options

`-infoOnly` – Display timestamp for XML configuration file.

`-contentOnly` – Display contents of XML configuration file.

`-noprompt` – Don’t prompt to ‘Continue?’ for each page of display.

6.2.1.5.3 Notes

With no arguments, the XML configuration file timestamp and contents will be displayed, one screen at a time. Enter 'Y' or 'Enter' at the prompt to continue displaying command output. Enter 'N' at the prompt to terminate the output. The `-infoOnly` and `-contentOnly` flags will limit what gets displayed. Use the `-noprompt` flag to send all output to the screen at once.

This command is only available on the master Chassis Management Card.

6.2.1.5.4 Examples

```
->smShowConfig -infoOnly
XML config file loaded 09:43:07    04/09/2009
```

-or-

```
->smShowConfig
XML config file loaded 09:43:0704/09/2009
<?xml version="1.0" encoding="utf-8"?>
<Config>

<!-- Common FM configuration, applies to all FM instances/subnets -->
<Common>
<!-- Various sets of Applications which may be used in Virtual Fabrics -->
<!Applications defined here are available for use in all FM instances. -->
<!-- Additional Applications may be defined here or per FM instance. -->
<!-- Applications specified per FM instance will add to -->
<!-- instead of replace those Application definitions. -->
<Applications>
...
...
...
Continue? [Y]
```

6.2.1.6 smForceSweep

Force a fabric sweep by the embedded subnet manager.

6.2.1.6.1 Syntax

smForceSweep

6.2.1.6.2 Options

None.

6.2.1.6.3 Notes

This command has no output message. To see the resulting sweep information, the "Info" level log messages must be turned on.

6.2.1.6.4 Examples

```
-> smForceSweep
```

6.2.1.7 bmForceSweep

Force a fabric sweep by the embedded baseboard manager.

6.2.1.7.1 Syntax

bmForceSweep

6.2.1.7.2 Options

None.

6.2.1.7.3 Notes

Use this command to force a sweep by the baseboard manager.

6.2.1.7.4 Examples

```
-> bmForceSweep
```

6.2.1.8 smRestorePriority

Restore normal priorities from elevated states for the SM, PM, and BM.

6.2.1.8.1 Syntax

smRestorePriority [sm|bm|all]

6.2.1.8.2 Options

`sm` – Restore normal SM priority.

`bm` – Restore normal BM priority.

`all` – Restore normal priorities for the SM, BM, and PM.

6.2.1.8.3 Notes

This command restores the normal priorities of the various FMs after they have elevated their priority as a result of a failover. Issuing this command allows the 'unsticking' of a sticky failover. Issuing this command without arguments will restore the normal priorities of the SM and BM. The priority of the PM is based on the priority of the SM.

6.2.1.8.4 Examples

```
-> smRestorePriority
```

6.2.2 FM Queries

The following commands query the state of the fabric and the fabric manager. The information provided in most of these queries can also be obtained by FastFabric commands such as `iba_report`, `iba_saquery`, `fabric_info` or `iba_showmc`.

6.2.2.1 smShowLids

Display all fabric LID information as known by the subnet manager.

6.2.2.1.1 Syntax

smShowLids

6.2.2.1.2 Options

None.

6.2.2.1.3 Notes

Use this command to display the current LID assignments for the devices in the True Scale Fabric. This command requires the given chassis to be the master FM.

Similar information can also be obtained using the FastFabric commands:

- `iba_saquery`
- `iba_report`

6.2.2.1.4 Examples

```
sm_state = MASTER   count = 572781   LMC = 0, Topology Pass count = 339, Priority =
0, Mkey = 0x0


-------------------------------------------------------------------------------
-------

SilverStorm 9080 GUID=0x00066a00da000100 172.26.2.2 Spine 1, Ch

-------------------------------------------------------------------------------
-------

Node[  0] => 00066a000600013c  (2)  ports=24, path=

        Port ----  GUID  ----  (S)   LID        LMC     _VL_   __MTU__  _WIDTH_
____SPEED____  CAP_MASK   N#  P#

         0 00066a000600013c   4  LID=0001  LMC=0000   8  8   2k  2k   4X  4X   2.5
2.5   00000a4a    0   0

           4 0000000000000000   4                     8  8   2k  2k   4X  4X  2.5/
5    5.0   00000000    1  22  4 22

           5 0000000000000000   4                     8  8   2k  2k   4X  4X  2.5/
5    5.0   00000000    1  23  4 23

           6 0000000000000000   4                     8  8   2k  2k   4X  4X  2.5/
5    5.0   00000000    1  24  4 24

           7 0000000000000000   4                     8  8   2k  2k   4X  4X  2.5/
5    5.0   00000000    2  24  7 24

           8 0000000000000000   4                     8  8   2k  2k   4X  4X  2.5/
5    5.0   00000000    2  23  7 23

           9 0000000000000000   4                     8  8   2k  2k   4X  4X  2.5/
5    5.0   00000000    2  22  7 22

          22 0000000000000000   4                     8  8   2k  2k   4X  4X  2.5/
```

```
5    5.0   00000000    3  16 22 16

          23 0000000000000000   4                      8  8    2k  2k   4X  4X  2.5/
5    5.0   00000000    3  18 22 18

          24 0000000000000000   4                      8  8    2k  2k   4X  4X  2.5/
5    5.0   00000000    3  17 22 17


-------------------------------------------------------------------------------
-------

st19

-------------------------------------------------------------------------------
-------

Node[ 10] => 00066a009800035a  (1)  ports=2, path= 7  4

        Port ----  GUID  ----  (S)   LID        LMC     _VL_    __MTU__  _WIDTH_
____SPEED____   CAP_MASK   N#  P#

         1 00066a00a000035a   4  LID=009f  LMC=0000   4  4    2k  2k   4X  4X   2.5
2.5   02010048    2   4  7  4
```

6.2.2.2 smShowGroups

Display multicast group information in the embedded subnet manager.

6.2.2.2.1 Syntax

smShowGroups [-h]

6.2.2.2.2 Options

`-h` – Display the host name as part of the output.

6.2.2.2.3 Notes

Use this command to display multicast group information in the subnet manager. This command is not available unless the SM is in MASTER mode.

Similar information can also be obtained using the FastFabric command:

- `iba_showmc`

6.2.2.2.4 Examples

```
-> smShowGroups

Multicast Groups:

  join state key: F=Full N=Non S=SendOnly Member


0xff12601bffff0000:00000001ffffd5bb (c001)

  qKey = 0x00000000  pKey = 0xFFFF  mtu = 4  rate = 3  life = 19  sl = 0

  0x0011750000ffd5bb F
```

```
0xff12401bffff0000:00000000ffffffff (c000)
  qKey = 0x00000000  pKey = 0xFFFF  mtu = 4  rate = 3  life = 19  sl = 0
  0x00066a01a0007116 F  0x0002c902003fffd5 F  0x00066a00a00001ac F
  0x00066a01a000015d F  0x00066a00a00001a3 F  0x00066a00a00001dc F
  0x00066a00a000035a F  0x0011750000ffd5c2 F  0x0011750000ffd664 F
  0x0011750000ffd9c2 F  0x0011750000ffd9f8 F  0x0011750000ffd5b9 F
  0x0011750000ffda4a F  0x0011750000ffd5bb F  0x0011750000ffd9de F
```

6.2.2.3 smShowServices

Display subnet administration service records of the subnet manager.

6.2.2.3.1 Syntax

smShowServices

6.2.2.3.2 Options

None.

6.2.2.3.3 Notes

The components (fields) of each service record are displayed. Each service record is stored in a location identified by a 'Slot' number which is displayed before any component of that Service Record. If a group of slots do not contain Service Records, the first slot of the empty group is displayed as 'empty'. This command states that the SM is in the STANDBY mode if the SM is not in MASTER mode.

Similar information can also be obtained using the FastFabric command:

- `iba_saquery -o service`

6.2.2.3.4 Examples

```
-> smShowServices
*****************************************************************
                 There is 1 Service Records
*****************************************************************
Service ID          = 0x1100D03C34834444
Service GID         = 0xFE80000000000000:00066A000600013C
Service P_Key       = 0x0000
Service Lease       = infinite
Service Key         =
0x00 0x00 0x00 0x00  0x00 0x00 0x00 0x00  0x00 0x00 0x00 0x00  0x00 0x00 0x00 0x00
Service Name        = SilverStorm Fabric Executive service Rev 1.1
```

```
Service Data 8      =

0x00 0x00 0x00 0x00  0x00 0x00 0x00 0x00  0x00 0x00 0x00 0x00  0x00 0x00 0x00 0x00

Service Data 16     =

0x0000 0x0000  0x0000 0x0000  0x0000 0x0000  0x0000 0x0000

Service Data 32     =

0x0000 0x0000 0x0000 0x0000

Service Data 64     =

0x0000000000000000 0x0000000000000000

Service Expire Time = 0x0100000000000000
```

6.2.2.4 smShowSubscriptions

Display event forwarding (subscription) table in the embedded subnet manager.

6.2.2.4.1 Syntax

smShowSubscriptions

6.2.2.4.2 Options

None.

6.2.2.4.3 Notes

Use this command to display the event forwarding (subscription) table in the subnet manager. This command states that the SM is in the STANDBY mode if the SM is not in MASTER mode.

Similar information can also be obtained using the FastFabric command:

- `iba_saquery -o inform`

6.2.2.4.4 Examples

```
-> smShowSubscriptions

*****************************************************************

                There are 2 subscriptions

*****************************************************************

Subscriber GID         = 0xFE80000000000000:00066A00D8000163

Subscriber LID         = 0x0071

Subscriber PKey        = 0xFFFF

Subscriber Start LID   = 0x0001

Subscriber End LID     = 0xBFFF

Subscriber Record ID   = 0x00000001

Subscriber Inform Info =
```

```
GID                  = 0x0000000000000000:0000000000000000
Start LID            = 0xFFFF
End LID              = 0x0000
Is Generic?          = Yes
Subscribe?           = Subscribe
Type                 = All Types
Trap Number            = 0x0040
Queue Pair Number      = 0x000001
Response Time Value    = 19
Producer Type          = Subnet Management
*******************************************************************
Subscriber GID         = 0xFE80000000000000:00066A01A0007116
Subscriber LID         = 0x0007
Subscriber PKey        = 0xFFFF
Subscriber Start LID   = 0x0001
Subscriber End LID     = 0xBFFF
Subscriber Record ID   = 0x00000036
Subscriber Inform Info =
GID                  = 0x0000000000000000:0000000000000000
Start LID            = 0xFFFF
End LID              = 0x0000
Is Generic?          = Yes
Subscribe?           = Subscribe
Type                 = All Types
Trap Number            = 0x0043
Queue Pair Number      = 0x000001
Response Time Value    = 18
Producer Type          = Channel Adapter
*******************************************************************
                There are 2 subscriptions
```

6.2.2.5 smShowMasterLid

Display the LID of the Master subnet manager.

6.2.2.5.1 Syntax

smShowMasterLid

6.2.2.5.2 Options

None.

6.2.2.5.3 Notes

Use this command to display the Master SM's LID, which might not be this SM's lid.

Similar information can also be obtained using the FastFabric commands:

- `fabric_info`
- `iba_saquery -o sminfo`
- `iba_report -o comps -F sm`

6.2.2.5.4 Examples

```
-> smShowMasterLid

The Master SM LID is 0x0001
```

6.2.2.6 smShowLid

Display the LID of this subnet manager.

6.2.2.6.1 Syntax

smShowLid

6.2.2.6.2 Options

None.

6.2.2.6.3 Notes

Use this command to display this SM's LID.

Similar information can also be obtained using the FastFabric commands:

- `fabric_info`
- `iba_saquery -o sminfo`
- `iba_report -o comps -F sm`

6.2.2.6.4 Examples

```
-> smShowLid

The SM LID is 0x0001
```

6.2.2.7 smShowLidMap

Display the LID-to-port GUID map for the subnet manager.

6.2.2.7.1 Syntax

smShowLidMap

6.2.2.7.2 Options

None

6.2.2.7.3 Notes

Use this command to display the LID-to-port GUID map of the subnet manager. The pass count for a LID is incremented each time the SM sweep detects that LID.

If LMC has been used to assign multiple LIDs to a node, those assignments will be reflected in the smShowLidMap output. This command states that the SM is in the STANDBY mode if the SM is not in MASTER mode.

Similar information can also be obtained using the FastFabric commands:

- `iba_saquery`
- `iba_report -o lids`

6.2.2.7.4 Examples

```
-> smShowLidMap

-------------------------------------------------------------------------------

SM is currently in the MASTER state, with Topology Pass count = 341

-------------------------------------------------------------------------------

Lid 0x0001: guid = 0x00066a000600013c, pass = 341, SilverStorm 9080
GUID=0x00066a00da000100 172.26.2.2 Spine 1, Ch

Lid 0x0002: guid = 0x00066a0007000170, pass = 341, SilverStorm 9080
GUID=0x00066a00da000100 172.26.2.2 Leaf 4, Chi

Lid 0x0003: guid = 0x00066a100600013c, pass = 341, SilverStorm 9080
GUID=0x00066a00da000100 172.26.2.2 Spine 1, Ch

Lid 0x0006: guid = 0x00066a00a0000248, pass = 229

Lid 0x0007: guid = 0x00066a01a0007116, pass = 341, st149

Lid 0x0008: guid = 0x0000000000000000, pass = 0

Lid 0x0027: guid = 0x00066a026000016c, pass = 341, VFx in Chassis
0x00066a0050000135, Slot 5

Lid 0x0028: guid = 0x0000000000000000, pass = 0

Lid 0x0029: guid = 0x00066a0260000174, pass = 341, VFx in Chassis
0x00066a000100024d, Slot 2

Lid 0x002a: guid = 0x0000000000000000, pass = 0
```

6.2.2.8 smShowMaxLid

Display the highest LID allocated by the subnet manager.

6.2.2.8.1 Syntax

smShowMaxLid

6.2.2.8.2 Options

None.

6.2.2.8.3 Notes

Use this command to display the highest LID allocated by the subnet manager. This command is not available unless the SM is in MASTER mode.

Similar information can also be obtained using the FastFabric commands:

- `iba_saquery -o lid|sort`
- `iba_report -o lids`

6.2.2.8.4 Examples

```
-> smShowMaxLid

The maximum LID is 0x0138
```

6.2.2.9 smPKeys

Display partition keys (PKeys) in the PKey table.

6.2.2.9.1 Syntax

smPKeys

6.2.2.9.2 Options

None.

6.2.2.9.3 Notes

PKeys are used for partitioning the subnet. Only configure PKeys if the host driver supports this. Invalid configuration of the PKey may render the fabric inoperable. The smPKeys command will display the PKey values for each virtual fabric. Subnet manager must be running to display PKeys.

6.2.2.9.4 Examples

```
-> smPKeys

Virtual Fabric: Default PKey: 0xffff

Virtual Fabric: NetworkingPKey: 0x1234
```

6.2.2.10 smShowRemovedPorts

Display ports that have been automatically removed from the fabric.

6.2.2.10.1 Syntax

smShowRemovedPorts

6.2.2.10.2 Options

None.

6.2.2.10.3 Notes

This displays ports that have been removed from the fabric automatically by the SM, such as when a 1x link mode is set to 'ignore' or when a port has exceeded its urgent trap threshold. This command states that the SM is in the STANDBY mode if the SM is not in the MASTER mode.

6.2.2.10.4 Examples

```
-> smShowRemovedPorts

Disabled Ports:
```

6.2.2.11 smShowCounters

Display various statistics and counters maintained by the SM.

6.2.2.11.1 Syntax

smShowCounters

6.2.2.11.2 Options

None.

6.2.2.11.3 Examples

The following information is displayed if the SM is in the MASTER node:

```
-> smShowCounters


                              COUNTER: THIS SWEEP LAST SWEEP      TOTAL
---------------------------------- ---------- ---------- ----------
   SM State transition to DISCOVERY:          0          0          1
      SM State transition to MASTER:          0          0          1
     SM State transition to STANDBY:          0          0          0
    SM State transition to INACTIVE:          0          0          0
      Total transmitted SMA Packets:          0         29      10563
          Direct Routed SMA Packets:          0         29      10562
             LID Routed SMA Packets:          0          0          0
              SMA Query Retransmits:          0          0          0
    SMA Query Retransmits Exhausted:          0          0          0
                  SM TX GET(Notice):          0          0          0
                  SM TX SET(Notice):          0          0          0
                 SM RX TRAP(Notice):          0          0          0
            ....    ....
            ....    ....
            ....    ....
```

6.2.2.12 smResetCounters

Resets various statistics and counters maintained by the SM.

6.2.2.12.1 Syntax

smResetCounters

6.2.2.12.2 Options

None.

6.2.2.13 pmShowCounters

Display various statistics and counters maintained by the PM.

6.2.2.13.1 Syntax

pmShowCounters

6.2.2.13.2 Options

None.

6.2.2.13.3 Examples

```
-> pmShowCounters
```

COUNTER:	THIS SWEEP	LAST SWEEP	TOTAL
PM Sweeps:	0	1	32039
Ports whose PMA failed query:	0	0	206
Nodes with 1 or more failed Ports:	0	0	197
Total transmitted PMA Packets:	0	228	7307203
PMA Query Retransmits:	0	0	7418
PMA Query Retransmits Exhausted:	0	0	206
PM TX GET(ClassPortInfo):	0	0	98
PM TX GET(PortSamplesControl):	0	0	0
PM TX GET(PortSamplesResult):	0	0	0
PM TX GET(PortCounters):	0	172	5507335
PM TX SET(PortCounters):	0	35	1119563
PM TX GET(PortCountersExtended):	0	11	352409
PM TX GET(VendorPortCounters):	0	0	0
PM TX SET(VendorPortCounters):	0	10	320380
PM RX GETRESP(*):	0	228	7299579
PM RX STATUS BUSY:	0	0	0
PM RX STATUS REDIRECT:	0	0	0
PM RX STATUS BADCLASS:	0	0	0

```
                PM RX STATUS BADMETHOD:          0          0          0
             PM RX STATUS BADMETHODATTR:          0          0          0
                  PM RX STATUS BADFIELD:          0          0          0
                   PM RX STATUS UNKNOWN:          0          0          0
               PA RX GET(ClassPortInfo):          0          0          0
                     PA RX GET(GrpList):          0          0         13
                     PA RX GET(GrpInfo):          0          0         82
              ....    ....
              ....    ....
              ....    ....
```

6.2.2.14 pmResetCounters

Resets various statistics and counters maintained by the PM.

6.2.2.14.1 Syntax

pmResetCounters

6.2.2.14.2 Options

None.

6.2.2.15 pmShowRunningTotals

Display Running Total PMA counters retained in PM for every port in fabric.

6.2.2.15.1 Syntax

pmShowRunningTotals

6.2.2.15.2 Options

None.

6.2.2.15.3 Examples

```
-> pmShowRunningTotals
admin1 HCA-1 Guid 0x00066a009800ec5b LID 0x1 Port 1
    Neighbor: SilverStorm 9040 GUID=0x00066a00db000066 L02 Guid 0x00066a00070014
dc LID 0xb Port 1
    Rate:  20g MTU: 2048
    Xmit: Data:0          MB (2664       Quads) Pkts:37
    Recv: Data:0          MB (2592       Quads) Pkts:36
    Integrity:                        SmaCongest:
```

```
      Symbol:0                        VL15 Dropped:0
      Link Recovery:0
      Link Downed:1
      Port Rcv:0                    Security:
      Loc Lnk Integrity:0             Port Rcv Constrain:0
      Excess Bfr Overrun*:0           Port Xmt Constrain*:0
    Congestion:                     Routing:
      Port Xmt Discards*:7            Port Rcv Sw Relay:0
      Port Xmt Congest*:0
    Port Rcv Rmt Phy:0              Port Adapt Route:0
    Port Xmt Congest:0              Port Check Rate:0
admin1 HCA-1 Guid 0x00066a009800ec5b LID 0x2 Port 2
    Neighbor: SilverStorm 9040 GUID=0x00066a00db000066 L02 Guid 0x00066a00070014
dc LID 0xb Port 11
    Rate:  20g MTU: 2048
    Xmit: Data:0          MB (2232       Quads) Pkts:31
    Recv: Data:0          MB (2232       Quads) Pkts:31
    Integrity:                      SmaCongest:
      Symbol:0                        VL15 Dropped:1
      Link Recovery:0
      Link Downed:1
      Port Rcv:0                    Security:
      Loc Lnk Integrity:0             Port Rcv Constrain:0
      Excess Bfr Overrun*:0           Port Xmt Constrain*:0
    Congestion:                     Routing:
      Port Xmt Discards*:0            Port Rcv Sw Relay:0
      Port Xmt Congest*:0
    Port Rcv Rmt Phy:0              Port Adapt Route:0
    Port Xmt Congest:0              Port Check Rate:0........
           ....    ....
           ....    ....
```

6.2.3 FM Configuration Queries

The following commands query the individual configuration attributes of the FM. Some of these commands can also make non-persistent changes to the FM. Any such non-persistent changes will be lost the next time the FM or chassis is restarted. To make persistent changes to the FM configuration the desired `ifs_fm.xml` file must be downloaded to the chassis.

6.2.3.1 smShowSMParms

Display subnet manager parameters switch lifetime, HOQ lifetime, VLStall val, pkt lifetime, and dynamic PLT.

6.2.3.1.1 Syntax

smShowSMParms

6.2.3.1.2 Options

None.

6.2.3.1.3 Notes

Use this command to display a sampling of subnet manager parameters.

6.2.3.1.4 Examples

```
-> smShowSMParms

SM priority is set to 4

SM LMC is set to 0

SM sweep rate is set to 300

SM max retries on receive set to 3

SM max receive wait interval set to 250 millisecs

switchLifetime set to 15

HoqLife set to 9

VL Stall set to 5

packetLifetime constant is set to 18

Dynamic PLT ON using values: 1 hop=16, 2 hops=17, 3 hops=17, 4 hops=18, 5 hops=18,
6 hops=18, 7 hops=18, 8+hops=19

SM DBSync interval set to 900

SM topology errors threshold set to 0, max retry to 3
```

6.2.3.2 smPriority

Display the priority of the subnet manager.

6.2.3.2.1 Syntax

smPriority

6.2.3.2.2 Options

None.

6.2.3.2.3 Notes

The priority of the Subnet Manager (SM) determines which subnet manager will become the master SM for the fabric. Zero is the lowest priority and fifteen is the highest. The SM with the highest priority will become the master SM for the fabric. The elevated priority value determines what the priority of the SM will be if it becomes Master. This allows persistent failovers that do not automatically fail back by configuring the elevated priority to be higher than all normal priorities. This feature is effectively disabled when set to the default of zero.Subnet manager must be running to display the priority.

Similar information can also be obtained using the FastFabric commands:

```
iba_saquery -o sminfo or iba_report -o comps.
```

6.2.3.2.4 Examples

```
-> smPriority


The SM Priority is 0

The SM Elevated Priority is disabled
```

6.2.3.3 bmPriority

Display the priority of the embedded baseboard manager.

6.2.3.3.1 Syntax

bmPriority

6.2.3.3.2 Options

None.

6.2.3.3.3 Notes

The priority of the Baseboard Manager (BM) determines which BM will become the master BM for the fabric. Zero is the lowest priority and fifteen is the highest. The BM with the highest priority will become the master BM for the fabric. The elevated priority value determines what the priority of the BM will be if it becomes Master. This allows persistent failovers that do not automatically fail back by configuring the elevated priority to be higher than all normal priorities. This feature is effectively disabled when set to the default of zero.Subnet manager must be running to display the priority.

6.2.3.3.4 Examples

```
-> bmPriority


The BM Priority is 0

The BM Elevated Priority is disabled
```

6.2.3.4 pmPriority

Display the priority of the embedded performance manager.

6.2.3.4.1 Syntax

pmPriority

6.2.3.4.2 Options

None.

6.2.3.4.3 Notes

This command is not supported any more. The priority of the Performance Manager (PM) is now based on the priority of the Subnet manager (SM). Reference the smPriority CLI command.

6.2.3.5 smSweepRate

Display/Dynamically set the sweep rate of the subnet manager.

6.2.3.5.1 Syntax

smSweepRate [*sweepRate*]

6.2.3.5.2 Options

sweepRate – The sweep rate (in seconds) of the subnet manager. Valid values are 3-86400, or 0 to turn the sweep off. The sweepRate is the interval between the end of one sweep and the start of the next sweep.

6.2.3.5.3 Notes

The sweep rate determines how often the subnet manager scans the fabric for changes and events.

The Subnet manager must be running to use this command. Changes made with this command will only affect the currently running SM, and will be forgotten if the FM is restarted or the chassis is rebooted. To make persistent changes edit the FM XML configuration file.

6.2.3.5.4 Examples

```
-> smSweepRate 200

The SM sweep rate has been dynamically set to 200 seconds

for the currently runing SM. To make a persistent change,

the xml config file must be modified and activated.
```

6.2.3.6 smMasterLMC

Display the Master SM's LMC value to be used on Channel Adapter ports.

6.2.3.6.1 Syntax

smMasterLMC

6.2.3.6.2 Options

None.

6.2.3.6.3 Notes

The value of the LMC determines how many LID's are assigned to an endport; 2^{LMC} LIDs are assigned to endports based on this value. For example, setting the LMC to a value of 3 will assign 2^3(example: 8) LID's per endport. Allowed LMC values are between zero and seven, inclusive. Subnet manager must be running as MASTER to display the LMC value.

6.2.3.6.4 Examples

```
-> smMasterLMC

Master SM LMC: 2 (4 LID(s) per port)
```

6.2.3.7 smSwitchLifetime

Display/Dynamically set the default switch lifetime (time a packet can live in a switch) of the subnet manager.

6.2.3.7.1 Syntax

smSwitchLifetime [*lifetime*]

6.2.3.7.2 Options

lifetime – The packet lifetime value between 0 and 31, inclusive.

6.2.3.7.3 Notes

The switch lifetime value determines the maximum time a packet may remain in a switch, calculated using the formula: 4.096 * (2 to the power of switchlifetime) microseconds.

The Subnet manager must be running to use this command. Changes made with this command will only affect the currently running SM, and will be forgotten if the FM is restarted or the chassis is rebooted. To make persistent changes edit the FM XML configuration file.

6.2.3.7.4 Examples

```
-> smSwitchLifetime 15

SM switch packet lifetime: 15 (~134217 microseconds)
```

6.2.3.8 smHoqLife

Display/Dynamically set the head of queue packet lifetime for switch ports.

6.2.3.8.1 Syntax

smHoqLife [*lifetime*]

6.2.3.8.2 Options

lifetime – The packet lifetime value between 0 and 31, inclusive.

6.2.3.8.3 Notes

Use this command to display or set the maximum lifetime that a packet may remain at the head of virtual lane's transmission queue before it is discarded by a switch, calculated using the formula: 4.096 * (2 to the power of Hoqlifetime) microseconds.

The Subnet manager must be running to use this command. Changes made with this command will only affect the currently running SM, and will be forgotten if the FM is restarted or the chassis is rebooted. To make persistent changes edit the FM XML configuration file.

6.2.3.8.4 Examples

```
-> smHoqLife 9

SM HOQ Liftime: 9 (~2097 microseconds)
```

6.2.3.9 smVLStall

Display/Dynamically set the VL stall value of the SM.

6.2.3.9.1 Syntax

smVLStall [*packets*]

6.2.3.9.2 Options

packets – The number of sequential packets dropped before port enters VL stalled state.

6.2.3.9.3 Notes

Use this command to display or dynamically set the VL stall value for ports in the fabric. This value determines the how quickly a virtual lane for a particular switch or endport enters a 'stalled' state after dropping packets.

The Subnet manager must be running to use this command. Changes made with this command will only affect the currently running SM, and will be forgotten if the FM is restarted or the chassis is rebooted. To make persistent changes edit the FM XML configuration file.

6.2.3.9.4 Examples

```
-> smVLStall 5

SM VL Stall Threshhold: 5 packets
```

6.2.3.10 smInfoKey

Display the subnet manager key (SMInfo) value.

6.2.3.10.1 Syntax

smInfoKey

6.2.3.10.2 Options

None.

6.2.3.10.3 Notes

Use this command to display the SM key. The key value is up to 8 byte hex. Subnet manager must be running to display the SMInfo key value.

6.2.3.10.4 Examples

```
-> smInfoKey

SM Key: 0x0000000000000001 (1 decimal).
```

6.2.3.11 smMgmtKey

Display the subnet manager management key (portInfo) value.

6.2.3.11.1 Syntax

smMgmtKey

6.2.3.11.2 Options

None.

6.2.3.11.3 Notes

Use this command to display the SM management key. The mkey value is up to 8 byte hex. Subnet manager must be running to display the mkey value.

6.2.3.11.4 Examples

```
-> smMgmtKey

SM management key: 0x0000000000000011 (17 decimal)
```

6.2.3.12 smOptionConfig

Use this command to display non-default modes of operation.

6.2.3.12.1 Syntax

smOptionConfig

6.2.3.12.2 Options

None.

6.2.3.12.3 Notes

Subnet manager must be running to display the non-default modes of operation.

6.2.3.12.4 Examples

```
-> smOptionConfig

        [dyn-plt] Dynamic packet lifetime support is enabled


Virtual Fabric: Default

Multicast Group: 0
```

```
    [def-mcgrp-create] Default multicast group auto-creation is enabled

Multicast Group: 1

    [def-mcgrp-create] Default multicast group auto-creation is enabled


Virtual Fabric: Networking

    Multicast Group: 0

    [def-mcgrp-create] Default multicast group auto-creation is enabled
```

6.2.3.13 smDefBcGroup

Display default multicast group configuration.

6.2.3.13.1 Syntax

smDefBcGroup

6.2.3.13.2 Options

None.

6.2.3.13.3 Notes

- Valid MTU values are 1(256), 2(512), 3(1024), 4(2048), and 5(4096)
- Valid RATE values are 2(2.5GB), 3(10GB), 4(30GB), 5(5GB), 6(20GB), 7(40GB), 8(60GB), 9(80GB), 10(120GB)
- Valid Values for SL is 0 (only value supported at this time)
- Valid Values for QKEY are 0 to 0xFFFFFFFF
- Valid Values for FlowLabel are 0 to 0xFFFFF
- Valid Values for TClass are 0 to 0xff

Values for each virtual fabric will be displayed. Subnet manager must be running to display this information.

6.2.3.13.4 Examples

```
-> smDefBcGroup

Virtual Fabric: Default

   Multicast Group: 0

        PKey: 0xffff

        MTU: 4

        Rate: 3

        SL: 0x0

        QKey: 0x00000000

        FlowLabel: 0x00000

        TClass: 0x00
```

```
        Auto-creation of default group at SM start-up is enabled
    Multicast Group: 1
        PKey: 0xffff
        MTU: 4
        Rate: 3
        SL: 0x0
        QKey: 0x00000000
        FlowLabel: 0x00000
        TClass: 0x00
        Auto-creation of default group at SM start-up is enabled
Virtual Fabric: Networking
    Multicast Group: 0
        PKey: 0xffff
        MTU: 4
        Rate: 3
        SL: 0x0
        QKey: 0x00000000
        FlowLabel: 0x00000
        TClass: 0x00
           Auto-creation of default group at SM start-up is enabled
```

6.2.3.14 smGidPrefix

Display the Subnet Prefix (default=0xfe80000000000000).

6.2.3.14.1 Syntax

smGidPrefix

6.2.3.14.2 Options

None.

6.2.3.14.3 Notes

Use this command to display the subnet prefix of the SM. The subnet prefix value is 8 byte hex. If there is more than 1 SM in the fabric, the GID Prefix must be the same on all SMs in order for the fabric to be functioning properly after an SM failover. The subnet manager must be running to display the subnet prefix.

6.2.3.14.4 Examples

```
-> smGidPrefix

Subnet Prefix: 0xfe80000000000001
```

6.2.3.15 smSubnetSize

Display the subnet size for the subnet manager.

6.2.3.15.1 Syntax

smSubnetSize

6.2.3.15.2 Options

None.

6.2.3.15.3 Notes

Use this command to display the configured fabric size. This should be expressed in terms of the upper limit of HCA ports on the subnet. Setting this value will not take effect until the Subnet Manager is restarted. Subnet manager must be running to display subnet size.

6.2.3.15.4 Examples

```
-> smSubnetSize

Subnet Size: 288
```

6.2.3.16 smTopoErrorThresh

Display the error threshold for a topology sweep.

6.2.3.16.1 Syntax

smTopoErrorThresh

6.2.3.16.2 Options

None.

6.2.3.16.3 Notes

Displays the maximum number of errors the SM may encounter during a sweep before abandoning the sweep. Subnet manager must be running to display the error threshold for a topology sweep.

6.2.3.16.4 Examples

```
-> smTopoErrorThresh

Topology error threshold:100
```

6.2.3.17 smTopoAbandonThresh

Display the max consecutive times the SM can abandon a sweep due to too many errors.

6.2.3.17.1 Syntax

smTopoAbandonThresh

6.2.3.17.2 Options

None.

6.2.3.17.3 Notes

Subnet manager must be running to display this information.

6.2.3.17.4 Examples

```
-> smTopoAbandonThresh

Topology sweep abandonment threshold: 3
```

6.2.3.18 smMaxRetries

Display maximum number of SM receive retries.

6.2.3.18.1 Syntax

smMaxRetries

6.2.3.18.2 Options

None.

6.2.3.18.3 Notes

Subnet manager must be running to display this information.

6.2.3.18.4 Examples

```
-> smMaxRetries

Max retries: 3
```

6.2.3.19 smRcvWaitTime

Display max time to wait for a reply to an SM packet in millisecs.

6.2.3.19.1 Syntax

smRcvWaitTime

6.2.3.19.2 Options

None.

6.2.3.19.3 Notes

Subnet manager must be running to display this information.

6.2.3.19.4 Examples

```
-> smRcvWaitTime

Recieve wait time: 250 milliseconds
```

6.2.3.20 smNonRespDropTime

Display seconds to wait before dropping a non-responsive node.

6.2.3.20.1 Syntax

smNonRespDropTime

6.2.3.20.2 Options

None.

6.2.3.20.3 Notes

Subnet manager must be running to display this information.

6.2.3.20.4 Examples

```
-> smNonRespDropTime

Non-responsive node drop time: 300 seconds
```

6.2.3.21 smNonRespDropSweeps

Display sweeps to wait before dropping a non-responsive node.

6.2.3.21.1 Syntax

smNonRespDropSweeps

6.2.3.21.2 Options

None.

6.2.3.21.3 Notes

Subnet manager must be running to display this information.

6.2.3.21.4 Examples

```
-> smNonRespDropSweeps

Non-responsive node drop sweeps: 3 sweeps
```

6.2.3.22 smMcLidTableCap

Display the limit of multicast LIDs available for allocation.

6.2.3.22.1 Syntax

smMcLidTableCap

6.2.3.22.2 Options

None.

6.2.3.22.3 Notes

A value of zero disables limiting multicast LIDs. Subnet manager must be running to display this information.

6.2.3.22.4 Examples

```
-> smMcLidTableCap

Mc lid limit: 1024
```

6.2.3.23 smMasterPingInterval

Displays SM ping interval in seconds.

6.2.3.23.1 Syntax

smMasterPingInterval

6.2.3.23.2 Options

None.

6.2.3.23.3 Notes

Value must be between 3 and 10. Subnet manager must be running to display this information.

6.2.3.23.4 Examples

```
-> smMasterPingInterval

Master ping interval: 4 seconds
```

6.2.3.24 smMasterPingFailures

Display number of master ping failures allowed.

6.2.3.24.1 Syntax

smMasterPingFailures

6.2.3.24.2 Options

None.

6.2.3.24.3 Notes

Value must be between 2 and 5. Subnet manager must be running to display this information.

6.2.3.24.4 Examples

```
-> smMasterPingFailures

Master ping failures: 3 failures
```

6.2.3.25 smDbSyncInterval

Display how often a Master SM should perform a full sync with standby SMs.

6.2.3.25.1 Syntax

smDbSyncInterval

6.2.3.25.2 Options

None.

6.2.3.25.3 Notes

Value must be between 0 and 60 minutes (0=OFF). Subnet manager must be running to display this information.

6.2.3.25.4 Examples

```
-> smDbSyncInterval

SM DB full sync interval currently set to 15 minutes


-----SM DB SYNCHRONIZATION interval set to 900 seconds, 2 SM(s) in fabric-----

MASTER SM node at SilverStorm 9024 DDR GUID=0x00066a00d90003fa, LID 0x0008,
PortGuid 0x00066a00d90003fa

     Sync Capability is  SUPPORTED

STANDBY SM node at st44, LID 0x0100, PortGuid 0x00066a00a0000357

     Sync Capability is  SUPPORTED

     Full sync status is      SYNCHRONIZED

     Time of last Full sync is THU APR 10 15:37:47 2008

     Time of last INFORM records sync is THU APR 10 15:37:47 2008

     Time of last GROUP records sync is THU APR 10 15:37:47 2008

     Time of last SERVICE records sync is THU APR 10 15:37:47 2008
```

6.2.3.26 smDynamicPlt

Display dynamic packet lifetime values.

6.2.3.26.1 Syntax

smDynamicPlt

6.2.3.26.2 Options

None.

6.2.3.26.3 Notes

Setting values to numbers greater than 19 give an effectively-infinite packet lifetime. Subnet manager must be running to display dynamic packet lifetime values.

6.2.3.26.4 Examples

```
-> smDynamicPlt

Index: 1 PLT Value: 16 (~268435 usec)

Index: 2 PLT Value: 17 (~536870 usec)

Index: 3 PLT Value: 17 (~536870 usec)

Index: 4 PLT Value: 18 (~1073741 usec)

Index: 5 PLT Value: 18 (~1073741 usec)
```

```
Index: 6 PLT Value: 18 (~1073741 usec)

Index: 7 PLT Value: 18 (~1073741 usec)

Index: 8 PLT Value: 19 (~2147483 usec)

Index: 9 PLT Value: 19 (~2147483 usec)


Dynamic packet lifetime values for pathrecord queries are enabled

(use the smOptionConfig command to change)
```

6.2.3.27 sm1xLinkMode

Display how the SM handles links that come up at 1x.

6.2.3.27.1 Syntax

sm1xLinkMode

6.2.3.27.2 Options

None.

6.2.3.27.3 Notes

When set to 'off', all links come up normally. When set to 'ignore', links that only come up at 1x (when they were enabled for a higher rate) are forced down. These downed ports can be queried to aid debugging errors in the fabric. Subnet manager must be running to display this information.

6.2.3.27.4 Examples

```
-> sm1xLinkMode

Mode is 'off'. Erroneous 1x links will be activated normally.
```

6.2.3.28 smTrapThreshold

Display the urgent trap threshold (in minutes) for port auto-disable.

6.2.3.28.1 Syntax

smTrapThreshold

6.2.3.28.2 Options

None.

6.2.3.28.3 Notes

When enabled, ports generating urgent traps at a rate higher than the threshold will be disabled. This value can range from 10 to 100 traps/minute. A value of zero disables this feature. Subnet manager must be running to display this value.

6.2.3.28.4 Examples

```
-> smTrapThreshold

Trap Threshold is 0 (disabled).
```

6.2.3.29 smLogLevel

Displays or dynamically sets the Subnet Manager logging level.

6.2.3.29.1 Syntax

smLogLevel [*loglevel*]

6.2.3.29.2 Options

loglevel – Logging level 1-7:

- 1 – WARN+
- 2 – INFINI_INFO+
- 3 – INFO+
- 4 – VERBOSE+
- 5 – DEBUG2+
- 6 – DEBUG3+
- 7 – TRACE+

6.2.3.29.3 Notes

The Subnet manager must be running to use this command. Changes made with this command will only affect the currently running SM, and will be forgotten if the FM is restarted or the chassis is rebooted. To make persistent changes edit the FM XML configuration file.

6.2.3.29.4 Examples

```
-> smLogLevel

Log Level:2
```

6.2.3.30 smLogMode

Displays or dynamically sets the Subnet Manager logging mode.

6.2.3.30.1 Syntax

smLogMode [*logmode*]

6.2.3.30.2 Options

logmode – Logging mode 0 or 1:

- 0 - use normal logging levels
- 1 - logging is quieted by downgrading the majority of fatal, error, warn and infiniinfo log messages to level
- 3 (INFO) and only outputting user actionable events when LogLevel is 1 or 2.

6.2.3.30.3 Notes

The Subnet manager must be running to use this command. Changes made with this command will only affect the currently running SM, and will be forgotten if the FM is restarted or the chassis is rebooted. To make persistent changes edit the FM XML configuration file.

6.2.3.30.4 Examples

```
-> smLogMode

Log Mode:0
```

6.2.3.31 smLogMask

Displays or dynamically sets the Subnet Manager logging mask for a specific subsystem.

6.2.3.31.1 Syntax

smLogMask *subsystem* [*mask*]

6.2.3.31.2 Options

subsystem – FM subsystem.

mask – Bit mask for logging to enable.

6.2.3.31.3 Notes

Subsystems: CS, MAI, CAL, DVR, IF3, SM, SA, PM, PA, BM, FE, APP

The Subnet manager must be running to use this command. Changes made with this command will only affect the currently running SM, and will be forgotten if the FM is restarted or the chassis is rebooted. To make persistent changes edit the FM XML configuration file.

6.2.3.31.4 Examples

```
-> smLogMask SA

SA Log Mask: 0x1ff
```

6.2.3.32 smAppearanceMsgThresh

Display the threshold for Appearance & Disappearance messages.

6.2.3.32.1 Syntax

smAppearanceMsgThresh

6.2.3.32.2 Options

None.

6.2.3.32.3 Notes

This command displays the threshold for the number of fabric appearance and disappearance log messages that may be logged as NOTICEs per sweep by the SM. A value of zero causes all such messages to be logged at the NOTICE level. A value greater than zero will cause the priority of any subsequent messages to be logged at the INFO priority. Subnet manager must be running to display this information.

6.2.3.32.4 Examples

```
-> smAppearanceMsgThresh

Message Threshold is 0 (disabled).
```

6.2.3.33 smMcastCheck

Enables or disables the Common MTU and rate for multicast groups.

6.2.3.33.1 Syntax

smMcastCheck

6.2.3.33.2 Options

None

6.2.3.33.3 Notes

MTU and rate checking for multicast groups verifies that multicast group creation and join requests have MTU's and rates supported by the entire fabric. If disabled, the check allows end nodes to create and join any multicast groups that their port MTU and rate may support.

6.2.3.33.4 Examples

```
-> smMcastCheck

Common MTU and rate for multicast groups is enabled
```

6.2.4 FM Loop Test

The following commands set up and perform loop test in the fabric.The Loop Test should be performed after an installation or major changes have occurred in the fabric or cluster to help validate that data is being transported as intended.

- smLooptestStart – Starts the SM Loop Test in normal mode
- smLooptestFastModeStart – Starts the SM Loop Test in fast mode
- smLooptestStop – Stops the SM Loop Test
- smLooptestInjectPackets – Injects packets into the SM Loop Test
- smLooptestInjectAtNode – Injects packets to a specific switch node for the SM Loop Test
- smLooptestInjectEachSweep – Injects packets on each sweep for the SM Loop Test
- smLooptestPathLength – Sets the loop path length for the SM Loop Test
- smLooptestMinISLRedundancy – Sets the minimum ISL redundancy in fast mode for the SM Loop Test
- smLooptestShowLoopPaths – Displays the loop paths for the SM Loop Test
- smLooptestShowSwitchLft – Displays a switch's LFT for the SM Loop Test
- smLooptestShowTopology – Displays the topology for the SM Loop Test
- smLooptestShowConfig – Displays the con figuration for the SM Loop Test

6.2.4.1 smLooptestStart

Starts the SM Loop Test in normal mode. Either accepts a single argument of "number of packets" or no arguments which will default to 0 packets. Command will start SM if it is not running.

6.2.4.1.1 Syntax

smLooptestStart [*packets*]

6.2.4.1.2 Options

packets – The number of 256 byte packets used when starting the SM Loop Test. Valid values are 0-10.

6.2.4.1.3 Notes

Use this command to start the SM Loop Test with the specified number of 256 byte packets. Valid values for number of packets are 0-10 (default=0). If the number of packets is 0, then no packets will be injected. If SM has not been previously started this command will start SM. Note that Loop Test will only function if the SM is in the Master state.

6.2.4.1.4 Examples

```
> smLooptestStart 5

Waiting for SM to complete startup.....................done

The SM Loop Test is being started
```

6.2.4.2 smLooptestFastModeStart

Starts the SM Loop Test in fast mode. Either accepts a single argument of "number of packets" or no arguments which will default to 4 packets. Command will start SM if it is not running.

6.2.4.2.1 Syntax

smLooptestFastModeStart [*packets*]

6.2.4.2.2 Options

packets – The number of 256 byte packets used when starting the SM Loop Test. Valid values are 0-10.

6.2.4.2.3 Notes

Use this command to start the SM Loop Test in Fast Mode with the specified number of 256 byte packets. Valid values for number of packets are 0-10 (default=4). If the number of packets is 0, then no packets will be injected. If SM has not been previously started this command will start SM. Note that Loop Test will only function if the SM is in the Master state.

6.2.4.2.4 Examples

```
> smLooptestFastModeStart 5

Waiting for SM to complete startup.....................done

The SM Loop Test is being started
```

6.2.4.3 smLooptestStop

Stops the SM Loop Test. Command will stop SM if it was not started using the `smcontrol start` command but was instead started by either the `smLooptestStart` command or the `smLooptestFastModeStart` command.

6.2.4.3.1 Syntax

smLooptestStop

6.2.4.3.2 Options

N/A

6.2.4.3.3 Notes

Use this command to stop the SM Loop Test. Returns switch LFTs back to normal.

6.2.4.3.4 Examples

```
> smLooptestStop

0:ESM: SM: TT: SM Forced Sweep scheduled: ESM Loop Test Stop

The SM Loop Test is being stopped
```

6.2.4.4 smLooptestInjectPackets

Injects packets into the SM Loop Test

6.2.4.4.1 Syntax

smLooptestInjectPackets[*packets*]

6.2.4.4.2 Options

packets – The number of packets to inject into the SM Loop Test. Valid values are 1-10.

6.2.4.4.3 Notes

Use this command to inject packets into the SM Loop Test. Valid values for number of packets are 1-10 (default=1).

6.2.4.4.4 Examples

```
> smLooptestInjectPackets

Sending 1 packets to all loops

0:ESM: SM: TT: SM Forced Sweep scheduled: ESM Loop Test Inject Packets

Packets have been injected into the SM Loop Test
```

6.2.4.5 smLooptestInjectAtNode

Injects packets to a specific switch node for the SM Loop Test

6.2.4.5.1 Syntax

smLooptestInjectAtNode [*node index*]

6.2.4.5.2 Options

node index – The node index of the switch in which to inject packets.

6.2.4.5.3 Notes

Use this command to inject packets into the SM Loop Test at a particular switch node.

6.2.4.5.4 Examples

```
> smLooptestInjectAtNode 0

Sending 5 packets to node index 0

0:ESM: SM: TT: SM Forced Sweep scheduled: ESM Loop Test Inject Packets at Node

Packets have been injected into the SM Loop Test for node 0
```

6.2.4.6 smLooptestInjectEachSweep

Injects packets on each sweep for the SM Loop Test

6.2.4.6.1 Syntax

smLooptestInjectEachSweep[*inject/not inject*]

6.2.4.6.2 Options

inject/not inject – Inject (1) or do not Inject (0) packets on each sweep for the SM Loop Test. Valid values are 1 or 0.

6.2.4.6.3 Notes

Use this command to inject (1) or not inject (0) packets on each sweep for the SM Loop Test.

6.2.4.6.4 Examples

```
> smLooptestInjectEachSweep 1

sm_looptest_inject_packets_each_sweep: loop test will inject packets every sweep,
numPackets=5

The SM Loop Test will inject packets every sweep
```

6.2.4.7 smLooptestPathLength

Sets the loop path length for the SM Loop Test

6.2.4.7.1 Syntax

smLooptestPathLength[*length*]

6.2.4.7.2 Options

length – The loop path length for the SM Loop Test. Valid values are 2-4.

6.2.4.7.3 Notes

Use this command to set the loop path length for the SM Loop Test. Valid values for loop path length are 2-4 (default=3).

6.2.4.7.4 Examples

```
> smLooptestPathLength 4

0:ESM: SM Info: setLoopPathLength: Loop path length =  4

0:ESM: SM: TT: SM Forced Sweep scheduled: ESM Loop Test Path Length Changed
```

```
The SM Loop Test path length has been set to 4
```

6.2.4.8 smLooptestMinISLRedundancy

Sets the minimum number of loops in which to include each ISL in fast mode.

6.2.4.8.1 Syntax

smLooptestMinISLRedundancy[*loops*]

6.2.4.8.2 Options

loops – The minimum number of loops in which to include each ISL for the SM Loop Test.

6.2.4.8.3 Notes

Use this command to set the minimum number of loops (default=4) in which to include each ISL for the SM Loop Test in Fast Mode. Note this command is only applicable if running Loop Test in Fast Mode.

6.2.4.8.4 Examples

```
> smLooptestMinISLRedundancy 3

0:ESM: SM: TT: SM Forced Sweep scheduled:
ESM Loop Test Minimum ISL Redundancy Changed

0:ESM: SM Info: setLoopMinISLRedundancy:
Loop MinISLRedundancy =  3
```

6.2.4.9 smLooptestShowLoopPaths

Shows the loop paths through a node, or all loop paths if no "node index" is specified.

6.2.4.9.1 Syntax

smLooptestShowLoopPaths [*node index*]

6.2.4.9.2 Options

node index – The node index of the node in which to print the loop paths.

6.2.4.9.3 Notes

Use this command to print the loop paths through a node specified by node index, or all nodes (default) for the SM Loop Test.

6.2.4.9.4 Examples

```
> smLooptestShowLoopPaths


Node Idx: 0, Guid: 0x00066a00d8000118 Desc i9k118

------------------------------------------------------------

 Node  Node                      Node     Path

 Idx   Lid        NODE GUID      #Ports    LID    PATH[n:p->n:p]
```

```
 ---- ------  ---------------  ------  ------  ---------------

------------------------------------------------------------

There are 0 total loop paths of <=3 links in length in the fabric!

------------------------------------------------------------
```

6.2.4.10 smLooptestShowSwitchLft

This command shows the Lid Forwarding Table for all switches or a specific switch in the fabric.

The "smLooptestShowSwitchLft" command appears to be a unique display of a switches (or just an individual switch) LFT. Therefore it appears to provide important LFT information with the Loop Test grouping in the CLI help menu. It can be removed easily if it is deemed necessary since there are other commands to get this information

6.2.4.10.1 Syntax

smLooptestShowSwitchLft [*node index*]

6.2.4.10.2 Options

node index – The node index of the switch in which to print switch LFT.

6.2.4.10.3 Notes

Use this command to print switch LFT specified by switch index, or all switches (default) for the SM Loop Test test.

6.2.4.10.4 Examples

```
-> smLooptestShowSwitchLft

Node[0000]  LID=0x0001  GUID=0x00066a00d8000118 [i9k118] Linear Forwarding Table

   LID    PORT

 ------   ----

 0x0001   0000

 0x0002   0009

 0x0003   0016

 --------------
```

6.2.4.11 smLooptestShowTopology

This command shows the topology of the fabric relevant to the SM Loop Test.

The "smLooptestShowTopology" command it a replication of the "smShowLids" command but it is still include in the Loop Test list since it provide vital topology info within the Loop Test grouping in the CLI help menu. It can be removed easily if it is deemed necessary.

6.2.4.11.1 Syntax

smLooptestShowTopology

6.2.4.11.2 Options

None

6.2.4.11.3 Notes

Use this command to print the topology for the subnet manager loop test.

6.2.4.11.4 Examples

```
> smLooptestShowTopology

sm_state = MASTER   count = 673   LMC = 0, Topology Pass count = 5, Priority = 0,
Mkey = 0x0000000000000000


------------------------------------------------------------

i9k118

------------------------------------------------------------

Node[  0] => 00066a00d8000118  (2)  ports=24, path=

Port ---- GUID ---- (S)   LID      LMC   _VL_ __MTU__ __WIDTH__ ___SPEED____
CAP_MASK  N#  P#

0   00066a00d8000118  4 LID=0001 LMC=0000 8 8 2k  2k      1X/4X 4X    2.5 2.5
   00000a4a 0   0

9  0000000000000000   4                   8 8 2k  2k      1X/4X 4X    2.5 2.5
00000000 1   1

16 0000000000000000   4                   8 1 2k  2k      1X/4X 4X    2.5 2.5
00000000 2   1


------------------------------------------------------------

burns HCA-1

------------------------------------------------------------

Node[  1] => 00066a0098006cad  (1)  ports=2, path= 9

Port ---- GUID ---- (S)   LID      LMC   _VL_ __MTU__ __WIDTH__ ___SPEED____
CAP_MASK  N#  P#

1   0066a00a0006cad  4  LID=0002  LMC=0000 8 8 2k 2k      1X/4X 4X   2.5/5 2.5
02510a68 0 9 9


------------------------------------------------------------

shaggy HCA-1

------------------------------------------------------------

Node[  2] => 0011750000ff8f4d  (1)  ports=1, path=16

Port ---- GUID ---- (S)   LID      LMC   _VL_ __MTU__ __WIDTH__ ___SPEED____
CAP_MASK  N#  P#
```

```
1 0011750000ff8f4d    4  LID=0003  LMC=0000 1 1 4k 2k      1X/4X  4X   2.5/5 2.5
0761086a 0  16 16
```

6.2.4.12 smLooptestShowConfig

Displays the SM Loop Test Configuration

6.2.4.12.1 Syntax

smLooptestShowConfig

6.2.4.12.2 Options

None

6.2.4.12.3 Notes

Use this command to print the loop test con figuration for the subnet manager loop test.

6.2.4.12.4 Examples

```
> smLooptestShowConfig

Loop Test is running with following parameters:

 Max Path Length    #Packets    Inject Point

 ---------------    --------    ------------

        3            00005         Node 0


FastMode=0, FastMode MinISLRedundancy=4, InjectEachSweep=1, TotalPktsInjected
since start=0
```

§ §

7.0 Installation and Set Up

7.1 Installing the Host FM on Linux

Note: The Host FM software requires that the same version of the Intel® HCA host stack be installed with at least the adapter drivers. These drivers must be set to **startup**.

The installation provides an interactive INSTALL and is packaged as a tgz. Intel® recommends to use the `./INSTALL`.The rpms are still within the tgz file and can be installed using standard rpm commands if required. The INSTALL command installs the FM. When using the INSTALL command located in the `IntelIB-FM.*` or `IntelIB-IFS.*` directories, the installation process interactively prompts the user to keep or upgrade the FM configuration file. The new configuration file is always placed in a `-sample` file for later reference. This procedure is different from the previous rpm install method, where the user had to manually copy the new `.rpm` file.

Note: For IntelIB-IFS installations, refer to the *Intel® True Scale Fabric Software Installation Guide* for installation procedures.

Note: After installing the FM, the user must reboot the server or use the following startup procedures for the new installation to take effect.

7.2 Controlling the FM

The following CLI command controls the FM, with commands to start, stop, restart, and other functions

7.2.1 ifs_fm /syntax

/etc/init.d/ifs_fm [start|stop|restart|reload|sweep|status] [-i *instance*] [-f] [*component*|*compname*|*insname*]...]

7.2.2 ifs_fm Options

`start` – Starts the selected instances/managers.

`stop` – Stops the selected instances/managers.

`restart` – Stops, then restarts the selected instances/managers.

`reload` – Reloads the configuration for the selected instances and managers (this option only handles changes to the <Start> parameters).

`sweep` – Forces a fabric resweep for the selected instances/managers.

`status` – Shows status (running, not running, or disable) for the selected instances/managers.

By default, `start`, `stop`, `restart`, `reload`, `sweep`, and `status simultaneously` controls all instances of all FM components and managers.

`-i` *`instance`* – This option can be specified multiple times and indicates that only specific instances (as configured in the `ifs_fm.xml` configuration file) are to be started or stopped. This value should be an integer value greater than or equal to 0. Without this option, all instances are acted on.

`-f` – When used with `start`, this option forces a manager to start even if it already appears to be running.

component|*compname*|*insname* – This option can be specified more than once and can be any of the following:

- *Component*:
  - `sm` – Subnet manager
  - `fe` – Fabric executive
- *Compname*:
  A specific component/manager name such as `fm0_sm` or `fm1_fe` (manager names are formed by combining an instance name and one of the manager names listed previously, separated by an underscore).
- *Insname*:
  A specific instance name such as `fm0` or `fm1` (instance names are defined in the `Fm.Name` parameter in the configuration file).

When `-i` is used, only the specified components in the instance are started. The `-i` option can be specified more than once to select multiple instances.

When a *`compname`* is specified, that component is started (regardless of `-i`).

When an *`insname`* is specified, all components of that instance are started.

If components are specified, then all components in selected instances are started.

If no arguments are specified, all instances are acted on.

7.2.3 ifs_fm Examples

To start the SM for instance 0:

```
/etc/init.d/ifs_fm start -i 0 sm
```

To start the SM for instances 0 and 1:

```
/etc/init.d/ifs_fm start -i fm0 -i fm1 sm
```

To start the SM for all instances:

```
/etc/init.d/ifs_fm start sm
```

To start all managers for instances 0 and 1:

```
/etc/init.d/ifs_fm start -i 0 -i 1
```

To start the SM only for instance 1:

```
/etc/init.d/ifs_fm start fm1_sm
```

Note: `Start`, `restart`, and `sweep` only act on instances and managers enabled in the configuration file using their corresponding `Start` parameters.

7.3 Starting the Fabric Manager

1. Start the Fabric Manager:
   `/etc/init.d/ifs_fm start`
2. Verify that all the tasks are up and running:
   `/etc/init.d/ifs_fm status`

Note: The default configuration runs FM on port 1 of the first HCA in the system.

To run the Fabric Manager on either port 2, or ports 1 and 2, the user must edit the `/etc/sysconfig/ifs_fm.xml` configuration file.

7.4 Removing the Fabric Manager

Refer to the *Intel® True Scale Fabric Software Installation Guide* for removal procedures for the Fabric Manager.

7.5 Stopping the Fabric Manager

To stop the Fabric Manager, follow these steps:

1. Log in to the Fabric Manager system as **`root`** or as a user with **`root`** privileges.
2. Stop the Fabric Manager.

```
/etc/init.d/ifs_fm stop
```

The `ifs_fm` script shows this message:

```
st99:/etc/sysconfig # /etc/init.d/ifs_fm stop

Stopping Intel Fabric Manager

Stopping FE 0: fm0_fe: [  OK  ]

Stopping BM 0: fm0_bm: [  OK  ]

Stopping PM 0: fm0_pm: [  OK  ]

Stopping SM 0: fm0_sm: [  OK  ]
```

7.6 Automatic Startup

Refer to the *Intel® True Scale Fabric Suite FastFabric User Guide* for information on controlling the automatic startup of FM.

§ §

Appendix A Fabric Manager Command Line Interface

The following pages describe the host FM Command Line Interface (CLI) commands and gives examples of each. The directory path for each CLI command is shown beside the Path heading.

A.1 config_check

Verifies the syntax of a FM XML configuration file.

A.1.1 Syntax

config_check [-c *config_file*] [-d] [-v] [-s]

A.1.2 Path

`/opt/ifs_fm/etc/`

A.1.3 Options

`-c` *`config_file`* – Configuration file to check. Default is `/etc/sysconf/ifs_fm.xml`.

`-d` – Display configuration consistency checksum information for all FM instances. The checksum calculation method configured within the XML configuration being tested will be used.

`-v` – Enable verbose output to display debugging and status information.

`-s` – Strict check mode. This provides warnings for inconsistencies in the configuration file such as missing tags or references to undefined names. Such warnings will not be fatal during normal FM operation, but may represent mistakes in the supplied configuration file.

A.1.4 Notes

The exit code of this command will indicate the success (0) or failure (non-zero) of syntax check. On success the command is silent by default. On failure the configuration syntax errors and warnings will be shown.

Unless `-s` is specified, checks are limited to syntax.

Warnings will not affect the exit code.

When using an embedded FM, this command is also available on hosts with Fast Fabric installed in the `/opt/iba/fm_tools` directory.

A.1.5 Examples

```
config_check -c my_fm_config.xml
```

```
config_check -d -c my_fm_config.xml
```

```
FM instance 0
```

```
   SM overall checksum   427870    consistency checksum   417845

   PM overall checksum     3722    consistency checksum     1092

   FE overall checksum     5875    consistency checksum     3255

   BM overall checksum     2642    consistency checksum     2052

   VF database consistency checksum 3037065446

       VF Networking consistency checksum 1796288057

       VF Default consistency checksum    67778

...
```

A.2 config_convert

Converts an iview_fm.config file from an older release into a comparable FM XML configuration file using `ifs_fm_src.xml`.

A.2.1 Syntax

config_convert [-d] [-D] *old_file* /opt/ifs_fm/etc/ifs_fm_src.xml

A.2.2 Path

`/opt/ifs_fm/etc/`

A.2.3 Options

`-d` – Show tags not found in `old_file`.

`-D` – Extra debug output to `stderr`.

`old_file` – An `iview_fm.config` file to convert.

`/opt/ifs_fm/etc/ifs_fm_src.xml` – File which describes mapping of old parameters to XML parameters. This exact filename should be specified.

A.2.4 Notes

The converted file is output to stdout.

When using an embedded FM, this command is also available on hosts with Fast Fabric installed in the `/opt/iba/fm_tools` directory. In which case the `/opt/iba/fm_tools/ifs_fm_src.xml` file can be used as the mapping file.

A.2.5 Examples

```
config_convert /opt/ifs_fm/etc/iview_fm.config.VERSION
/opt/ifs_fm/etc/ifs_fm_src.xml > my_fm_config.xml
```

A.3 config_diff

Compares two FM configuration files ignoring differences in white space and comments.

A.3.1 Syntax

config_diff [-f][-l] [-d *diff_args*] file1 file2

A.3.2 Path

`/opt/ifs_fm/etc/`

A.3.3 Options

`-f` – Filter out FM parameters which are not part of consistency check.

`-l` – Include comments in XML to indicate original line numbers (this is only recommended when the files are very similar, including whitespace).

`-d` *`diff_args`* – Additional arguments to diff command `-c`.

file1 file2 – The two FM XML configuration files to compare.

A.3.4 Notes

The exit code of this command will indicate the success (0) or failure (non-zero) of the diff. On success the command is silent by default. On failure the differences will be shown.

Whitespace and comment differences are always filtered out.

When using an embedded FM, this command is also available on hosts with Fast Fabric installed in the `/opt/iba/fm_tools` directory.

A.3.5 Examples

```
config_diff /ete/sysconf/ifs_fm.xml my_fm_config.xml
```

A.4 config_generate

Interactively generates a FM XML configuration file.

A.4.1 Syntax

config_generate [-e] dest_file

A.4.2 Path

`/opt/ifs_fm/etc/`

A.4.3 Options

`-e` – Generate file for embedded FM (For example, don't prompt for features not applicable to embedded such as multiple FM instances). Default is to generate a file for host FM.

`dest_file` – Name of file to generate.

A.4.4 Notes

This command presently allows interactive selection of the following FM configuration parameters:

- SubnetSize
- LMC (multi-LID control)
- AdaptiveRouting (Enable, LostRouteOnly, Tier1FatTree)
- LogMode
- NodeAppearanceMsgThreshold
- Which FM instances should be enabled
- Name for each FM Instance
- IPoIB MulticastGroup Rate and MTU for each FM instance
- FM primary or secondary status for failover for each FM instance
- sticky failover
- SubnetPrefix for each FM instance
- Performance Manager
  - Sweep Interval
  - Logging Thresholds
  - Number of PM/PA clients
  - Number of historical images to retain

When using an embedded FM, this command is also available on hosts with FastFabric installed in the `/opt/iba/fm_tools` directory.

A.4.5 Examples

```
config_generate my_fm_config.xml
```

A.5 fm_capture

Provides a capture of FM configuration and present status to aid Intel® support in troubleshooting problems.

A.5.1 Syntax

fm_capture [*fm_instance* ...]

A.5.2 Path

```
/opt/ifs_fm/bin/
```

A.5.3 Options

`fm_instance` – One or more FM instance numbers, by default all running instances will be captured.

A.5.4 Notes

A dated tgz file will be created in the current directory.

This command is automatically included in the information gathered by `iba_capture`. This command should only be used if directed by Intel® Support.

A.5.5 Examples

```
fm_capture 0
```

A.6 fm_cmd

A utility that provides access to some diagnostic, control, and status capabilities of the SM.

A.6.1 Syntax

fm_cmd [-i *value*] [smForceSweep] [smRestorePriority] [smShowCounters]
[smResetCounters] [smStateDump] [smLogLevel *loglevel*]
[smLogMode *logmode*] [smLogMask *subsystem mask*]
[smPerfDebug] [saPerfDebug] [saRmppDebug]
[bmForceSweep] [bmLogLevel *loglevel*] [bmLogMode *logmode*]
[bmLogMask *subsystem mask*] [bmDebug] [bmRmppDebug]
[pmShowCounters] [pmResetCounters] [pmDebug]
[pmRmppDebug] [feLogLevel *loglevel*] [feLogMode *logmode*]
[feLogMask *subsystem mask*] [feDebug] [feRmppDebug]
[smLopptestStart] [smLooptestFastModeStart]
[smLooptestStop] [smLooptestInjectPackets *numPkts*]
[smLooptestInjectAtNode *SwNodeIndex*]
[smLooptestPathLength *pathlength*]
[smLooptestMinISLRedundancy *number of loops*]
[smLooptestShowLoopPaths *node index* | *all*]
[smLooptestShowSwitchLft *node index* | *all*]
[smLooptestShowTopology] [smLooptestShowConfig]
[smForceRebalance] [smBroadcastConfig]

A.6.2 Path

```
/opt/ifs_fm/bin/
```

A.6.3 Options

`-i` *`value`* – Instance to connect to (default is 0).

`smForceSweep` – Make the SM sweep now.

`smRestorePriority` – Restore the normal priority of the SM (if it is currently elevated).

`smShowCounters` – Get statistics and performance counters from the SM.

`smResetCounters` – Reset SM statistics and performance counters.

`smStateDump` – Dump Internal SM state into directory specified.

`smLogLevel` *`loglevel`* – Set the SM logging level (1=WARN+, 2=INFINI_INFO+, 3=INFO+, 4=VERBOSE+, 5=DEBUG2+, 6=DEBUG3+, 7=TRACE+).

`smLogMode` *`logmode`* – Set the SM log mode flags (0/1 1=downgrade non-actionable, 0/2 2=logfile only).

`smLogMask` *`subsystem mask`* – Set the SM log mask for a specific subsystem to the value given; see Section 4.1.2, "Shared Parameters" on page 67 for a list of subsystems and mask bit meanings.

`smPerfDebug` – Toggle performance debug output for SM.

`saPerfDebug` – Toggle performance debug output for SA.

`saRmppDebug` – Toggle Rmpp debug output for SA.

`bmForceSweep` – Make the BM sweep now.

`bmRestorePriority` – No longer supported, use `smRestorePriority`.

`bmLogLevel` *`loglevel`* – Set the BM logging level (1=WARN+, 2=INFINI_INFO+, 3=INFO+, 4=VERBOSE+, 5=DEBUG2+, 6=DEBUG3+, 7=TRACE+).

`bmLogMode` *`logmode`* – Set the BM log mode flags (0/1 1=downgrade non-actionable, 0/2 2=logfile only).

`bmLogMask` *`subsystem mask`* – Set the BM log mask for a specific subsystem to the value given; see Section 4.1.2, "Shared Parameters" on page 67 for a list of subsystems and mask bit meanings.

`bmDebug` – Toggle debug output for BM.

`bmRmppDebug` – Toggle Rmpp debug output for BM.

`pmRestorePriority` – No longer supported, use `smRestorePriority`.

`pmLogLevel` – No longer supported, use `smLogLevel` *`loglevel`*.

`pmLogMode` – No longer supported, use `smLogMode` *`logmode`*.

`pmLogMask` – No longer supported, use `smLogMask` *`subsystem mask`*.

`pmShowCounters` – Get statistics and performance counters about the PM.

`pmResetCounters` – Reset statistics and performance counters about the PM.

`pmDebug` – Toggle debug output for PM.

`pmRmppDebug` – Toggle Rmpp debug output for PM.

`feLogLevel` *`loglevel`* – Set the FE logging level (1=WARN+, 2=INFINI_INFO+, 3=INFO+, 4=VERBOSE+, 5=DEBUG2+, 6=DEBUG3+, 7=TRACE+).

`feLogMode` *`logmode`* – Set the FE log mode flags (0/1 1=downgrade non-actionable, 0/2 2=logfile only).

feLogMask *subsystem mask* – Set the FE log mask for a specific subsystem to the value given see Section 4.1.2, "Shared Parameters" on page 67 for a list of subsystems and mask bit meanings.

`feDebug` – Toggle debug output for FE.

`feRmppDebug` – Toggle Rmpp debug output for FE.

`smLooptestStart` – START loop test in normal mode - specify the number of 256 byte packets (default=0).

`smLooptestFastModeStart` – START loop test in fast mode - specify the number of 256 byte packets (default=4).

`smLooptestStop` – STOP the loop test (puts switch LFTs back to normal).

`smLooptestInjectPackets` *`numPkts`* – Enter numPkts to send to all switch loops (default=1).

`smLooptestInjectAtNode` *`SwNodeIndex`* – Enter the switch node index to inject loop packets (default=0).

`smLooptestInjectEachSweep` – 1 to inject packets each sweep, 0 to stop injecting each sweep.

`smLooptestPathLength` *`pathlength`* – Sets the loop path length 2-4 (default=3.)

`smLooptestMinISLRedundancy` *`number of loops`* – Sets the minimum number of loops in which to include each ISL (default=4).

`smLooptestShowLoopPaths` *`node index`* | *`all`* – Displays the loop paths given node index or all loop paths (default=all).

`smLooptestShowSwitchLft` *`node index`* | *`all`* – Displays a switch LFT given node index or all switches LFTs (default=all).

`smLooptestShowTopology` – Displays the topology for the SM Loop Test.

`smLooptestShowConfig` – Displays the current active loop configuration.

`smForceRebalance` – Toggle Force Rebalance setting for SM.

`smBroadcastConfig` – Broadcast the XML configuration file to STANDBY and INACTIVE SM's.

A.6.4 Examples

```
-> fm_cmd -i 2 smForceSweep
```

A.7 getlids

Displays all nodes within a fabric.

Note: This command is not recommended for use.

A.7.1 Syntax

`getlids`

A.7.2 Path

`/opt/ifs_fm/old/`

A.7.3 Options

None

A.7.4 Notes

This command is not recommended for use. Use FastFabric commands `iba_saquery` or `iba_report` to display all nodes within a fabric.

A.7.5 Examples

```
getlids
```

A.8 sm_capture

This command has been renamed, `fm_capture`. Refer to Section A.5, "fm_capture" on page 196.

A.9 sm_diag

This command has been renamed, `fm_cmd`. Refer to Section A.6, "fm_cmd" on page 197.

A.10 smpoolsize

A utility that determines the SM memory requirements of a particular fabric size.

A.10.1 Syntax

smpoolsize [-n *number*] [-s *size*] [-p *ports*] [-l *LID*]

A.10.2 Path

```
/opt/ifs_fm/bin/
```

A.10.3 Options

`-n` *`number`* – Number of HCAs.

`-s` *`size`* – Number of switch chips.

`-p` *`ports`* – Number of ports per switch.

`-l` *`LID`* – Local Identifier (LID) Mask Control.

A.10.4 Notes

The size computed is a rough estimate and does not account for the other managers (Pm, Bm, Fe). The size of a fabric with 244 HCAs and thirty 24-port switches is 10,676,432 bytes.

A.10.5 Examples

A 244-node fabric with thirty 24-port switch chips and an LMC of 0 would be input as follows:

```
-> smpoolsize -n 244 -s 30 -p 24 -l 0
```

§ §

Appendix B FM Log Messages

B.1 FM Event Messages

The host-based and embedded FM both log significant fabric events in a standard machine-readable format. The format for these special event messages provides information not only about the event, but information regarding what nodes in the fabric are causing the event.

B.1.1 FM Event Message Format

The format of these messages is as follows:

```
<prefix>;MSG:<msgType>|SM:<sm_node_desc>:port <sm_port_number>|
COND:<condition>|NODE:<node_desc>:port <port_number>:<node_guid>|
LINKEDTO:<linked_desc>:port <linked_port>:<linked_guid>| DETAIL:<details>
```

Where:

- <prefix> – Includes the date and time information of the event along with either the slot number OR hostname and IP address of the FM reporting the message.
- <msgType> – Is one of the following values:
  - ERROR
  - WARNING
  - NOTICE
  - INFORMATION
- <sm_node_desc> and <sm_port_number> – Indicate the node name and port number of the SM that is reporting the message, prefixed with the word 'port'. For the embedded version of the SM, the port number will be 0. Any pipes (|) or colons (:) in the node description will be converted to spaces in the log message.
- <condition> – Is one of the thirteen conditions from the event SM Reporting Table that are detailed in the section Section B.1.2, "FM Event Descriptions" on page 202. The condition text includes a unique identification number. The possible conditions are as follows:
  1. Redundancy Lost
  2. Redundancy Restored
  3. Appearance in Fabric
  4. Disappearance from Fabric
  5. SM State Change to Master
  6. SM State Change to Standby
  7. SM Shutdown
  8. Fabric Initialization Error
  9. Link Integrity Error
  10. Security Error
  11. Other Exception
  12. Fabric Summary
  13. SM State Change to Inactive
- <node_desc>, <port_number>, and <node_guid> are the node description, port number and node GUID of the port and node that are primarily responsible for

the event. Any pipes (|) or colons (:) in the node description will be converted to spaces in the log message.

- <linked_desc>, <linked_port> and <linked_guid> are optional fields describing the other end of the link. These fields and the 'LINKEDTO' keyword will only be shown in applicable messages. Any pipes (|) or colons (:) in the node description will be converted to spaces in the log message.
- <details> is an optional free-form field detailing additional information useful in diagnosing the log message cause.

An example of such an event message would be (line wrapped to show separate fields):

```
Oct 10 13:14:37 slot 101:172.21.1.9; MSG:ERROR|
SM:SilverStorm 9040 GUID=0x00066a00db000007 Spine 101, Chip A:port
0|
COND:#9 Link Integrity Error|
NODE:SilverStorm 9040 GUID=0x00066a00db000007 Spine 101, Chip
A:port 10:0x00066a00db000007|
LINKEDTO:9024 DDR GUID=0x00066a00d90001db:port
15:0x00066a00d90001db|
DETAIL:Excessive Buffer Overrun threshold trap received.
```

B.1.2 FM Event Descriptions

The following are the FM event messages, their severity, an explanation, possible causes for the event.

B.1.2.1 #1 Redundancy Lost

B.1.2.1.1 Severity

Warning

B.1.2.1.2 Explanation

The subnet manager emits this message when it is the only running Subnet Manager on a given subnet.

B.1.2.1.3 Causes

No redundant SM exists on the subnet

A user shutdown a redundant SM or possibly disconnected or shutdown the node on which the SM was running.

B.1.2.1.4 Action

If running redundant SM's on a fabric, verify health of each host or switch running an SM.

B.1.2.1.5 Example

```
Apr  4 18:29:19 nibbler iview_sm[6145]: nibbler.dev.silverstorm.com;
MSG:WARNING|SM:nibbler.dev.silverstorm.com:port 1|COND:#1 Redundancy
lost|NODE:nibbler.dev.silverstorm.com:port 1:0x00066a00a0006f73|DETAIL:SM
redundancy not available
```

B.1.2.2 #2 Redundancy Restored

B.1.2.2.1 Severity

Notice

B.1.2.2.2 Explanation

The Master SM for the subnet detected that another SM has come online

B.1.2.2.3 Causes

A user started a redundant SM on another host or switch.

A user just connected to separate subnets together

B.1.2.2.4 Action

None

B.1.2.2.5 Example

```
Apr  8 20:25:27 nibbler iview_sm[6145]: nibbler.dev.silverstorm.com;
MSG:NOTICE|SM:nibbler.dev.silverstorm.com:port 1|COND:#2 Redundancy
restored|NODE:nibbler.dev.silverstorm.com:port 1:0x00066a00a0006f73|DETAIL:2 SM's
now online in fabric
```

B.1.2.3 #3 Appearance in Fabric

B.1.2.3.1 Severity

Notice

B.1.2.3.2 Explanation

A new HCA port, switch, inter-switch link or Subnet Manager was detected by the master Subnet Manager.

B.1.2.3.3 Causes

User action

B.1.2.3.4 Action

None

B.1.2.3.5 Example

```
Apr  8 20:16:09 nibbler iview_sm[6145]: nibbler.dev.silverstorm.com;
MSG:NOTICE|SM:nibbler.dev.silverstorm.com:port 1|COND:#3 Appearance in
fabric|NODE:hubert:port 1:0x00066a00a0006caa|DETAIL:Node type: hca
```

B.1.2.4 #4 Disappearance from Fabric

B.1.2.4.1 Severity

Notice

B.1.2.4.2 Explanation

An HCA port, switch, inter-switch link or Subnet Manager has disappeared from fabric. This encompasses system shutdowns, and loss of connectivity.

B.1.2.4.3 Action

The administrator should validate whether or not the components have disappeared from the fabric due to user action or not. Nodes will typically disappear from the fabric when they are rebooted, re-cabled, or if their True Scale Fabric stacks are stopped.

B.1.2.4.4 Example

```
Apr  8 20:25:54 nibbler iview_sm[6145]: nibbler.dev.silverstorm.com;
MSG:NOTICE|SM:nibbler.dev.silverstorm.com:port 1|COND:#4 Disappearance from
fabric|NODE:SilverStorm 9024 GUID=0x00066a00d8000123:port
11:0x00066a00d8000123|LINKEDTO:SilverStorm 9024 DDR GUID=0x00066a00d90002a6:port
23:0x00066a00d90002a6|DETAIL:inter-switch link disappeared
```

B.1.2.5 #5 SM State Change to Master

B.1.2.5.1 Severity

Notice

B.1.2.5.2 Explanation

Subnet manager transitioned into the master state from one of the 'standby', 'discovering' or 'not active' states.

B.1.2.5.3 Action

The administrator should check the state of the machine (or chassis) that was providing the master SM service to determine if it has failed and needs to be replaced, or whether the state change occurred due to user action.

B.1.2.5.4 Example

```
Nov 28 17:45:25 endrin iview_sm[29326]:
;MSG:NOTICE|SM:endrin.dev.infiniconsys.com:port 1|COND:#5 SM state to
master|NODE:endrin.dev.infiniconsys.com:port
1:0x0x00066a00a0000405|DETAIL:transition from DISCOVERING to MASTER
```

B.1.2.6 #6 SM State Change to Standby

B.1.2.6.1 Severity

Notice

B.1.2.6.2 Explanation

Subnet manager transitioned from 'master' into 'standby' state.

B.1.2.6.3 Action

The administrator should validate that this was due to a modification in the True Scale Fabric network configuration. If not, then this issue should be reported to customer support.

B.1.2.6.4 Example

```
Nov 29 12:15:28 endrin iview_sm[31247]:
;MSG:NOTICE|SM:endrin.dev.infiniconsys.com:port 1|COND:#6 SM state to
standby|NODE:endrin.dev.infiniconsys.com:port
1:0x0x00066a00a0000405|DETAIL:transition from MASTER to STANDBY
```

B.1.2.7 #7 SM Shutdown

B.1.2.7.1 Severity

Notice

B.1.2.7.2 Explanation

The master subnet manager is shutting down.

B.1.2.7.3 Action

The administrator should check the state of the machine (or chassis) that was providing the master SM service, or whether the state change occurred due to user action.

B.1.2.7.4 Example

```
Nov 28 17:42:22 endrin iview_sm[29030]:
;MSG:NOTICE|SM:endrin.dev.infiniconsys.com:port 1|COND:#7 SM
shutdown|NODE:endrin.dev.infiniconsys.com:port 1:0x0x00066a00a0000405|DETAIL:
```

B.1.2.8 #8 Fabric Initialization Error

B.1.2.8.1 Severity

Notice

B.1.2.8.2 Explanation

Some form of error occurred during fabric initialization. Examples of possible errors include:

- Link could not be activated in 4x mode.
- Subnet manager could not initialize a port or node with proper configuration.

B.1.2.8.3 Action

The administrator should perform the fabric troubleshooting procedure to isolate and repair the faulty component. The faulty component could be the SM platform itself (For example, its own HCA) or a component in the True Scale Fabric network.

B.1.2.8.4 Example

```
Apr  6 22:48:42 endrin iview_sm[21458]: endrin; MSG:NOTICE|SM:endrin:port 2|COND:#8
Fabric initialization error|NODE:blackberry:port
1:0x0011750000ffd7af|LINKEDTO:InfiniCon System InfinIO 9024 Lite:port
18:0x00066a00d9000108|DETAIL:Failed to set portinfo for node
```

B.1.2.9 #9 Link Integrity Error

B.1.2.9.1 Severity

Notice

B.1.2.9.2 Explanation

The SM received an asynchronous trap from a switch or end-port indicating a link integrity problem.

B.1.2.9.3 Action

The administrator should perform the fabric troubleshooting procedure to isolate and repair the faulty component. This is typically due to a bad cable, an incorrect cable being used for the signaling rate and cable length (For example, too small a wire gauge), or a hardware failure on one of the two HCA ports.

B.1.2.9.4 Example

```
Dec  5 14:28:10 i9k123 slot101:172.21.1.66;MSG:NOTICE|SM:SilverStorm 9024
GUID=0x00066a00d8000123:port 0|COND:#9 Link integrity/symbol
error|NODE:SilverStorm 9240 GUID=0x00066a000300016d Spine 2, Chip A:port
16:0x00066a000600056d|LINKEDTO:SilverStorm 9240 GUID=0x00066a000300016d Leaf 22,
Chip A:port 17:0x00066a0007000b33|DETAIL:Received LOCAL LINK INTEGRITY trap from
LID=0x007
```

B.1.2.10 #10 Security Error

B.1.2.10.1 Severity

Notice

B.1.2.10.2 Explanation

The SM received an asynchronous trap from a switch or end-port indicating a management key violation.

B.1.2.10.3 Action

The administrator should validate that the software configuration has not changed, because this issue is most likely due to a configuration issue. However, this event could also indicate a more serious issue such as a hacking attempt.

B.1.2.10.4 Example

```
Dec  5 14:28:10 i9k123 slot101:172.21.1.66;MSG:NOTICE|SM:SilverStorm 9024
GUID=0x00066a00d8000123:port 0|COND:#10 Security Error|NODE:SilverStorm 9240
GUID=0x00066a000300016d Spine 2, Chip A:port
16:0x00066a000600056d|LINKEDTO:SilverStorm 9240 GUID=0x00066a000300016d Leaf 22,
Chip A:port 17:0x00066a0007000b33|DETAIL:Received BAD MKEY trap from LID=0x007
```

B.1.2.11 #11 Other Exception

B.1.2.11.1 Severity

Notice

B.1.2.11.2 Explanation

The subnet manager encountered an error at some time after fabric initialization. Examples of possible errors are:

- The SM received an invalid request for information.
- The SM could not perform action requested by another fabric entity such as a request to create or join a multicast group with an unrealizable MTU or rate.

B.1.2.11.3 Action

The administrator should check to see if other SM related problems have occurred and perform the corrective actions for those items. If these other exceptions continue to persist, then customer support should be contacted.

B.1.2.11.4 Example

```
Dec  5 14:28:10 i9k123 slot101:172.21.1.66;MSG:NOTICE|SM:SilverStorm 9024
GUID=0x00066a00d8000123:port 0|COND:#11 Security Error|NODE:SilverStorm 9240
GUID=0x00066a000300016d Spine 2, Chip A:port
16:0x00066a000600056d|LINKEDTO:SilverStorm 9240 GUID=0x00066a000300016d Leaf 22,
Chip A:port 17:0x00066a0007000b33|DETAIL: Multicast group
0xFF12b6000000:00066a0007000b33 with mtu of 4k and rate of 20 has become
unrealizable due to fabric changes; current fabric mtu: 2k, current fabric rate: 20
```

B.1.2.12 #12 Fabric Summary

B.1.2.12.1 Severity

Notice

B.1.2.12.2 Explanation

A brief message describing the number of changes that the SM detected on its last subnet sweep. This message will include totals for the number of switches, HCAs, end-ports, total physical ports and SMs that have appeared or disappeared from the fabric. This message will only be logged at the end of a subnet sweep if the SM had detected changes.

B.1.2.12.3 Action

As this is only a summary of events detected during a fabric sweep, the administrator should examine the logs for preceding messages that describe the fabric changes in detail.

B.1.2.12.4 Example

```
Apr  8 15:31:36 endrin iview_sm[21458]: endrin; MSG:NOTICE|SM:endrin:port
2|COND:#12 Fabric Summary|NODE:endrin:port 2:0x00066a01a0000405|DETAIL:Change
Summary: 1 SWs disappeared, 0 HCAs appeared, 1 end ports disappeared, 3 total ports
disappeared, 0 SMs appeared
```

B.1.2.13 #13 SM State Change to Inactive

B.1.2.13.1 Severity

Notice

B.1.2.13.2 Explanation

Subnet manager transitioned from 'standby' into 'inactive' state.

B.1.2.13.3 Action

The administrator should check for inconsistencies in XML configurations between the master SM and this SM.

B.1.2.13.4 Example

```
Nov 29 12:15:28 endrin iview_sm[31247]:
;MSG:NOTICE|SM:endrin.dev.infiniconsys.com:port 1|COND:#13 SM state to
inactive|NODE:endrin.dev.infiniconsys.com:port
1:0x0x00066a00a0000405|DETAIL:transition from STANDBY to NOTACTIVE
```

B.1.2.14 #14 SM Inconsistency

B.1.2.14.1 Severity

Warning

B.1.2.14.2 Explanation

Deactivating the Standby Subnet Manager and Secondary Performance Manager due to inconsistent Subnet Manager XML configuration on Standby.

B.1.2.14.3 Action

If the condition persist, compare the XML configuration files between the master and standby SM for inconsistencies

B.1.2.14.4 Example

```
Oct 22 12:49:06 shaggy fm0_sm[31032]: shaggy; MSG:WARNING|SM:shaggy:port 1|COND:#14
SM standby configuration inconsistency|NODE:i9k118:port
0:0x00066a00d8000118|DETAIL:Deactivating standby SM i9k118 : 0x00066a00d8000118
which has a SM configuration inconsistency with master! The secondary PM will also
be deactivated.
```

B.1.2.15 #15 SM Virtual Fabric Inconsistency

B.1.2.15.1 Severity

Warning

B.1.2.15.2 Explanation

Deactivating the Standby Subnet Manager and Secondary Performance Manager due to inconsistent Subnet Manager Virtual Fabrics XML configuration on Standby.

B.1.2.15.3 Action

If the condition persist, compare the XML configuration files between the master and standby SM for inconsistencies.

B.1.2.15.4 Example

```
Oct 22 12:23:41 shaggy fm0_sm[30778]: shaggy; MSG:WARNING|SM:shaggy:port 1|COND:#15
SM standby virtual fabric configuration inconsistency|NODE:i9k118:port
0:0x00066a00d8000118|DETAIL:Deactivating standby SM i9k118 : 0x00066a00d8000118
which has a Virtual Fabric configuration inconsistency with master! The secondary
PM will also be deactivated.
```

B.1.2.16 #16 PM Inconsistency

B.1.2.16.1 Severity

Warning

B.1.2.16.2 Explanation

Deactivating the Secondary Performance Manager and Standby Subnet Manager due to inconsistent Performance Manager XML configuration on Secondary.

B.1.2.16.3 Action

If the condition persist, compare the XML configuration files between the primary and secondary PM for inconsistencies.

B.1.2.16.4 Example

```
Oct 22 12:51:42 shaggy fm0_sm[31173]: shaggy; MSG:WARNING|SM:shaggy:port 1|COND:#17
PM secondary configuration inconsistency|NODE:i9k118:port
0:0x00066a00d8000118|DETAIL:Attempting to deactivate secondary PM which has a
configuration inconsistency with primary! The standby SM will also be deactivated.
```

B.1.2.17 #17 BM Inconsistency

B.1.2.17.1 Severity

Warning

B.1.2.17.2 Explanation

Deactivating the Secondary Baseboard Manager due to inconsistent Baseboard Manager XML configuration on Secondary.

B.1.2.17.3 Action

If the condition persist, compare the XML configuration files between the primary and secondary BM for inconsistencies.

B.1.2.17.4 Example

```
Oct 22 12:53:45 shaggy fm0_sm[31314]: shaggy; MSG:WARNING|SM:shaggy:port 1|COND:#16
BM secondary configuration inconsistency|NODE:i9k118:port
0:0x00066a00d8000118|DETAIL:Attempting to deactivate secondary BM which has a
configuration inconsistency with primary!
```

B.2 Other Log Messages

In addition to the FM Event messages detailed in the previous section, the FM software suite may emit other log messages that provide extra detail for use by technical personnel in troubleshooting fabric issues.

B.2.1 Information (INFINI_INFO)

B.2.1.1 Switch node 'Sw1' (NodeGUID=0x00066a00d9000143,) has joined the fabric

B.2.1.1.1 SM Area

Discovery

B.2.1.1.2 Meaning

New switch has come online.

B.2.1.1.3 Action

None.

B.2.1.2 HCA node 'Hca1', port X (PortGUID=0x00066a00d9000143) has joined the fabric

B.2.1.2.1 SM Area

Discovery.

B.2.1.2.2 Meaning

New HCA port has come online.

B.2.1.2.3 Action

None.

B.2.1.3 Last full member of multicast group GID 0xff12401bffff0000:00000000ffffffff is no longer in fabric, deleting all members

B.2.1.3.1 SM Area

Discovery.

B.2.1.3.2 Meaning

The last full member of the group has left, The group is removed from the fabric.

B.2.1.3.3 Action

None.

B.2.1.4 topology_discovery: now running as a STANDBY SM

B.2.1.4.1 SM Area

Discovery.

B.2.1.4.2 Meaning

SM has transitioned to STANDBY mode.

B.2.1.4.3 Action

None.

B.2.1.5 TT: DISCOVERY CYCLE START

B.2.1.5.1 SM Area

Discovery.

B.2.1.5.2 Meaning

Discovery sweep has started.

B.2.1.5.3 Action

None.

B.2.1.6 TT, DISCOVERY CYCLE END

B.2.1.6.1 SM Area

Discovery.

B.2.1.6.2 Meaning

Discovery sweep has ended.

B.2.1.6.3 Action

None.

B.2.1.7 Port x of node [y] Hca1 belongs to another SM [0x0001]; Marking port as NOT MINE!

B.2.1.7.1 SM Area

Discovery.

B.2.1.7.2 Meaning

Usually happens during the merging of 2 fabrics.

B.2.1.7.3 Action

None.

B.2.1.8 createMCastGroups: vFabric VF0013 Multicast Group failure, multicast GID not configured

B.2.1.8.1 SM Area

Administrator.

B.2.1.8.2 Meaning

Attempt to pre-create mcast group failed due to configuration error. The default MGID is not configured.

B.2.1.8.3 Action

Configuration change required if pre-create is required.

B.2.1.9 sa_PathRecord: requested source Guid/Lid not found/active in current topology

B.2.1.9.1 SM Area

Administrator.

B.2.1.9.2 Meaning

May be caused by simultaneous removal/insertion events in the fabric.

B.2.1.9.3 Action

Check the health of the requester and the connected port if the message persist.

B.2.1.10 sa_PathRecord: requested destination GUID not an active port nor a Multicast Group

B.2.1.10.1 SM Area

Administrator.

B.2.1.10.2 Meaning

May be caused by simultaneous removal/insertion events in the fabric or the destination has dropped from fabric.

B.2.1.10.3 Action

None.

B.2.1.11 sa_XXXXXXXX: Can not find source lid of 0x0001 in topology in request to subscribe/unsubscribe...

B.2.1.11.1 SM Area

Administrator.

B.2.1.11.2 Meaning

May be caused by simultaneous removal/insertion events in the fabric or request received from the node that the SM has dropped from the fabric due to non-response to SMA queries.

B.2.1.11.3 Action

Check health of the node at lid 0x0001 if the fabric is stable.

B.2.1.12 sa_XXXXXXXX: requested source Lid/GUID not found/active in current topology

B.2.1.12.1 SM Area

Administrator.

B.2.1.12.2 Meaning

May be caused by simultaneous removal/insertion events in the fabric or request received from the node that the SM has dropped from the fabric due to non-response to SMA queries.

B.2.1.12.3 Action

Check the health of node at lid 0x0001 if the fabric is stable.

B.2.1.13 sa_McMemberRecord_Set: Port GID in request (0xFE8000000000000:00066a00d9000143) from Hca1, Port 0x00066a00d9000143, LID 0x0001, for group 0xFF12401BFFFF0000:00000000FFFFFFFF can't be found or not active in current topology, returning status 0x0001/0x0200

B.2.1.13.1 SM Area

Administrator.

B.2.1.13.2 Meaning

May be caused by simultaneous removal/insertion events in the fabric or request received from the node that the SM has dropped from the fabric due to the non-response to SMA queries.

B.2.1.13.3 Action

Check health of node at lid 0x0001 if fabric is stable.

B.2.1.14 sa_McMemberRecord_Set: Last full member left multicast group GID 0xFF12401BFFFF0000:00000000FFFFFFFF, deleting group and all members

B.2.1.14.1 SM Area

Administrator.

B.2.1.14.2 Meaning

Group is cleaned out when last the FULL member leaves.

B.2.1.14.3 Action

None.

B.2.2 Warning (WARN)

B.2.2.1 xxx Mismatch smKey[0x1] SMInfo from node Hca1 with, lid[0x1], guid 0x00066a00d9000143, TID=0x811E796027000000

B.2.2.1.1 SM Area

SM to SM Communication.

B.2.2.1.2 Meaning

The SMkey reported by the 2 SMs does not match.

B.2.2.1.3 Action

Reset the smkey on one of the subnet managers and restart.

B.2.2.2 failed to send reply [status=x] to SMInfo GET request from node Hca1 guid 0x00066a00d9000143, TID=0x811E796027000000

B.2.2.2.1 SM Area

SM to SM Communication.

B.2.2.2.2 Meaning

Lost communication path to other SM on node HCA1.

B.2.2.2.3 Action

Check the health of the node described in the message and status of the SM node.

B.2.2.3 failed to send reply [status=x] to SMInfo SET request from node Hca1 guid 0x00066a00d9000143, TID=0x811E796027000000

B.2.2.3.1 SM Area

SM to SM Communication.

B.2.2.3.2 Meaning

Lost communication path to the other SM on node HCA1.

B.2.2.3.3 Action

Check the health of the node described in the message and the status of the SM node.

B.2.2.4 SmInfo SET control packet not from a Master SM on node Hca1, lid [0x1], guid 0x00066a00d9000143, TID=0x811E796027000000

B.2.2.4.1 SM Area

SM to SM Communication.

B.2.2.4.2 Meaning

The SM on node HCA1 is violating the InfiniBand Architecture Specification protocol.

B.2.2.4.3 Action

If the condition persist, turn off the SM on node HCA1.

B.2.2.5 Standby SM received invalid AMOD[1-5] from SM node Hca1, LID [0x1], guid [0x00066a00d9000143], TID=0x811E796027000000

B.2.2.5.1 SM Area

SM to SM Communication.

B.2.2.5.2 Meaning

SM on node HCA1 is violating the InfiniBand Architecture Specification protocol.

B.2.2.5.3 Action

If the condition persist, turn off the SM on node HCA1.

B.2.2.6 MASTER SM did not receive response to Handover Acknowledgement from SM node Hca1, LID [0x1], guid [0x00066a00d9000143]

B.2.2.6.1 SM Area

SM to SM Communication.

B.2.2.6.2 Meaning

The SM on node HCA1 is incompatible or lost the communication path.

B.2.2.6.3 Action

Remove the incompatible SM from the fabric or check the health of node HCA1.

B.2.2.7 INACTIVE SM received invalid STANDBY transition request from SM node Hca1, LID [0x1], guid [0x00066a00d9000143], TID=0x811E796027000000

B.2.2.7.1 SM Area

SM to SM Communication.

B.2.2.7.2 Meaning

The SM on node HCA1 is violating the InfiniBand Architecture Specification protocol.

B.2.2.7.3 Action

If the condition persist, turn off the SM on node HCA1.

B.2.2.8 Master SM received invalid Handover Ack from remote SM Hca1, LID [0x1], guid [0x00066a00d9000143], TID=0x811E796027000000; remote not in STANDBY state [Discovering]

B.2.2.8.1 SM Area

SM to SM Communication.

B.2.2.8.2 Meaning

The SM on node HCA1 is violating the InfiniBand Architecture Specification protocol.

B.2.2.8.3 Action

If the condition persist, turn off the SM on node HCA1.

B.2.2.9 Master SM received invalid MASTER transition [requested state] from remote [remote state] SM Hca1, LID [0x1], guid [0x00066a00d9000143], TID=0x811E796027000000

B.2.2.9.1 SM Area

SM to SM Communication.

B.2.2.9.2 Meaning

The SM on node HCA1 is violating the InfiniBand Architecture Specification protocol.

B.2.2.9.3 Action

If the condition persist, turn off the SM on node HCA1.

B.2.2.10 Master SM did not receive response to Handover Acknowledgement from [remote state] SM node Hca1, LID [0x1], guid [0x00066a00d9000143]

B.2.2.10.1 SM Area

SM to SM Communication.

B.2.2.10.2 Meaning

Lost communication path to the other SM on node HCA1.

B.2.2.10.3 Action

Check the health of node described in the message and the status of the SM node.

B.2.2.11 SM at shaggy HCA-1, portGuid=0x0011750000ff8f4d has a different SM configuration consistency checksum [418863] from us [417845]

B.2.2.11.1 SM Area

SM to SM Communication.

B.2.2.11.2 Meaning

The SM on node HCA1 configuration does not match master.

B.2.2.11.3 Action

Verify that the XML configuration between master and standby SM is consistent.

B.2.2.12 No transitions allowed from DISCOVERING state; Got (ANY) request from [state] SM node Hca1, LID [0x1], guid [0x00066a00d9000143]

B.2.2.12.1 SM Area

SM to SM Communication.

B.2.2.12.2 Meaning

The SM on node HCA1 is violating the InfiniBand Architecture Specification protocol.

B.2.2.12.3 Action

If the condition persist, turn off the SM on node HCA1.

B.2.2.13 SmInfo from SM at SMLID[0x1] indicates SM is no longer master, switching to DISCOVERY state

B.2.2.13.1 SM Area

SM to SM Communication.

B.2.2.13.2 Meaning

Remote SM may have handed over to another SM on the fabric.

B.2.2.13.3 Action

None.

B.2.2.14 Switching to DISCOVERY state; Failed to get SmInfo from master SM at LID 0x1

B.2.2.14.1 SM Area

SM to SM Communication.

B.2.2.14.2 Meaning

Lost the communication path to the other SM at lid 0x1.

B.2.2.14.3 Action

Check the health of the node described in the message and the status of the SM node.

B.2.2.15 too many errors during sweep, will re-sweep in a few seconds

B.2.2.15.1 SM Area

Discovery.

B.2.2.15.2 Meaning

Multiple cable pulls or chassis removal/insertion event.

B.2.2.15.3 Action

Check links with high error count and reseat or replace cable, If the condition persist, capture the log information and call support.

B.2.2.16 unable to setup port [x] of node Sw1/Hca1, nodeGuid 0x00066a00d9000143, marking port down!

B.2.2.16.1 SM Area

Discovery.

B.2.2.16.2 Meaning

Lost communication path to node HCA1.

B.2.2.16.3 Action

Check the health of the node port described in the message.

B.2.2.17 Get NodeInfo failed for node off Port x of Node 0x00066a00d9000143:Hca1, status=7

B.2.2.17.1 SM Area

Discovery.

B.2.2.17.2 Meaning

The node connected to port x of HCA1 is not responding.

B.2.2.17.3 Action

Check the health of the node connected to the port.

B.2.2.18 Get NodeDesc failed for node off Port X of Node 0x00066a00d9000143:Hca1, status = 7

B.2.2.18.1 SM Area

Discovery.

B.2.2.18.2 Meaning

The node connected to port x of HCA1 is not responding.

B.2.2.18.3 Action

Check the health of the node connected to the port.

B.2.2.19 Failed to get Switchinfo for node sw1 guid 0x00066a00d9000143: status = 7

B.2.2.19.1 SM Area

Discovery.

B.2.2.19.2 Meaning

Switch node 1 is not responding.

B.2.2.19.3 Action

If the condition persist, check the health of the switch and capture health data if possible.

B.2.2.20 Failed to set Switchinfo for node sw1 nodeGuid 0x00066a00d9000143: status = 7, port mkey=0x0, SM mkey=0x0

B.2.2.20.1 SM Area

Discovery.

B.2.2.20.2 Meaning

Switch node 1 not responding.

B.2.2.20.3 Action

If the condition persist, check the health of the switch and capture health data if possible.

B.2.2.21 Failed to get PORTINFO from port 1 of node [x] Hca1; Marking port Down!

B.2.2.21.1 SM Area

Discovery.

B.2.2.21.2 Meaning

Port x of HCA1 not responding.

B.2.2.21.3 Action

Check the health of node HCA1.

B.2.2.22 Failed to init switch-to-switch link from node [x] sw1 guid 0x00066a00d9000143 port index X to node [x] sw2 guid 0x00066a00d9000144 port index Y which was reported down

B.2.2.22.1 SM Area

Discovery.

B.2.2.22.2 Meaning

May be caused by simultaneous removal/insertion events in the fabric.Persistence indicates a possible bad inter-switch cable or malfunctioning leaf or spine switch component.

B.2.2.22.3 Action

If the situation persist, perform a health check on the 2 switch ports. Replace the cable or switch component.

B.2.2.23 port on other side of node sw1 index x port X is not active

B.2.2.23.1 SM Area

Discovery.

B.2.2.23.2 Meaning

The node may have been marked down if it did not respond to SMA queries.

B.2.2.23.3 Action

Check the health of the node connected to switch 1 port X.

B.2.2.24 Node Hca1, port [1], NodeGuid 0x00066a00d9000143 is running at 1X width

B.2.2.24.1 SM Area

Discovery.

B.2.2.24.2 Meaning

Usually a bad cable. In rare instances, may be a bad port on a switch or HCA.

B.2.2.24.3 Action

Replace the cable. If it is a physical port issue, replacement of the HCA or switch module is required.

B.2.2.25 Node Hca1 [0x00066a00d9000143] port[x] returned MKEY[0x1] when MKEY[0x0] was requested!

B.2.2.25.1 SM Area

Discovery.

B.2.2.25.2 Meaning

Another SM with a different Mkey configured.

B.2.2.25.3 Action

Stop one of the subnet managers and make the configuration consistent.

B.2.2.26 Failed to get/set SLVL Map for node Hca1 nodeGuid 0x00066a00d9000143 output port X

B.2.2.26.1 SM Area

Discovery.

B.2.2.26.2 Meaning

May be caused by simultaneous removal/insertion events in the fabric. Persistence indicates that the node may be having problems.

B.2.2.26.3 Action

Check the health of the node HCA1/SW1 if the fabric was idle or persistent condition.

B.2.2.27 Failed to get/set SLVL Map for switch node sw1 nodeGuid 0x00066a00d9000143 output port X

B.2.2.27.1 SM Area

Discovery.

B.2.2.27.2 Meaning

May be caused by simultaneous removal/insertion events in the fabric. Persistence indicates that the node may be having problems.

B.2.2.27.3 Action

Check the health of node HCA1/SW1 if the fabric was idle or persistent condition.

B.2.2.28 Cannot get PORTINFO for node Hca1 nodeGuid 0x00066a00d9000143 port X status=Y

B.2.2.28.1 SM Area

Discovery.

B.2.2.28.2 Meaning

May be caused by simultaneous removal/insertion events in the fabric. Persistence indicates that the node may be having problems.

B.2.2.28.3 Action

Check the health of the node HCA1/SW1 if the fabric was idle or persistent condition.

B.2.2.29 Cannot set PORTINFO for node Hca1 nodeGuid 0x00066a00d9000143 port X status=Y

B.2.2.29.1 SM Area

Discovery.

B.2.2.29.2 Meaning

May be caused by simultaneous removal/insertion events in the fabric. Persistence indicates the node may be having problems.

B.2.2.29.3 Action

Check the health of node HCA1/SW1 if the fabric was idle or persistent condition.

B.2.2.30 Could not find neighborSw for switch node [x], port [y] in new topology; spanning tree not up to date

B.2.2.30.1 SM Area

Discovery.

B.2.2.30.2 Meaning

Caused by simultaneous removal/insertion events in the fabric.

B.2.2.30.3 Action

None.

B.2.2.31 failed to send DR getGuidInfo request to node Hca1, LID 0x1, nodeGuid 0x00066a00d9000143, portGuid 0x00066a00d9000143

B.2.2.31.1 SM Area

Discovery.

B.2.2.31.2 Meaning

May be caused by simultaneous removal/insertion events in the fabric. Persistence indicates that the node may be having problems.

B.2.2.31.3 Action

Check the health of the node HCA1/SW1 if the fabric was idle or persistent condition.

B.2.2.32 setting of port GUIDINFO failed for DR getGuidInfo request to node Hca1, LID 0x1, nodeGuid 0x00066a00d9000143, portGuid 0x00066a00d9000143

B.2.2.32.1 SM Area

Discovery.

B.2.2.32.2 Meaning

May be caused by simultaneous removal/insertion events in the fabric. Persistence indicates that the node may be having problems.

B.2.2.32.3 Action

Check the health of the node HCA1/SW1 if the fabric was idle or persistent condition.

B.2.2.33 failed to send async LR getGuidInfo request to node Hca1, LID 0x1, nodeGuid 0x00066a00d9000143, portGuid 0x00066a00d9000143

B.2.2.33.1 SM Area

Discovery.

B.2.2.33.2 Meaning

May be caused by simultaneous removal/insertion events in the fabric. Persistence indicates that the node may be having problems.

B.2.2.33.3 Action

Check the health of the node HCA1/SW1 if the fabric was idle or persistent condition.

B.2.2.34 Switch node 'Sw1' (NodeGUID=0x00066a00d9000143) has disappeared from fabric

B.2.2.34.1 SM Area

Discovery.

B.2.2.34.2 Meaning

Switch node 1 has dropped off the fabric.

B.2.2.34.3 Action

Check the health of the switch if the drop is not intentional.

B.2.2.35 HCA node 'Hca1', port X (PortGUID=0x00066a00d9000143) has disappeared from fabric

B.2.2.35.1 SM Area

Discovery.

B.2.2.35.2 Meaning

HCA node HCA1, port X has dropped off the fabric.

B.2.2.35.3 Action

Check the health of HCA1 port X if the drop is not intentional.

B.2.2.36 sa_NodeRecord_GetTable: Invalid node type[~1-3] in request from lid 0x1

B.2.2.36.1 SM Area

Administrator.

B.2.2.36.2 Meaning

Invalid data in the SA request.

B.2.2.36.3 Action

Check the health of the requester at lid 0x1.

B.2.2.37 sa_PathRecord_Set: Cannot find path to port 0x00066a00d9000144 from port 0x00066a00d9000143: failing src/dst pkey 0x800d validation

B.2.2.37.1 SM Area

Administrator.

B.2.2.37.2 Meaning

The source and destination do not share a partition with the given PKey

B.2.2.37.3 Action

Configuration change required if they should have access.

B.2.2.38 sa_PathRecord_Set: Cannot find path to port 0x00066a00d9000144 from port 0x00066a00d9000143: failing req/dst pkey validation

B.2.2.38.1 SM Area

Administrator.

B.2.2.38.2 Meaning

The requesting node and destination do not share a PKey.

B.2.2.38.3 Action

Configuration change required if they should have access.

B.2.2.39 sa_PathRecord_Set: Cannot find path to port 0x00066a00d9000144 from port 0x00066a00d9000143: failing vFabric serviceId validation

B.2.2.39.1 SM Area

Administrator.

B.2.2.39.2 Meaning

A path record request was made for a given serviceId. The source and destination do not share a vFabric that contains an application with the given serviceId.

B.2.2.39.3 Action

Configuration change required if they should have access.

B.2.2.40 sa_PathRecord_Set: Cannot find path to port 0x00066a00d9000144 from port 0x00066a00d9000143: failing vFabric rate validation (mtu=2,rate=3)

B.2.2.40.1 SM Area

Administrator.

B.2.2.40.2 Meaning

The source and destination do not share a path in a vFabric that contains limitations on max mtu and rate.

B.2.2.40.3 Action

Configuration change required if path is valid.

B.2.2.41 sa_PathRecord/SA_TraceRecord: Failed PKey check for src, input PKey is 0x800d

B.2.2.41.1 SM Area

Administrator.

B.2.2.41.2 Meaning

A request was for given pkey, but source of query is not a member of the same partition.

B.2.2.41.3 Action

Configuration change may be necessary.

B.2.2.42 sa_PathRecord/SA_TraceRecord: Failed pairwise PKey check for request

B.2.2.42.1 SM Area

Administrator.

B.2.2.42.2 Meaning

A query request failed pairwise pkey checks.

B.2.2.42.3 Action

Configuration change may be necessary.

B.2.2.43 sm_process_vf_info: Virtual Fabric VF0011 has undefined pkey. Changing pkey value to 0x3.

B.2.2.43.1 SM Area

Configuration

B.2.2.43.2 Meaning

A vFabric with an undefined pkey has been assigned a pkey.

B.2.2.43.3 Action

None.

B.2.2.44 sm_process_vf_info: Default PKey not being used by Default Virtual Fabric (configured as 0x8001). Changing pkey value to default 0x7fff

B.2.2.44.1 SM Area

Administrator.

B.2.2.44.2 Meaning

The Default Virtual Fabric was configured with a non-default pkey. Changing the pkey to the default.

B.2.2.44.3 Action

None.

B.2.2.45 sa_ServiceRecord_GetTable: Filter serviced record ID=0x1000000000003531 from lid 0x4 due to pkey mismatch from request port

B.2.2.45.1 SM Area

Administrator.

B.2.2.45.2 Meaning

PKey validation failed for service record, request node does not have valid pkey.

B.2.2.45.3 Action

Configuration change required if request should be valid.

B.2.2.46 sa_xxxxxx: too many records for SA_CM_GET

B.2.2.46.1 SM Area

Administrator.

B.2.2.46.2 Meaning

May have duplicate data in the fabric.

B.2.2.46.3 Action

Check topology data for duplicate GUIDs.

B.2.2.47 sa_TraceRecord/Pathrecord_set: Cannot find path to port 0x00066a00d9000144 from port 0x00066a00d9000144: LFT entry for destination is 255 from switch Sw1 (nodeGuid 0x00066a00d9000999)

B.2.2.47.1 SM Area

Administrator.

B.2.2.47.2 Meaning

May be caused by simultaneous removal/insertion events in the fabric.

B.2.2.47.3 Action

Check SW1 for a bad port and health of the destination node if the condition persists.

B.2.2.48 sa_TraceRecord/Pathrecord_set: Cannot find path to destination port 0x00066a00d9000144 from source port 0x00066a00d9000143; INVALID TOPOLOGY, next/last_nodep is NULL

B.2.2.48.1 SM Area

Administrator.

B.2.2.48.2 Meaning

May be caused by simultaneous removal/insertion events in the fabric.

B.2.2.48.3 Action

Check SW1 for a bad port and health of the destination node if the condition persists.

B.2.2.49 sa_updateMcDeleteCountForPort: MC Dos threshold exceeded for: Node= HCA1, GUID=0x00066a00d9000143, PortIndex=1; bouncing port.

B.2.2.49.1 SM Area

Administrator.

B.2.2.49.2 Meaning

SM Multicast denial of service configured and threshold has been reached. Bouncing the port in attempt to clear issue.

B.2.2.49.3 Action

If multiple occurrence, check health of node HCA1.

B.2.2.50 sa_updateMcDeleteCountForPort: MC Dos threshold exceeded for: Node= HCA1, GUID=0x00066a00d9000143, PortIndex=1; disabling port.

B.2.2.50.1 SM Area

Administrator.

B.2.2.50.2 Meaning

SM Multicast denial of service configured and threshold has been reached. disabling the port.

B.2.2.50.3 Action

Check the health of the node HCA1.

B.2.3 Error

B.2.3.1 could not perform HANDOVER to remote SM Hca1: 0x00066a00d9000143

B.2.3.1.1 SM Area

SM to SM communication.

B.2.3.1.2 Meaning

Lost communication path to the other SM on node guid 0x00066a00d9000143.

B.2.3.1.3 Action

Check the health of the node HCA1 and status of the SM node.

B.2.3.2 topology_initialize: can't get PortInfo, sleeping

B.2.3.2.1 SM Area

Discovery.

B.2.3.2.2 Meaning

topology_initialize: cannot get PortInfo; sleeping.

B.2.3.2.3 Action

Make sure the stack is running. Restart the SM node and stack.

B.2.3.3 topology_initialize: port state < INIT, sleeping

B.2.3.3.1 SM Area

Discovery.

B.2.3.3.2 Meaning

The node port of the SM is down.

B.2.3.3.3 Action

Be certain the host cable is connected to a switch (host SM only).

B.2.3.4 topology_initialize: can't get/set isSM, sleeping

B.2.3.4.1 SM Area

Discovery.

B.2.3.4.2 Meaning

SM cannot communicate with the stack.

B.2.3.4.3 Action

Make sure the stack is running. Restart the SM node and stack.

B.2.3.5 topology_discovery: can't setup my port, sleeping

B.2.3.5.1 SM Area

Discovery.

B.2.3.5.2 Meaning

The SM cannot communicate with stack.

B.2.3.5.3 Action

Make sure the stack is running. Restart the SM node and stack.

B.2.3.6 sm_set_node: Get NodeInfo failed for local node. status 7

B.2.3.6.1 SM Area

Discovery.

B.2.3.6.2 Meaning

The SM cannot communicate with the stack.

B.2.3.6.3 Action

If condition persist, restart the SM node.

B.2.3.7 sm_setup_node: Get NodeDesc failed for local node, status 7

B.2.3.7.1 SM Area

Discovery.

B.2.3.7.2 Meaning

The SM cannot communicate with the stack.

B.2.3.7.3 Action

If the condition persists, restart the SM node.

B.2.3.8 Error adding Node GUID: 0x00066a00d9000143 to tree. Already in tree!

B.2.3.8.1 SM Area

Discovery.

B.2.3.8.2 Meaning

Duplicate Node GUID in fabric.

B.2.3.8.3 Action

Using fabric tools, locate the device with the duplicate node GUID and remove it.

B.2.3.9 Error adding Port GUID: 0x00066a00d9000143 to tree. Already in tree!

B.2.3.9.1 SM Area

Discovery.

B.2.3.9.2 Meaning

Duplicate Port GUID in the fabric.

B.2.3.9.3 Action

Using fabric tools, locate the device with the duplicate port GUID and remove it.

B.2.3.10 Duplicate NodeGuid for Node Hca1 nodeType[1-3] guid 0x00066a00d9000143 and existing node[x] nodeType=1-3, Hca2, guid 0x00066a00d9000143

B.2.3.10.1 SM Area

Discovery.

B.2.3.10.2 Meaning

A duplicate Node Guid in fabric.

B.2.3.10.3 Action

Using fabric tools, locate the device with the duplicate node GUID and remove it.

B.2.3.11 Marking port[x] of node[x] Hca1 guid 0x00066a00d9000143 DOWN in the topology

B.2.3.11.1 SM Area

Discovery.

B.2.3.11.2 Meaning

port x of HCA1 is not responding.

B.2.3.11.3 Action

Check the health of node HCA1.

B.2.3.12 Failed to init SLVL Map (setting port down) on node Hca1/sw1 nodeGuid 0x00066a00d9000143 node index X port index Y

B.2.3.12.1 SM Area

Discovery.

B.2.3.12.2 Meaning

May be caused by simultaneous removal/insertion events in the fabric. Persistence indicates that the node may be having problems.

B.2.3.12.3 Action

Check the health of node Hca1/Sw1 if the fabric was idle or persistent condition.

B.2.3.13 Failed to init VL Arb (setting port down) on node Hca1/Sw1 nodeGuid 0x00066a00d9000143 node index X port index Y

B.2.3.13.1 SM Area

Discovery.

B.2.3.13.2 Meaning

May be caused by simultaneous removal/insertion events in the fabric. Persistence indicates that the node may be having problems.

B.2.3.13.3 Action

Check the health of node Hca1/Sw1 if the fabric was idle or persistent condition.

B.2.3.14 TT(ta): can't ARM/ACTIVATE node Hca1/sw1 guid 0x00066a00d9000143 node index X port index Y

B.2.3.14.1 SM Area

Discovery.

B.2.3.14.2 Meaning

May be caused by simultaneous removal/insertion events in the fabric. Persistence indicates that the node may be having problems.

B.2.3.14.3 Action

Check the health of node Hca1/Sw1 if the fabric was idle or persistent condition.

B.2.3.15 sa_xxxxx: Reached size limit at X records

B.2.3.15.1 SM Area

Administrator.

B.2.3.15.2 Meaning

The response buffer is too large.

B.2.3.15.3 Action

Contact support.

B.2.3.16 sa_NodeRecord_Set: NULL PORTGUID for Node Guid[0x00066a00d9000143], Hca1, Lid 0x1

B.2.3.16.1 SM Area

Administrator.

B.2.3.16.2 Meaning

Possible data corruption.

B.2.3.16.3 Action

Contact support.

B.2.3.17 sa_TraceRecord: destination port is not in active state; port LID: 0x1 (port GUID 0x00066a00d9000144)

B.2.3.17.1 SM Area

Administrator.

B.2.3.17.2 Meaning

May be caused by simultaneous removal/insertion events in the fabric.

B.2.3.17.3 Action

Check the health of the destination if the condition persist.

B.2.3.18 sa_TraceRecord: Cannot find path to port LID 0x2 (port guid 0x00066a00d9000144) from port LID 0x1 (port guid 0x00066a00d9000143)

B.2.3.18.1 SM Area

Administrator.

B.2.3.18.2 Meaning

May be caused by simultaneous removal/insertion events in the fabric.

B.2.3.18.3 Action

Check for the next 3 messages.

B.2.3.19 sa_TraceRecord_Fill: Reached size limit while processing TRACE_RECORD request

B.2.3.19.1 SM Area

Administrator.

B.2.3.19.2 Meaning

The response buffer is too large.

B.2.3.19.3 Action

Contact Support.

B.2.3.20 sa_PathRecord: NULL PORTGUID in Source/Destination Gid 0xFE80000000000000:0000000000000000 of PATH request from Lid 0x1

B.2.3.20.1 SM Area

Administrator.

B.2.3.20.2 Meaning

Invalid data in the SA request.

B.2.3.20.3 Action

Check the health of the requester at lid 0x1.

B.2.3.21 sa_PathRecord: Cannot find path to port LID 0x2 (port guid 0x00066a00d9000144) from port LID 0x1 (port guid 0x00066a00d9000143)

B.2.3.21.1 SM Area

Administrator.

B.2.3.21.2 Meaning

May be caused by simultaneous removal/insertion events in the fabric.

B.2.3.21.3 Action

Check the health of the destination lid 0x2.

B.2.3.22 sa_PathRecord: Cannot find path to port LID 0x2 (port guid 0x00066a00d9000144) from port LID 0x1 (port guid 0x00066a00d9000143) with pkey 0x800d

B.2.3.22.1 SM Area

Administrator.

B.2.3.22.2 Meaning

A path does not exist in the partition with the given PKey between the given source and destination.

B.2.3.22.3 Action

Check configuration to determine if path should exist in given pkey. Check the health of the destination lid 0x2 if configuration is valid.

B.2.3.23 sa_PathRecord: port LID 0x1 (port guid 0x00066a00d9000143) not a member of multicast group 0xff12401bffff0000:00000000ffffffff

B.2.3.23.1 SM Area

Administrator.

B.2.3.23.2 Meaning

Group may have just been deleted or the requester is not a member of the group.

B.2.3.23.3 Action

None.

B.2.3.24 sa_McMemberRecord_Set: Port GID in request (0x0080000000000000:0x0000000000000000)from Hca1, Port 0x00066a00d9000143, LID 0x1 has a NULL GUID/invalid prefix, returning status 0x0500

B.2.3.24.1 SM Area

Administrator.

B.2.3.24.2 Meaning

Invalid data in the SA request.

B.2.3.24.3 Action

Check the health of HCA1.

B.2.3.25 sa_McMemberRecord_Set: MTU selector of 2 with MTU of 4 does not work with port MTU of 1 for request from compute-0-24, Port 0x00066A00A00005C5, LID 0x009C, returning status 0x0200

B.2.3.25.1 SM Area

Administrator.

B.2.3.25.2 Meaning

The requester port data is not compatible with the group data.

B.2.3.25.3 Action

Create group at lowest common denominator or host should join with rate selector of “less than” rather than “exactly”.

B.2.3.26 sa_McMemberRecord_Set: Rate selector of 2 with Rate of 3 does not work with port Rate of 2 for request from compute-0-24, Port 0x00066A00A00005C5, LID 0x009C, returning status 0x0200

B.2.3.26.1 SM Area

Administrator.

B.2.3.26.2 Meaning

Node compute-0-24 has requested a port rate that is incompatible with the group rate.

B.2.3.26.3 Action

Check that the requester port is not running at 1X width or that the multicast group was not created with a rate greater than what some the host ports can support.

B.2.3.27 sa_McMemberRecord_Set: Component mask of 0x00000000000XXXXX does not have bits required (0x00000000000130C6) to create a group for new MGID of 0xFF12401BFFFF0000:00000000FFFFFFFF for request from Hca1, Port 0x00066a00d9000143

B.2.3.27.1 SM Area

Administrator.

B.2.3.27.2 Meaning

End node may be trying to join a group that does not exist.

B.2.3.27.3 Action

OpenIB and Sun stacks require that the broadcast group be pre-created by the SM.

B.2.3.28 sa_McMemberRecord_Set: Component mask of 0x0000000000010083 does not have bits required (0x00000000000130C6) to create a new group in request from Hca1, Port 0x00066a00d9000143

B.2.3.28.1 SM Area

Administrator.

B.2.3.28.2 Meaning

Specific bits must be set in a CREATE group request.

B.2.3.28.3 Action

The requester is violating the InfiniBand Architecture Specification protocol.

B.2.3.29 sa_McMemberRecord_Set: Bad (limited member) PKey of 0x1234 for request from ibhollab54 HCA-1, Port 0x00066a00d9000143, LID 0x1, returning status 0x200

B.2.3.29.1 SM Area

Administrator.

B.2.3.29.2 Meaning

The PKey specified in request was limited, it should be full.

B.2.3.29.3 Action

Check configuration.

B.2.3.30 sa_McMemberRecord_Set: MC group create request denied for node ibhollab54 HCA-1, port 0x00066a00d9000144 from lid 0x2, failed VF validation (mgid=0xFF12401BFFFF0000:0x0000000000000016)

B.2.3.30.1 SM Area

Administrator.

B.2.3.30.2 Meaning

An attempt to create an mcast group failed due to validation failures.

B.2.3.30.3 Action

Check configuration if create by source should be valid.

B.2.3.31 Invalid MGID (0xFF270000FFFF0000:00000000FFFFFFFF) in CREATE/JOIN request from Hca1, Port 0x00066a00d9000143, LID 0x0001, returning status 0x0500

B.2.3.31.1 SM Area

Administrator.

B.2.3.31.2 Meaning

MGID requested violates InfiniBand Architecture Specification protocol.

B.2.3.31.3 Action

The requester is violating the InfiniBand Architecture Specification protocol.

B.2.3.32 Join state of 0x1-2 not full member for NULL/NEW GID request from Hca1, Port 0x00066a00d9000143, LID 0x0001, returning status 0x0200

B.2.3.32.1 SM Area

Administrator.

B.2.3.32.2 Meaning

Creation of a Multicast Group requires FULL membership.

B.2.3.32.3 Action

The requester is violating the InfiniBand Architecture Specification protocol.

B.2.3.33 Join state of ~0x1 not full member for request to CREATE existing MGID of 0xFF12401BFFFF0000:00000000FFFFFFFF

B.2.3.33.1 SM Area

Administrator.

B.2.3.33.2 Meaning

Creation of a Multicast Group requires FULL membership.

B.2.3.33.3 Action

The requester is violating the InfiniBand Architecture Specification protocol.

B.2.3.34 sa_McMemberRecord_Set: Component mask of 0x00000000000XXXXX does not have bits required (0x0000000000010083) to JOIN group with MGID 0xFF12401BFFFF0000:00000000FFFFFFFF in request from %s, Port 0x%.16"CS64"X, LID 0x%.4X, returning status 0x%.4X

B.2.3.34.1 SM Area

Administrator.

B.2.3.34.2 Meaning

Specific bits must be set in a JOIN group request.

B.2.3.34.3 Action

The requester is violating the InfiniBand Architecture Specification protocol.

B.2.3.35 Maximum number groups reached (1024), failing CREATE request from Hca1, Port 0x00066a00d9000143, LID 0x0011, returning status 0x0100

B.2.3.35.1 SM Area

Administrator.

B.2.3.35.2 Meaning

No resources.

B.2.3.35.3 Action

Delete some of the multicast groups or configure the SM to overload MLIDs during a group creation.

B.2.3.36 Failed to assign GID for CREATE request from Hca1, Port 0x00066a00d9000143, LID 0x0001, returning status 0x0100

B.2.3.36.1 SM Area

Administrator.

B.2.3.36.2 Meaning

No resources.

B.2.3.36.3 Action

Delete some of the multicast groups or configure the SM to overload the MLIDs during group creation.

B.2.3.37 No multicast LIDs available for request from Hca1, Port 0x00066a00d9000143, LID 0x0001, returning status 0x0100

B.2.3.37.1 SM Area

Administrator.

B.2.3.37.2 Meaning

No resources.

B.2.3.37.3 Action

Delete some of the multicast groups or configure the SM to overload the MLIDs during group creation.

B.2.3.38 MGID 0xFF12401BFFFF0000:00000000FFFFFFFF does not exist; Failing JOIN request from Hca1, Port 0x00066a00d9000143, LID 0x0001, returning status 0x0200

B.2.3.38.1 SM Area

Administrator.

B.2.3.38.2 Meaning

JOIN of a group that does not exist.

B.2.3.38.3 Action

OpenIB and Sun stacks require that the broadcast group be pre-created by SM.

B.2.3.39 Qkey/Pkey of 0x1234 does not match group QKey of 0x4321 for group 0xFF12401BFFFF0000:00000000FFFFFFFF for request from Hca1, Port 0x00066a00d9000143, LID 0x0001, returning status 0x0200

B.2.3.39.1 SM Area

Administrator.

B.2.3.39.2 Meaning

SM may have been set to create the default broadcast group with parameters not valid for the fabric.

B.2.3.39.3 Action

Reconfigure the default broadcast group with the proper parameters.

B.2.3.40 Group Rate/MTU of 5 greater than requester port mtu of 2/4 for group 0xFF12401BFFFF0000:00000000FFFFFFFF for request from Hca1, Port 0x00066a00d9000143, LID 0x0001, returning status 0x0200

B.2.3.40.1 SM Area

Administrator.

B.2.3.40.2 Meaning

SM may have been set to create the default broadcast group with parameters not valid for fabric.

B.2.3.40.3 Action

Reconfigure the default broadcast group with the proper parameters.

B.2.3.41 Group Rate/MTU of X is too low/high for requested rate/mtu of Y, rate/mtu selector of 2, and port rate/mtu of Z for group 0xFF12401BFFFF0000:00000000FFFFFFFF in request from Hca1

B.2.3.41.1 SM Area

Administrator.

B.2.3.41.2 Meaning

The SM may have been set to create a default broadcast group with parameters not valid for a fabric or a host has created the group at a RATE not supported by other hosts.

B.2.3.41.3 Action

Create the group at lowest common denominator or the host should join with rate selector of “less than” rather “exactly”.

B.2.3.42 Subscription for security trap not from trusted source[lid=0x0001], smkey=0x0, returning status 0x0200

B.2.3.42.1 SM Area

Administrator.

B.2.3.42.2 Meaning

Requester using wrong smkey.

B.2.3.42.3 Action

Make sure to use the same SMkey as what is configured in the SM.

B.2.3.43 sm_process_vf_info: Virtual Fabric VF0001 has application SA selected, bad PKey configured 0x1, must use Default PKey.",

B.2.3.43.1 SM Area

Administrator.

B.2.3.43.2 Meaning

The SA select is limited to Virtual Fabrics using the Default PKey 0x7fff.

B.2.3.43.3 Action

Configuration change needed for SA Select.

B.2.3.44 sm_process_vf_info: Virtual Fabric VF0013 MulticastGroup configuration error, MGID does not match app, disabling Default Group

B.2.3.44.1 SM Area

Administrator.

B.2.3.44.2 Meaning

The Multicast Group has an MGID configured that does not match any application that is part of this Virtual Fabric.

B.2.3.44.3 Action

Configuration change needed for mcast group creation.

B.2.3.45 sm_process_vf_info: Virtual Fabric VF0013 MulticastGroup configuration error, mismatch on pkey. Disabling Default Group

B.2.3.45.1 SM Area

Administrator.

B.2.3.45.2 Meaning

The MulticastGroup linked to this Virtual Fabric do not share a common pkey. Disabling the mcast group for this vFabric.

B.2.3.45.3 Action

Configuration change required if mcast group default creation is needed.

B.2.3.46 sm_process_vf_info: Virtual Fabric VF0013 MulticastGroup configuration error, mismatch on mtu/rate. Disabling Default Group

B.2.3.46.1 SM Area

Administrator.

B.2.3.46.2 Meaning

The Multicast Group has mtu/rate configured higher than the max mtu/rate configured for the Virtual Fabric.

B.2.3.46.3 Action

Configuration change required if mcast group default creation is required.

B.2.3.47 sm_initialize_port/sm_dbsync: cannot refresh sm pkeys

B.2.3.47.1 SM Area

Administrator.

B.2.3.47.2 Meaning

An internal error occurred when attempting to refresh the SM Pkeys.

B.2.3.47.3 Action

Contact customer support if condition persists.

B.2.3.48 sa_ServiceRecord_Add: Failed to ADD serviced record ID=0x1000000000003531 from lid 0x2 due to invalid pkey

B.2.3.48.1 SM Area

Administrator.

B.2.3.48.2 Meaning

Serviced record add failure due to request with invalid PKey

B.2.3.48.3 Action

Configuration change required if request should be granted.

B.2.3.49 sa_ServiceRecord_Add: Failed to ADD serviced record ID=0x1000000000003531 from lid 0x2 due to pkey mismatch from request port

B.2.3.49.1 SM Area

Administrator.

B.2.3.49.2 Meaning

Serviced record add failure due to request with PKey not shared by requestor

B.2.3.49.3 Action

Configuration change required if request should be granted.

B.2.3.50 sa_ServiceRecord_Add: Failed to ADD serviced record ID=0x1000000000003531 from lid 0x2 due to pkey mismatch from service port

B.2.3.50.1 SM Area

Administrator.

B.2.3.50.2 Meaning

Serviced record add failure due to request with PKey not shared by requestor

B.2.3.50.3 Action

Configuration change required if request should be granted.

§ §

Appendix C Mapping Old Parameters

Versions of the FM prior to version 4.4 used a different configuration file format (`iview_fm.config`). Table 42 provides a mapping of parameter names in the old file to the new file (`ifs_fm.xml`). Refer to Section 4.1, "Configuring the FM" on page 63 for more information about the xml configuration file format.

Parameters in the old format included the instance number in their name, for example `sm_0_hca`. These are shown as `sm_x_parameter` in the following tables.

The section names are shown as part of the new parameter. The `Shared.SubnetSize` indicates the SubnetSize parameter within the Shared section.

Note: When performing an upgrade installation of the FM, the existing `iview_fm.config` will be translated to create a new `ifs_fm.xml` configuration file. The user will have the option to retain this translated upgrade or use the new `ifs_fm.xml-sample` file as the default. Refer to the *Intel® True Scale Fabric Software Installation Guide* for more information. If required, the `config_convert` utility can also be used to convert existing `iview_fm.config` files into the `ifs_fm.xml` configuration file format.

C.1 Old Global Parameters

Table 42. Mapping of Old Global Parameters to New

Old Parameter	New Parameter	Notes
IVIEW_FM_BASE	—	Removed parameter
ENV_OVERWRITE	—	Removed parameter
HARDWARE_TYPE	—	Removed parameter
START_AFTER_ERROR	—	Removed parameter
STARTUP_LOG	—	Removed parameter
SUBNET_SIZE	Shared.SubnetSize	—
SYSLOG_MODE	Shared.LogMode	—

C.2 Old SM Parameters

Table 43. Mapping of Old SM Parameters to New

Old Parameter	New Parameter	Notes
SM_x_start	Sm.Start	Depends on Fm.Start
SM_x_device	Fm.Shared.Hca	In old file 0 was 1st HCA, in new file 1 is 1st HCA
SM_x_port	Fm.Shared.Port	—
SM_x_portGuid	Fm.Shared.PortGUID	—
SM_x_gidprefix	Fm.Shared.SubnetPrefix	—
—	Fm.Name	New parameter
SM_x_debug	Sm.Debug	Also in Shared
SM_x_key	Sm.SmKey	—
SM_x_mkey	Sm.MKey	—

Table 43. Mapping of Old SM Parameters to New (Continued)

Old Parameter	New Parameter	Notes
SM_x_priority	Sm.Priority	Recommend to set in Shared section
SM_x_elevated_priority	Sm.ElevatedPriority	Recommend to set in Shared section
SM_x_timer	Sm.SweepInterval	—
SM_x_max_retries	Sm.MaxAttempts	—
SM_x_rcv_wait_msec	Sm.RespTimeout	—
—	MinRespTimeout	New capability
SM_x_master_ping_interval	Sm.MasterPingInterval	—
SM_x_master_ping_max_fail	Sm.MasterPingMaxFail	—
SM_x_topo_errors_threshold	Sm.SweepErrorsThreshold	—
SM_x_topo_abandon_threshold	Sm.SweepAbandonThreshold	—
SM_x_switchlifetime	Sm.SwitchLifetime	The old parameters were expressed as N with the resulting time-out being 4.096 * 2^N microseconds. The new parameters are expressed in "natural" format such as 33ms or infinite. Values will be rounded up as needed.
SM_x_hoqlife	Sm.HoqLife	The old parameters were expressed as N with the resulting time-out being 4.096 * 2^N microseconds. The new parameters are expressed in "natural" format such as 33ms or infinite. Values will be rounded up as needed.
SM_x_vlstall	Sm.VlStallCount	—
SM_x_saRespTime	Sm.SaRespTime	The old parameters were expressed as N with the resulting time-out being 4.096 * 2^N microseconds. The new parameters are expressed in "natural" format such as 33ms. Values will be rounded up as needed.
SM_x_saPacketLifetime	Sm.PacketLifetime	The old parameters were expressed as N with the resulting time-out being 4.096 * 2^N microseconds. The new parameters are expressed in "natural" format such as 33ms. Values will be rounded up as needed.
SM_x_dbsync_interval	Sm.DbSyncInterval	—
SM_x_trap_threshold	Sm.TrapThreshold	—
SM_x_node_appearance_msg_thresh	Sm.NodeAppearanceMsgThreshold	—
SM_x_1x_link_mode	Sm.Suppress1x	—
SM_x_spine_first_routing	Sm.SpineFirstRouting	—
SM_x_sma_batch_size	Sm.SmaBatchSize	—
SM_x_ehca_sma_batch_size	Sm.EhcaSmaBatchSize	—
SM_x_max_parallel_reqs	Sm.MaxParallelReqs	—

Table 43. Mapping of Old SM Parameters to New (Continued)

Old Parameter	New Parameter	Notes
SM_x_dynamicPlt	Sm.DynamicPacketLifetime.Enable	—
SM_x_dynamicPlt_yy	Sm.DynamicPacketLifetime.Hopsyy	yy can be 01 to 09. The old parameters were expressed as N with the resulting time-out being 4.096 * 2^N microseconds. The new parameters are expressed in "natural" format such as 33ms. Values will be rounded up as needed.
SM_x_lid	Sm.LID	—
SM_x_lmc	Sm.Lmc	—
SM_x_lmc_e0	Sm.LmcE0	New capability
SM_x_nodaemon	—	Removed parameter
SM_x_log_level	Sm.LogLevel	Recommend to set in Shared section
SM_x_sm_debug_perf	Sm.SmPerfDebug	—
SM_x_sa_debug_perf	Sm.SaPerfDebug	—
SM_x_sa_debug_rmpp	Sm.RmppDebug	Also in Shared
SM_x_sa_rmpp_checksum	Sm.SaRmppChecksum	—
SM_x_loop_test_on	Sm.LoopTestOn	—
SM_x_loop_test_packets	Sm.LoopTestPackets	—
SM_x_topo_lid_offset	Sm.LIDSpacing	—
SM_x_log_filter	Sm.LogFilter	Use of LogLevel is recommended
SM_x_log_mask	Sm.LogMask	Use of LogLevel is recommended
SM_x_log_file	Sm,LogFile	—
SM_x_pkey_yy	—	Removed parameter. yy is 00 to 31
SM_x_disable_mcast_check	Sm.Multicast.DisableStrictCheck	—
SM_x_def_mc_create	Sm.Multicast.MulticastGroup.Create	The new format allows specification of what MGIDs to create and can create many MGIDs as needed
SM_x_def_mc_pkey	Sm.Multicast.MulticastGroup.PKey	The PKey is optional in the new format. When omitted an appropriate PKey will be used
SM_x_def_mc_mtu	Sm.Multicast.MulticastGroup.Mtu	The old format used IBTA constants (1=256 bytes, 2=512, 3=1024, 4=2048, 5=4096). The new format allows for direct use of "natural" values such as 2048.
SM_x_def_mc_rate	Sm.Multicast.MulticastGroup.Rate	The old format used IBTA constants (2=2.5g, 3=10g, 4=30g, 5=5g, 6=20g, 7=40g, 8=60g, 9=80g, 10=120g). The new format allows for direct use of "natural" values such as 10g.
SM_x_def_mc_sl	Sm.Multicast.MulticastGroup.SL	The SL is optional in the new format. When omitted an appropriate service level will be used
SM_x_def_mc_qkey	Sm.Multicast.MulticastGroup.QKey	—

Table 43. Mapping of Old SM Parameters to New (Continued)

Old Parameter	New Parameter	Notes
SM_x_def_mc_fl	Sm.Multicast.MulticastGroup.FlowLabel	—
SM_x_def_mc_tc	Sm.Multicast.MulticastGroup.TClass	—
SM_x_non_resp_tsec	Sm.NonRespTimeout	—
SM_x_non_resp_max_count	Sm.NonRespMaxCount	—
SM_x_mcastMLidTableCap	Sm.Multicast.MLIDTableCap	—
SM_x_mcastGrpMGidLimitMask_y	Sm.Multicast.MLIDShare.MGIDMask	y can be 0 to 31, each y must appear in a separate MLIDShare section
SM_x_mcastGrpMGidLimitValue_y	Sm.Multicast.MLIDShare.MGIDValue	y can be 0 to 31, each y must appear in a separate MLIDShare section
SM_x_mcastGrpMGidLimitMax_y	Sm.Multicast.MLIDShare.MaxMLIDs	y can be 0 to 31, each y must appear in a separate MLIDShare section
SM_x_dynamic_port_alloc	Sm.DynamicPortAlloc	—

C.3 Old FE Parameters

Table 44. Mapping of Old FE Parameters to New

Old Parameter	New Parameter	Notes
FE_x_start	Fe.Start	Depends on Fm.Start
FE_x_device	Fm.Shared.Hca	In old file 0 was 1st HCA, in new file 1 is 1st HCA
FE_x_port	Fm.Shared.Port	—
FE_x_portGuid	Fm.Shared.PortGUID	—
FE_x_listen	Fe.TcpPort	—
FE_x_login	—	Removed parameter
FE_x_nodaemon	—	Removed parameter
FE_x_log_level	Fe.LogLevel	Recommend to set in Shared section
FE_x_debug	Fe.Debug	Also in Shared
FE_x_if3_debug_rmpp	Fe.RmppDebug	Also in Shared
FE_x_log_filter	Fe.LogFilter	Use of LogLevel is recommended
FE_x_log_mask	Fe.LogMask	Use of LogLevel is recommended
FE_x_log_file	Fe,LogFile	—
FE_x_default_pkey	Fe.DefaultPKey	Also in Shared

C.4 Old PM Parameters

Table 45. Mapping of Old PM Parameters to New

Old Parameter	New Parameter	Notes
PM_x_start	Pm.Start	Depends on Fm.Start
PM_x_device	Fm.Shared.Hca	In old file 0 was 1st HCA, in new file 1 is 1st HCA
PM_x_port	Fm.Shared.Port	—
PM_x_portGuid	Fm.Shared.PortGUID	—
PM_x_priority	Pm.Priority	Recommend to set in Shared section
PM_x_elevated_priority	Pm.ElevatedPriority	Recommend to set in Shared section
PM_x_timer	Pm.ServiceLease	—
PM_x_nodaemon	—	Removed parameter
PM_x_log_level	Pm.LogLevel	Recommend to set in Shared section
PM_x_debug	Pm.Debug	Also in Shared
PM_x_if3_debug_rmpp	Pm.RmppDebug	Also in Shared
PM_x_log_filter	Pm.LogFilter	Use of LogLevel is recommended
PM_x_log_mask	Pm.LogMask	Use of LogLevel is recommended
PM_x_log_file	Pm,LogFile	—
PM_x_default_pkey	Pm.DefaultPKey	Also in Shared

C.5 Old BM Parameters

Table 46. Mapping of Old BM Parameters to New

Old Parameter	New Parameter	Notes
BM_x_start	Bm.Start	Depends on Fm.Start
BM_x_device	Fm.Shared.Hca	In old file 0 was 1st HCA, in new file 1 is 1st HCA
BM_x_port	Fm.Shared.Port	—
BM_x_portGuid	Fm.Shared.PortGUID	—
BM_x_bkey_key	Bm.BKey	—
BM_x_bkey_lease	Bm.BKeyLease	—
BM_x_debug	Bm.Debug	Also in Shared
BM_x_priority	Bm.Priority	Recommend to set in Shared section
BM_x_elevated_priority	Bm.ElevatedPriority	Recommend to set in Shared section
BM_x_timer	—	Removed parameter
BM_x_nodaemon	—	Removed parameter
BM_x_log_level	Bm.LogLevel	Recommend to set in Shared section
BM_x_if3_debug_rmpp	Bm.RmppDebug	Also in Shared

Table 46. Mapping of Old BM Parameters to New (Continued)

Old Parameter	New Parameter	Notes
BM_x_debug_flag	—	Removed parameter
BM_x_log_filter	Bm.LogFilter	Use of LogLevel is recommended
BM_x_log_mask	Bm.LogMask	Use of LogLevel is recommended
BM_x_log_file	Bm,LogFile	—
BM_x_default_pkey	Bm.DefaultPKey	Also in Shared

§ §

Appendix D QOS Options in a Mesh/Torus vFabric

Fabric Topology	Num QOS VF level	Total Needed	Max MTU	Total DOR	Total Up/ Down	QOS VF 1	QOS VF 2	QOS VF 3	QOS VF 4	QOS VF 5+	Comments
nD Mesh	1	2 SL/2 VL	4K	1SL/1VL	1SL/1VL	2 SL/2 VL					
	2	4 SL/4 VL	4K	2 SL/2 VL	2 SL/2 VL	2 SL/2 VL	2 SL/2 VL				
	2	3 SL/3 VL	4K	1SL/1VL	2 SL/2VL	2 SL/2VL	1 SL/1VL				QOS VF2 has SecondaryRouteOnly
	3	6 SL/6VL	2K	3SL/3VL	3SL/3VL	2 SL/2 VL	2 SL/2 VL	2 SL/2 VL			
	3	4 SL/4 VL	4K	1SL/1VL	3SL/3VL	2 SL/2VL	1 SL/1VL	1 SL/1VL			QOS VF2 and QOS VF3 have SecondaryRouteOnly
	4	8 SL/8VL	2K	4SL/4VL	4SL/4VL	2 SL/2 VL	2 SL/2 VL	2 SL/2 VL	2 SL/2 VL		
	4	6 SL/6VL	2K	2 SL/2 VL	4SL/4VL	2 SL/2 VL	2 SL/2 VL	1 SL/1VL	1SL/1VL		QOS VF3 and QOS VF4 have SecondaryRouteOnly
	5+	7SL/7VL	2K	2 SL/2 VL	5SL/5VL	2 SL/2 VL	2 SL/2 VL	1 SL/1VL	1SL/1VL	1SL/1VL	QOS VF3, QOS VF4 and QOS VF5 have SecondaryRouteOnly
nD Mesh +1d Torus	1	3SL/3VL	4K	2SL/2VL	1SL/1VL	3SL/3VL					
	2	6SL/6VL	2K	4SL/4VL	2SL/2VL	3SL/3VL	3SL/3VL				
	2	4SL/4VL	4K	2SL/2VL	2SL/2VL	3SL/3VL	1SL/1VL				QOS VF2 has SecondaryRouteOnly
	3	9SL/8VL	2K	6SL/6VL	3SL/2VL	3SL/3VL	3SL/2VL	3SL/2VL			Share Up/Down VL for non-SA QOS VF
	3	7SL/7VL	2K	4SL/4VL	3SL/3VL	3SL/3VL	3SL/3VL	1SL/1VL			QOS VF3 has SecondaryRouteOnly
	4	8 SL/8VL	2K	4SL/4VL	4SL/4VL	3SL/3VL	3SL/3VL	1SL/1VL	1SL/1VL		QOS VF3 and QOS VF4 have SecondaryRouteOnly
	5+	7SL/7VL	2K	2SL/2VL	5SL/5VL	3SL/3VL	1SL/1VL	1SL/1VL	1SL/1VL	1SL/1VL	QOS VF3, QOS VF4 and QOS VF5 have SecondaryRouteOnly
nD Mesh +2d Torus	1	5SL/3VL	4K	4SL/2VL	1SL/1VL	5SL/3VL					
	2	10SL/6VL	2K	8SL/4VL	2SL/2VL	5SL/3VL	5SL/3VL				
	2	6SL/4VL	4K	4SL/2VL	2SL/2VL	5SL/3VL	1SL/1VL				QOS VF2 has SecondaryRouteOnly
	3	15SL/8VL	2K	12SL/6VL	3SL/2VL	5SL/3VL	5SL/2VL	5SL/2VL			Share Up/Down VL for non-SA QOS VF
	4	12SL/8VL	2K	8SL/4VL	4SL/4VL	5SL/3VL	5SL/3VL	1SL/1VL	1SL/1VL		QOS VF3 and QOS VF4 have SecondaryRouteOnly

Fabric Topology	Num QOS VF level	Total Needed	Max MTU	Total DOR	Total Up/ Down	QOS VF 1	QOS VF 2	QOS VF 3	QOS VF 4	QOS VF 5+	Comments
	5+	9SL/7VL	2K	4SL/2VL	5SL/5VL	5SL/3VL	1SL/1VL	1SL/1VL	1SL/1VL	1SL/1VL	QOS VF3, QOS VF4 and QOS VF5 have SecondaryRouteOnly
nD Mesh +3d Torus	1	9SL/3VL	4K	8SL/2VL	1SL/1VL	9SL/3VL					
	2	10SL/4VL	4K	8SL/2VL	2SL/2VL	9SL/3VL	1SL/1VL				QOS VF2 has SecondaryRouteOnly
	3+	11SL/5VL	2K	8SL/2VL	3SL/3VL	9SL/3VL	1SL/1VL	1SL/1VL			QOS VF2 and QOS VF3 have SecondaryRouteOnly

Key:

Total - total resources used/needed

Total DOR - part of total resources used for DOR routes

Total Up/Down - part of total resources used for Up/Down routes

QOS VF1 … QOS VF5 - resources exclusively assigned to given QOS VF level. Note that resources shared between multiple QOS VF levels only reflected in Totals

Appendix E ./INSTALL Command

E.1 Syntax

```
./INSTALL [-r root] [-v|-vv] [-a|-n|-U|-u|-s] [-E comp] [-D comp]
                  [--without-depcheck] [--force] [--answer
                  keyword=value]./INSTALL
```

or

```
./INSTALL -C
```

or

```
./INSTALL -V
```

Note: To access help for this command type `./INSTALL -?` and press **Enter**.

E.2 Options

`-a` – Install all ULPs and drivers with default options.

`-n` – Install all ULPs and drivers with default options but with no change to autostart options.

`-U` – Upgrades/re-install all presently installed ULPs and drivers with default options and no change to autostart options.

`-u` – Uninstall all ULPs and drivers with the default options.

`-s` – Enable autostart for all installed drivers.

`-E` *comp* – Enables autostart of given component. This option can appear with `-D` or multiple times on a command line.

This option can be combined with `-a`, `-n`, `-i`, `-e` and `-U` to permit control over which installed software will be configured for autostart.

`-D` *comp* – Disable autostart of given component. This option can appear with `-E` or multiple times once on command line.

This option can be combined with `-a`, `-n`, and `-U` to permit control over which installed software will be disabled for autostart.

Additional component names allowed for `-E` and `-D` options:
`ifs_fm_snmp`

`-C` – Outputs the list of supported component names.

`-V` – Outputs the version number of the software.

E.3 Additional Options

-r dir – Specify an alternate root directory. The default is `/`.

--without-depcheck – Disable check of OS dependencies.

--force – Force installation even if distributions do not match,
Use of this option can result in undefined behaviors

--answer keyword=*value* – Provides an answer to a question which might occur during the operation. Answers to questions which are not asked are ignored. Invalid answers will result in prompting for interactive installs or use of the default for non-interactive.

- Possible Questions:
  `UserQueries` – Permit non-root users to query the fabric default options retain existing configuration files supported component names:
  `fmtools ifs_fm`

`-v` – Verbose logging.

`-vv` – Outputs version.

§ §