Intel® Ethernet Fabric Suite FastFabric

User Guide

Rev. 1.7

March 2024

Doc. No.: 632487, Rev.: 1.7

Revision History

Date	Revision	Description
March 2024	1.7	Product 11.6.0.0 release - Changes to this document include: • Enhanced MPI Sample Applications run_* script infrastructure to simplify use and better integrate with environments using the SLURM job scheduler. See: MPI Sample Applications • Documented how to build mpi_apps for GPUs. See Building MPI Sample Applications. • Added description of ethbw. • Improved descriptions for ethextractbadlinks, ethextracterror, ethextractlink, ethextractperf, ethextractperf2, ethextractsellinks, ethextractstat, and ethextractstat2.
September 2023	1.6	Product 11.5.1.0 release - Changes to this document include: • Assorted grammatical, formatting and style improvements through the document.
May 2023	1.5	Product 11.5.0.0 release - Changes to this document include: • Refined Overview section.
March 2023	1.4	Product 11.4.1.0 release - Changes to this document include: • Added new tool dsa_setup
October 2022	1.3	Product 11.4.0.0 release - Changes to this document include: • Added fabric plane support on FastFabric TUI • Added Switches List Configuration Files • Updated ethfabricanalysis, ethallanalysis, ethfabricinfo, ethfindgood, ethlinkanalysis, ethreport and ethscpall • Added new tool ethshmcleanup • Updated OSU Micro-Benchmark version to 5.9
March 2022	1.2	Product 11.2.0.0 release - Changes to this document include: • Updated figures: Intel® EFS Host Fabric Software Stack, Intel® EFS Fabric and Software Components
July 2021	1.1	Product 11.1.0.0 release - Changes to this document include: • Renamed chassis to switch(es) to be consistent • Added PFC verification feature • Added multi-plane support • Updated MPI APPs build with CUDA support • Updated screen output with the latest version Intel® EFS
February 2021	1.0	Product 11.0.0.0 release - Initial public release.

Contents

Figures

Tables

Preface

This manual is part of the documentation set for the Intel® Ethernet Fabric Suite Fabric (Intel® EFS Fabric), which is an end-to-end solution consisting of Network Interface Cards (NICs), fabric management, and diagnostic tools.

The Intel® EFS Fabric delivers the next generation, High-Performance Computing (HPC) network solution that is designed to cost-effectively meet the growth, density, and reliability requirements of HPC and AI training clusters.

Intended Audience

The intended audience for the Intel® Ethernet Fabric Suite (Intel® EFS) document set is network administrators and other qualified personnel.

Intel® Ethernet Fabric Suite Documentation Library

Intel® Ethernet Fabric Suite publications are available at the following URL:

https://www.intel.com/content/www/us/en/support/articles/000088090/ethernet-products/intel-ethernet-software.html

Use the tasks listed in this table to find the corresponding Intel® Ethernet Fabric Suite document.

Task	Document Title	Description
Installing host software Installing NIC firmware	*Intel® Ethernet Fabric Suite Software Installation Guide*	Describes using a Text-based User Interface (TUI) to guide you through the installation process. You have the option of using command line interface (CLI) commands to perform the installation or install using the Linux distribution software.
Managing a fabric using FastFabric	*Intel® Ethernet Fabric Suite FastFabric User Guide*	Provides instructions for using the set of fabric management tools designed to simplify and optimize common fabric management tasks. The management tools consist of Text-based User Interface (TUI) menus and command line interface (CLI) commands.
Running MPI applications on Intel® EFS Running middleware that uses Intel® EFS	*Intel® Ethernet Fabric Suite Host Software User Guide*	Describes how to set up and administer the Network Interface Card (NIC) after the software has been installed and provides a reference for users working with Intel PSM3. Performance Scaled Messaging 3 (PSM3) is an Open Fabrics Interface (OFI, also called libfabric) provider which implements an optimized user-level communications protocol. The audience for

continued...

Task	Document Title	Description
		this document includes cluster administrators and those running or implementing Message-Passing Interface (MPI) programs.
Optimizing system performance	*Intel® Ethernet Fabric Performance Tuning Guide*	Describes BIOS settings and parameters that have been shown to ensure best performance, or make performance more consistent, on Intel® Ethernet Fabric Suite Software. If you are interested in benchmarking the performance of your system, these tips may help you obtain better performance.
Learning about new release features, open issues, and resolved issues for a particular release	*Intel® Ethernet Fabric Suite Software Release Notes*	

How to Search the Intel® Ethernet Fabric Suite Documentation Set

Many PDF readers, such as Adobe Reader and Foxit Reader, allow you to search across multiple PDFs in a folder.

Follow these steps:

1. Download and unzip all the Intel® Ethernet Fabric Suite PDFs into a single folder.
2. Open your PDF reader and use **CTRL-SHIFT-F** to open the Advanced Search window.
3. Select **All PDF documents in...**
4. Select **Browse for Location** in the dropdown menu and navigate to the folder containing the PDFs.
5. Enter the string you are looking for and click **Search**.

Use advanced features to further refine your search criteria. Refer to your PDF reader Help for details.

Documentation Conventions

The following conventions are standard for Intel® Ethernet Fabric Suite documentation:

- *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 to a figure, table, or section in this guide. Links to websites are also shown in blue. For example:

  See License Agreements for more information.

  For more information, visit www.intel.com.
- Text in **bold** font indicates user interface elements such as menu items, buttons, check boxes, key names, key strokes, or column headings. For example:

Click the **Start** button, point to **Programs**, point to **Accessories**, and then click **Command Prompt**.

Press **CTRL+P** and then press the **UP ARROW** key.

- Text in `Courier` font indicates a file name, directory path, or command line text. For example:

  Enter the following command: `sh ./install.bin`

- Text in *italics* indicates terms, emphasis, variables, or document titles. For example:

  Refer to *Intel® Ethernet Fabric Suite Software Installation Guide* for details.

  In this document, the term *chassis* refers to a managed switch.

Procedures and information may be marked with one of the following qualifications:

- **(Linux)** – Tasks are only applicable when Linux is being used.
- **(Host)** – Tasks are only applicable when Intel® Ethernet Host Software or Intel® Ethernet Fabric Suite is being used on the hosts.
- Tasks that are generally applicable to all environments are not marked.

Best Practices

- Intel recommends that users update to the latest versions of Intel® Ethernet Fabric Suite software to obtain the most recent functional and security updates.
- To improve security, the administrator should log out users and disable multi-user logins prior to performing provisioning and similar tasks.

License Agreements

This software is provided under one or more license agreements. Refer to the license agreement(s) provided with the software for specific detail. Do not install or use the software until you have carefully read and agree to the terms and conditions of the license agreement(s). By loading or using the software, you agree to the terms of the license agreement(s). If you do not wish to so agree, do not install or use the software.

Technical Support

Creating a technical support ticket for Intel® Ethernet Fabric Suite products is available 24 hours a day, 365 days a year. Contact Intel® Customer Support or visit https://www.intel.com/content/www/us/en/support.html for additional details.

1.0 Introduction

This manual provides instructions for using the Intel® Ethernet Fabric Suite FastFabric, a set of fabric management tools designed to simplify and optimize common fabric management tasks.

For details about the other documents for the Intel® Ethernet Fabric Suite product line, refer to Intel® Ethernet Fabric Suite Documentation Library on page 10 of this document.

The management tools consist of TUI menus and command line interface (CLI) commands. All of the functions that the TUI menus perform can also be performed using CLI commands. To aid in learning the commands, the TUI shows each CLI command as it executes it.

NOTE

This manual assumes that you have already installed the Intel® Ethernet Fabric Suite Software as prescribed in the *Intel® Ethernet Fabric Suite Software Installation Guide*

1.1 Documentation Organization

This manual is organized as follows:

- This **Introduction** provides an overview of this document and its structure.
- **Overview** provides an overview of the Intel® Ethernet Fabric Suite and FastFabric architecture and capabilities.
- **Getting Started** provides instructions and information for starting up and using the FastFabric TUI and CLI tools as well as an introduction to configuration files.
- **FastFabric TUI Menus** provide instructions for setting up, managing, and verifying hosts.
- **Descriptions of Command Line Tools** provides complete descriptions of each CLI tool and its parameters.
- **FastFabric Diagnostics Capabilities** provides information about the FastFabric features that help you diagnose fabric issues.
- **MPI Sample Applications** provides a variety of sample applications that can be used to perform basic tests and performance analysis.

2.0 Overview

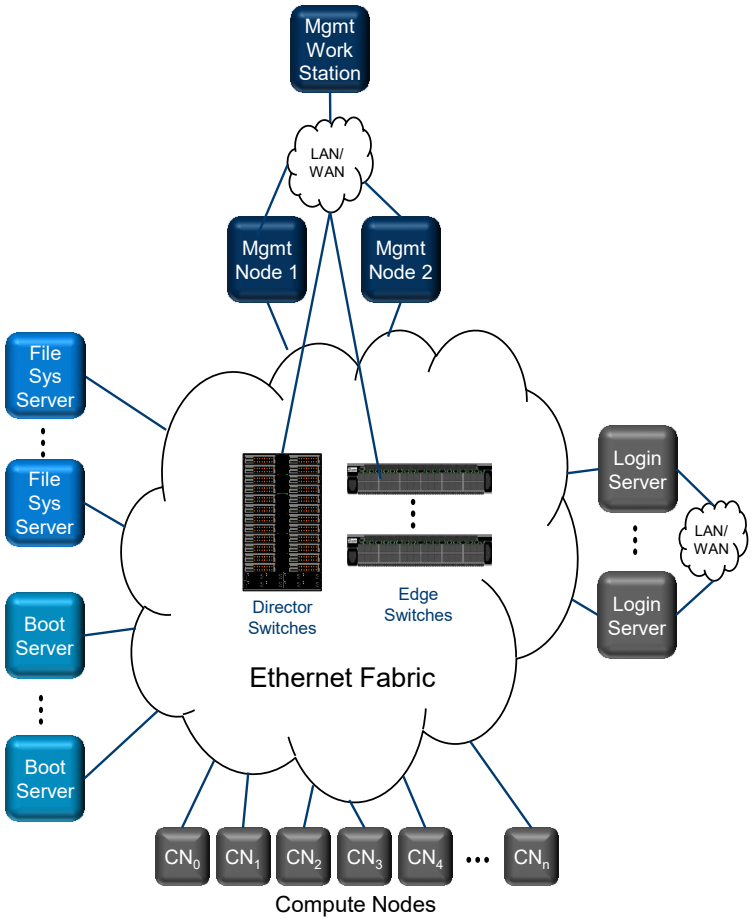
This section provides an overview of the Intel® Ethernet Fabric Suite and Intel® Ethernet Fabric Suite FastFabric.

2.1 Intel® Ethernet Fabric Suite Overview

The Intel® Ethernet Fabric Suite (Intel® EFS) interconnect fabric design enables a broad class of multiple node computational applications requiring scalable, tightly-coupled processing, memory, and storage resources. With open standard APIs developed by the OpenFabrics Alliance (OFA) Open Fabrics Interface (OFI) workgroup, NICs in the Intel® EFS family are optimized to provide the low latency, high bandwidth, and high message rate needed by High Performance Computing (HPC) and AI training applications.

The following figure shows a sample Intel® EFS-based fabric, consisting of different types of nodes and servers.

Figure 1. Intel® EFS Fabric



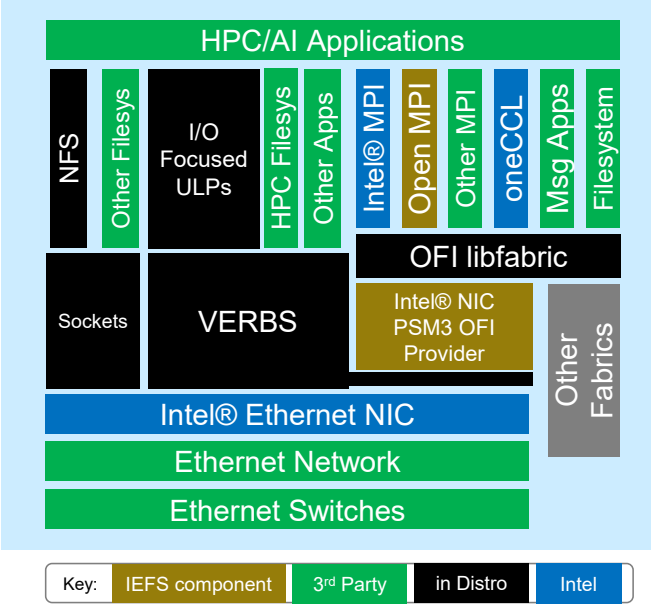
The software ecosystem is built around OFA software and includes three key APIs.

1. The OFA OFI represents a long-term direction for high-performance user-level and kernel-level network APIs.
2. OFA Verbs provides support for existing remote direct memory access (RDMA) applications.
3. Sockets are supported and permits many existing applications to immediately run on Intel® Ethernet Fabric Suite as well as provide TCP/IP features such as IP routing and network bonding.

Higher-level communication libraries, such as the Message Passing Interface (MPI), are layered on top of these low level OFA APIs. This permits existing HPC applications to immediately take advantage of advanced Intel® Ethernet Fabric Suite features.

Intel® Ethernet Fabric Suite combines the Network Interface Card (NIC), standard third-party Ethernet switches, and fabric management tools into an end-to-end solution. The host fabric software stack is shown in the following figure.

Figure 2. Intel® EFS Host Fabric Software Stack



2.1.1 Network Interface Card

Each host is connected to the fabric through a Network Interface Card (NIC). The NIC translates instructions between the host processor and the fabric. It includes the logic necessary to implement the physical and link layers of the fabric architecture, so that a node can attach to a fabric and send and receive packets to other servers or devices. NICs also include specialized logic for executing and accelerating upper layer protocols, such as RDMA transport layers.

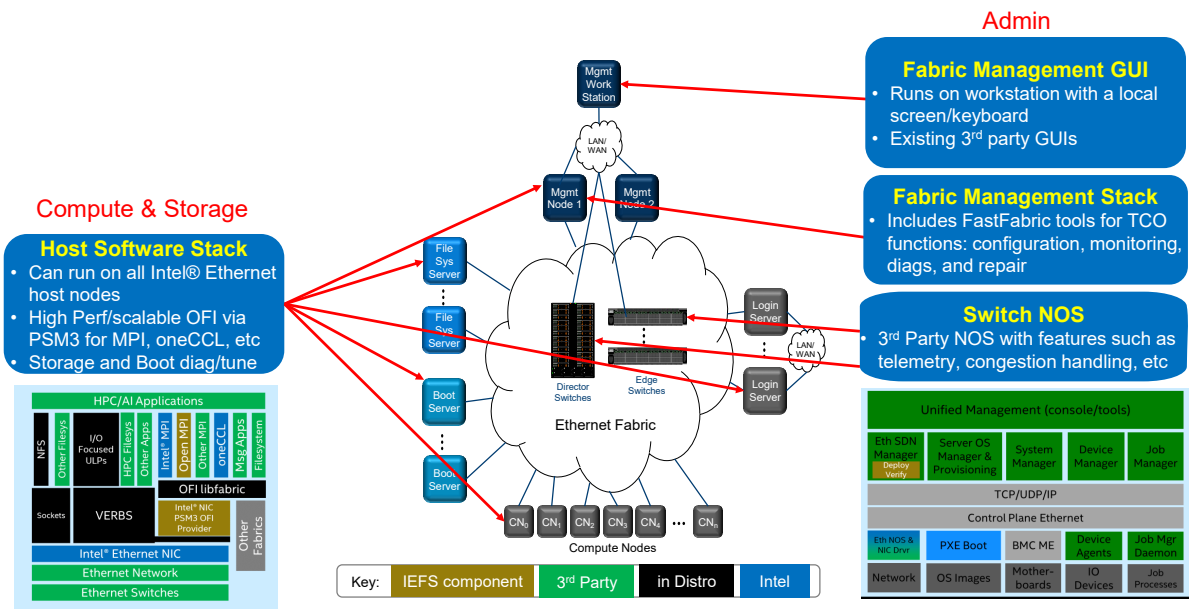
2.2 Intel® Ethernet Fabric Suite Software Overview

For software applications, Intel® EFS maintains consistency and compatibility with existing standard APIs through the open source OpenFabrics Alliance (OFA) software stack on Linux distribution releases.

Software Components

The key software components and their usage models are shown in the following figure and described in the following table.

Figure 3. Intel® EFS Fabric and Software Components



Software Component Descriptions
Switch Network Operating System (NOS) Intel® EFS supports a variety of third-party NOS solutions on standard Ethernet switches. Each of these switches may include features such as: • An embedded processor that runs switch management and control functions. • System management capabilities, including signal integrity, thermal monitoring, and voltage monitoring, among others. • Ethernet port access using command line interface (CLI) or graphical user interface (GUI).
Host Software Stack • Runs on all Intel® EFS-connected host nodes and supports compute, management, and I/O nodes. • Provides a rich set of APIs including OFI, sockets, and OFA verbs. • Provides high performance, highly scalable MPI implementation through the Intel PSM3 OFI (also known as libfabric) provider, and multiple MPI middlewares. • Includes Boot over Fabric mechanism for configuring a server to boot over the Intel® Ethernet Fabric using the NIC Unified Extensible Firmware Interface (UEFI) firmware. User documents: • *Intel® Ethernet Fabric Suite Host Software User Guide* • *Intel® Ethernet Fabric Performance Tuning Guide*
Fabric Management Stack Intel® EFS supports a variety of third-party Ethernet management solutions including popular Software Defined Networking (SDN) stacks. As part of the management solution, the Intel® EFS FastFabric tools are provided to aid deployment verification, fabric tuning, and diagnosis. • Runs on Intel® EFS-connected management nodes. • Includes a toolkit for configuration, monitoring, diagnostics, and repair. User documents: • *Intel® Ethernet Fabric Suite FastFabric User Guide*

2.3 FastFabric Overview

Intel® Ethernet Fabric Suite FastFabric is a set of fabric management tools designed to simplify and optimize common fabric management tasks. FastFabric includes the following capabilities:

- Monitoring and diagnostic tools
- Fabric deployment and verification
- Host management

FastFabric consists of a hierarchy of commands and tools. In order to simplify learning and use, these tools all have similar command line arguments. Many of the FastFabric tools are designed to be easily extended via scripting or exporting data into other formats, such as spreadsheets.

The higher level tools allow you to focus on the names assigned to devices, and avoid the need to figure out IfIDs or remember IfAddr for basic operations. As such, Intel recommends that you establish a naming convention for the cluster, and assign names to all the hosts and switches in the cluster.

2.3.1 FastFabric Architecture

FastFabric is typically installed on one or more Fabric Management Nodes. The Fabric Management Node must be connected to the rest of the cluster through the Intel® Ethernet Fabric and a management network. The management network is used for FastFabric host setup and administration tasks. It may also be used for other aspects of server administration or operation.

Depending on cluster size and design, the Fabric Management node may also be used as the master node for starting Message Passing Interface (MPI) jobs. It may also be used to run other management software.

If remote access to FastFabric is desired, set up remote access to the Fabric Management Node using the ssh, Telnet, X-Windows, VNC, or any other mechanism that will allow the remote user to access a Linux Command Line shell. Typically, FastFabric is used only by cluster administrators.

2.3.1.1 How FastFabric Works

FastFabric consists of a variety of tools to administer hosts. Depending on the tool, the method of accessing and administering the target devices may differ.

The following table describes the access methods that FastFabric uses.

Table 1. FastFabric Methods

Method	Examples
SNMP query	Fabric performance, statistics, and congestion monitoring. Fabric topology reports, fabric error and link speed analysis, etc.
Log in through a management network	Host setup and installation, etc.
MPI job startup	Verify MPI performance, running sample MPI benchmarks, host-to-switch cable test.

Tools that log into other hosts will do so in a password-less manner using ssh. Tools that log into managed chassis can also use ssh. These approaches permit the tools to operate with minimal user interaction, and for this reason reduce the time to perform operations against many hosts .

2.3.2 FastFabric Capabilities

2.3.2.1 FastFabric Command Hierarchy

FastFabric provides numerous, powerful commands. These commands can be best understood as a hierarchy of capabilities permitting operations at high, mid, and low levels.

2.3.2.1.1 Monitoring and Diagnostics

At the highest level, FastFabric provides an interactive Text-based User Interface (TUI), called `ethfastfabric`. The TUI provides an easy and efficient way to perform fabric deployment and verification, and diagnosis of typical fabrics. The TUI is structured in the typical sequence of operations for fabric verification. All of the functions that the TUI performs are also available using command line interface (CLI) commands. To aid in learning the commands, the TUI shows each CLI command as it executes it.

Other high-level tools can provide an initial view of fabric status and health. These include the tools to verify cluster status as compared to a previous baseline (`ethallanalysis` and its sub-tools: `ethlinkanalysis`, `ethfabricanalysis`)

When analyzing the fabric at a mid-tier of information, the next tier of tools include: `ethfabricinfo`, `ethreport`, `ethextractbadlinks`, `ethextractlink`, `ethextractsellinks`, and `ethextractstat2`. These tools provide powerful ways to query the fabric. The `ethextract*` family of tools are all scripts that take advantage of `ethreport` to generate delimited files that can be easily parsed or exported into spreadsheets for offline analysis. These scripts can also be good samples for the creation of site-specific sysadmin scripts.

At the next level of lower analysis, there are additional tools. These provide direct access to more of the raw fabric information, such as port counters, interface addresses, and other configured parameters. Tools in this tier include: `ethshowallports`, `ethextractifids`, `ethextracterror`, `ethextractperf`, `ethextractstat`. Many of these tools are scripts that are also built on top of `ethreport`. `ethreport` is a foundational tool in FastFabric that provides a rich set of fabric analysis capabilities, and can provide both high level and very detailed output.

2.3.2.1.2 Benchmark and Stress Tests

FastFabric includes a number of benchmarks and stress tests. These can be found in `/usr/src/eth/mpi_apps` and `/usr/mpi/*/*/tests`. The `ethcabletest` tool also provides a simple way to create high stress on all links in the fabric to aid in the verification of fabric stability.

In addition, other existing Intel® Ethernet Fabric Suite benchmarks and test programs may also be used to exercise the `libfabric` and verbs interfaces.

2.3.2.2 Host Management

For hosts, `ethhostadmin` provides typical control and query functions to manage host software and configuration. `ethfindgood` and `ethverifyhosts` can provide analysis of the host status. In addition, `ethpingall`, `ethcmdall`, `ethscpall`, `ethdownloadall`, `ethuploadall`, and `ethsetupssh` are tools that are included to perform basic ssh and scp operations against the hosts.

2.3.2.3 Topology Analysis

FastFabric includes a rich set of topology analysis and verification capabilities. This can start with a pre-assembly description of the cluster design, from which `ethxlattopology` can generate a `topology.xml` file for use by FastFabric.

`ethreport` has a number of reports for verifying the topology (`-o verify*`). In addition, reports such as `ethreport -o links`, `ethextractlink`, and `ethextractsellinks` can provide an in-depth view of the fabric connectivity and design.

2.3.2.4 Focused Fabric Feature Analysis

Tools and reports are available to provide in-depth analysis of various fabric features.

Link quality, signal integrity, security errors, and other issues can be analyzed using the following:

- `ethfastfabric` TUI
- `ethallanalysis`
- `ethextractbadlinks`
- `ethreport` (such as `-o errors`, `-o slow*`, and `-o mis*` reports)
- `ethshowallports`

To view or analyze link speed, the following commands can be used:

- `ethreport -o slowlinks`
- `ethfabricanalysis` (uses `ethreport -o slowlinks`)
- `ethextracterror` (uses `ethreport -o comps`, shows main error counters)
- `ethextractperf` (uses `ethreport -o comps`, shows per port counters)
- `ethlinkanalysis slowlinks`

2.3.2.5 Scripting and Integration Enablement

Various additional tools can facilitate extending FastFabric, or integrating it with other tools. Among these are the XML processing tools (`ethxmlextract`, `ethxmlfilter`, and `ethxmlindent`), which can permit the XML output formats from `ethreport` and/or the `mgt_config.xml` file itself to be easily parsed and analyzed in other scripts. The `ethextract*` scrips can provide samples of how to effectively use these tools.

2.3.2.6 Scripting on Top of FastFabric

Intel® Ethernet Fabric Suite FastFabric was designed to make the scripting of OEM or site-specific tools easy to use. However, to ensure forward compatibility, scripts should be created using the CLI tools and arguments.

A number of the tools, such as the `eth*analysis` set of tools, are designed for easy use through exit code checks. These tools can easily be scripted to be run, and then, on bad exit codes, to issue emails or other forms of alerts to system administrators. Such mechanisms can be scheduled for regular execution by way of cron jobs. The file that is created by these tools can then be analyzed by the system administrators.

`ethreport` is a powerhouse tool that provides a wide range of fabric data-gathering and analysis capabilities. The best way to script with this tool is to take advantage of its `-x` option to output XML. That output can then be easily parsed by `ethxmlextract` to extract sets of fields into delimited formats that can be easily parsed by scripts, or exported to external tools such as spreadsheets. The `eth*extract` set of scripts are built on top of `ethreport -x` and `ethxmlextract`. These scripts can provide a great starting point by copying them and then creating new variations to meet your unique needs.

Intel recommends that you DO NOT create scripts that attempt to directly parse `ethreport -o` snapshot output. This format cannot be guaranteed to be forward-compatible with future FastFabric software releases. Most of the information in an `ethreport -o` snapshot is also available in a forward-compatible format via `ethreport -x -o comps -d 10 -s`. The remainder can be found in other `ethreport` output by using different options.

Intel also recommends that you DO NOT create scripts that attempt to parse the human-readable output formats produced by the tools. Intel reserves the right to refine these formats in future FastFabric software releases, and therefore, these formats cannot be guaranteed to be forward-compatible.

2.3.2.7 Customer Support Data Gathering

Detailed information about the current fabric status and configuration can be quickly obtained using `ethcapture` for a single node (refer to *Intel® Ethernet Fabric Suite Host Software User Guide*), or `ethcaptureall` for multiple nodes, to aid customer support.

2.3.2.8 Other Tools and Capabilities

In addition, a number of the non-InfiniBand specific OpenFabrics Alliance (OFA) tools will continue to function on an Intel® Ethernet Fabric, and can provide additional information. Among these are `ibv_devinfo` (note that MTU will not correctly report MTUs beyond 4K), `ibstat`, `ibsrpdm`, and `ibv_devices`.

3.0 Getting Started

This section provides instructions and information for getting started with the Intel® Ethernet Fabric Suite FastFabric tools.

3.1 Important Note on First-Time Installations

This user guide is not an installation guide.

If you are installing and configuring the fabric for the first time, you should refer to the *Intel® Ethernet Fabric Suite Software Installation Guide*.

3.2 Working with TUI Menus

One method for working with the FastFabric toolset is through the TUI menus. This method provides a more guided, task-oriented approach for using the tools, prompting the user for information and values.

3.2.1 Starting Up the Tools

NOTE

To run the Intel® Ethernet Fabric Suite FastFabric tools described in this manual, you must have root privileges.

3.2.1.1 Accessing the Intel FastFabric TUI Tool Menu

The Intel FastFabric TUI Tool menu allows you to configure and manage the Intel® Ethernet Fabric.

Using the `ethfastfabric` Command

To start up the Intel FastFabric TUI Tool menu from the command prompt, perform the following steps:

1. Log in to the server as root.
2. At the command prompt, enter **`ethfastfabric`**.

   The Intel FastFabric TUI Tool menu is displayed.

```
Intel Ethernet FastFabric Tools
Version: X.X.X.X.X

   1) Host Setup
   2) Host Verification/Admin

   X) Exit (or Q)
```

From the Intel Ethernet Software Menu

To start up the Intel Ethernet FastFabric Tools menu from the Intel Ethernet Software main menu, perform the following steps:

1. Log in to the server as root.
2. At the command prompt, enter **iefsconfig**.

   The Intel Ethernet [version] Software main menu is displayed.

```
Intel Ethernet X.X.X.X.X Software

   1) Show Installed Software
   2) Reconfigure Eth RDMA
   3) Reconfigure Driver Autostart
   4) Generate Supporting Information for Problem Report
   5) FastFabric (Host/Admin)
   6) Uninstall Software

   X) Exit
```

3. At the cursor, type **5**.

   The Intel Ethernet FastFabric Tools menu is displayed.

```
Intel Ethernet FastFabric Tools
Version: X.X.X.X.X

   1) Host Setup
   2) Host Verification/Admin

   X) Exit (or Q)
```

3.2.2 Intel Ethernet FastFabric Tools Overview

The Intel Ethernet FastFabric Tools allows you to perform common fabric management tasks including setting up and managing the hosts.

The following is an example of the Intel Ethernet FastFabric Tools main menu.

```
Intel Ethernet FastFabric Tools
Version: X.X.X.X.X

   1) Host Setup
   2) Host Verification/Admin

   X) Exit (or Q)
```

3.2.3 How to Use the FastFabric TUI

The FastFabric TUI menus are set up for ease of use. The submenus are designed to present operations in the order they would typically be used during an installation.

NOTE

All FastFabric TUI menu, alpha-based options are case-insensitive.

Selecting Menu Items and Performing Operations

1. From the Intel Ethernet FastFabric Tools main menu, select the target menu item (1-2).

```
Intel Ethernet FastFabric Tools
Version: X.X.X.X.X

   1) Host Setup
   2) Host Verification/Admin

   X) Exit (or Q)
```

   The target menu is displayed as shown in the following example:

```
FastFabric Ethernet Host Setup Menu
Host File: /etc/eth-tools/hosts
Setup:
0) Edit Management Config File                [ Skip  ]
1) Edit FF Config and Select/Edit Host File   [ Skip  ]
2) Verify Hosts Pingable                      [ Skip  ]
3) Set Up Password-Less SSH/SCP               [ Skip  ]
4) Copy /etc/hosts to All Hosts               [ Skip  ]
5) Show uname -a for All Hosts                [ Skip  ]
6) Install/Upgrade Intel Ethernet Software    [ Skip  ]
7) Configure SNMP                             [ Skip  ]
8) Build Test Apps and Copy to Hosts          [ Skip  ]
9) Reboot Hosts                               [ Skip  ]
Admin:
a) Refresh SSH Known Hosts                    [ Skip  ]
b) Rebuild MPI Library and Tools              [ Skip  ]
c) Run a Command on All Hosts                 [ Skip  ]
d) Copy a File to All Hosts                   [ Skip  ]
Review:
e) View ethhostadmin Result Files             [ Skip  ]

P) Perform the Selected Actions N) Select None
X) Return to Previous Menu (or ESC or Q)
```

2. Type the key corresponding to the target menu item (0-9, a-e) to toggle the Skip/Perform selection.

   More than one item may be selected.

3. Type P to perform the operations that were selected.

   NOTES

   - If more than one menu item is selected, the operations are performed in the order shown in the menu. This is the typical order desired during fabric setup.
   - If you want to perform operations in a different order, you must select the first target menu item, type P to perform the operation, then repeat this process for the next menu item operation to be performed, and so on.

4. Type N to clear all selected items.
5. Type X or press Esc or Q to exit this menu and return to the Main Menu.

Aborting Operations

While multiple menu items are performing, you have an opportunity to abort individual operations as they come up. After each operation completes and before the next operation begins, you are prompted as shown:

```
Hit any key to continue...
```

- Press `Esc` or `Q` to stop the sequence of operations return to the previous menu.

  Any unperformed operations are still highlighted in the menu. To complete the selected operations, type `P`.

- Press any other key to perform the next selected menu item being performed.

  This prompt is also shown after the last selected item completes, providing an opportunity to review the results before the screen is cleared to display the menu.

Submenu Configuration Files

On each FastFabric submenu, item `0` allows reviewing and editing (using the editor selected by the EDITOR environment variable) of the `mgt_config.xml` file to specify planes in a fabric and SNMP query parameters. Item `1` permits a different file to be selected and edited. It also permits reviewing and editing of the `ethfastfabric.conf` file. The `ethfastfabric.conf` file guides the overall configuration of FastFabric and describes cluster-specific attributes of how FastFabric operates.

At the top of each FastFabric submenu screen beneath the title, the directory and configuration file containing the components on which to operate are shown.

In the following example, the configuration file is noted in bold.

```
FastFabric Ethernet Host Setup Menu
Host File: /etc/eth-tools/hosts
Setup:
0) Edit Management Config File               [ Skip  ]
1) Edit FF Config and Select/Edit Host File  [ Skip  ]
2) Verify Hosts Pingable                     [ Skip  ]
```

NOTE

During the execution of each menu selection, the actual FastFabric command line tool being used is shown. This can be used as an educational aid to learn the command line tools.

The following example snippet shows how the CLI is displayed in the TUI execution.

```
Performing Host Setup: Verify Hosts Pingable
Would you like to verify hosts are ssh-able? [n]:y
Executing: /usr/sbin/ethfindgood -A -f /etc/eth-tools/hostes
```

Related Links

Configuration Files for FastFabric on page 37

3.3 Working with CLI Commands

Another method for working with the FastFabric toolset is through CLI commands. This method requires more advanced knowledge of FastFabric, and provides more control of the tools through individual parameters.

3.3.1 Common Tool Options

The following table lists the common CLI options that are applicable to most of the tools.

Table 2. Common Tool Options

Command	Description
`-?`	Displays basic usage information for any of the commands. An invalid option also displays this information.
`--help`	Displays complete usage information for most of the commands.
`-p`	Runs the operation/command in parallel. This means the operation is performed simultaneously on batches of FF_MAX_PARALLEL hosts. (Default = 1000.) This option allows the overall time of an operation to be much lower. However, a side effect is that any output from the command is bursty and intermingled. Therefore, this option should be used for commands where there is no output or the output is of limited interest. For some commands (such as `ethscpall`), this performs the operation in a quiet mode to limit output. If you want to change the number of parallel operations, export `FF_MAX_PARALLEL=#` where `#` is the new number (such as 500). For more advanced operations (such as `ethhostadmin`), parallel operation is the default mode. Parallel operation can also be disabled by setting `FF_MAX_PARALLEL` to `1`.
`-S`	Prompts for password for root on host. The password is prompted for once, and the same password is then used to log in during the operation. For hosts, this option is only applicable to `ethsetupssh`.
`-h`	Selects which local NIC to use.
`-v`	Produces verbose output.

3.3.2 Selection of Devices

This section describes how you choose devices for CLI commands. In general, you can select a number of devices through list files or explicitly identify devices by their names or formats within the command.

3.3.2.1 Selection of Hosts

To perform operations against a set of hosts, you can specify the hosts on which to operate using one of the following methods:

- On the command line, using the `-h` option.
- Using the environment variable `HOSTS` to specify a space-separated list of hosts. Useful when multiple commands are performed against the same small set of hosts.
- Using the `-f` option or the `HOSTS_FILE` environment variable to specify a file containing the set of hosts. Useful for groups of hosts that are used often. The file is located here: `/etc/eth-tools/hosts` by default. The file must list all hosts in the cluster except the host running the FastFabric toolset itself.

Within the tools, the options are considered in the following order:

1. `-h` option
2. `HOSTS` environment variable
3. `-f` option
4. `HOSTS_FILE` environment variable
5. `/etc/eth-tools/hosts` file

For example, if the `-h` option is used and the `HOSTS_FILE` environment variable is also exported, the command operates only on hosts specified using the `-h` option.

3.3.2.1.1 Host List Files

You can use the `-f` option to provide the name of a file containing the list of hosts on which to operate. The default location is `/etc/eth-tools/hosts`.

It may be useful to create multiple files in `/etc/eth-tools` representing different subsets of the fabric. For example:

- `/etc/eth-tools/hosts-mpi` – List of MPI hosts
- `/etc/eth-tools/hosts-fs` – List of file server hosts
- `/etc/eth-tools/hosts` – List of all hosts except for the FastFabric toolset node
- `/etc/eth-tools/allhosts` – List of all hosts including the FastFabric toolset node

Host List File Format

Sample host list file:

```
# this is a comment
192.168.0.4  # host identified by IP address
n001  # host identified by resolvable TCP/IP name
n001:eth1,eth2  # host associated with network interfaces
include /etc/eth-tools/hosts-mpi  # included file
```

Each line of the host list file may specify a single host, a comment, or another host list file to include. You can augment a hostname with a list of network interfaces to specify the interfaces that are part of the fabric. If no network interfaces are defined for a host, all available interfaces on the host will be considered as part of the fabric.

Hosts are specified by resolvable TCP/IP hostnames (without appended domain names). Typically, management network hostnames are specified. However, if desired, IP addresses may be used to accelerate large file transfers and other operations.

Files to be included may be specified using an `include` directive followed by a file name. File names specified should generally be absolute pathnames. If relative pathnames are used, they are searched for in the current directory first, then `/etc/eth-tools`. To avoid cyclic includes, only one level of include is supported.

Comments may be placed on any line by using a `#` to precede the comment. On lines with hosts or include directives, the `#` must be white space-separated from any preceding hostname, IP address, or included file name.

3.3.2.1.2 Explicit Host Names

When hosts are explicitly specified using the -h option or the HOSTS environment variable, a space-separated list of host names (or IP addresses) may be supplied. For example: -h '*host1 host2 host3*'. A hostname can also be augmented with interface names, such as -h '*host1:eth2 host2:eth2 host3:eth2*'

3.3.2.2 Selection of Switches

To perform operations against a set of switches, you can specify the switches on which to operate using one of the following methods:

- On the command line, using the -H option.
- Using the environment variable SWITCHES to specify a space-separated list of switches. Useful when multiple commands are performed against the same small set of switches.
- Using the -F option or the SWITCHES_FILE environment variable to specify a file containing the set of switches. Useful for groups of switches that will be used often. The file is located here: /etc/eth-tools/switches by default. The file must list all switches in the cluster.

Within the tools, the options are considered in the following order:

1. -H option
2. SWITCHES environment variable
3. -F option
4. SWITCHES_FILE environment variable
5. /etc/eth-tools/switches file

For example, if the -H option is used and the SWITCHES_FILE environment variable is also exported, the command operates only on switches specified by the -H option.

3.3.2.2.1 Switch List Files

You can use the -F option to provide the name of a file containing the list of switches on which to operate. The default is /etc/eth-tools/switches.

It may be useful to create multiple files in /etc/eth-tools representing different subsets of the fabric. For example:

- /etc/eth-tools/switches-core - List of core switches
- /etc/eth-tools/switches-edge - List of edge switches
- /etc/eth-tools/switches - List of all switches

If a relative path is specified for the -F option, the current directory is checked first, followed by /etc/eth-tools/.

Switch List File Format

Sample switches file:

```
# this is a comment
192.168.0.5   # switch IP address
edge1   # switch resolvable TCP/IP name
include /etc/eth-tools/switches-core  # included file
```

Each line of the switch list file may specify a single switch, a comment, or another switch list file to include.

A switch may be specified by switch management network IP address or a resolvable TCP/IP name. Typically, names are used for readability.

Files to be included may be specified using an `include` directive followed by a file name. File names specified should be absolute path names. If relative path names are used, they are searched for in the current directory first, then `/etc/eth-tools`. To avoid cyclic includes, only one level of include supported.

Comments may be placed on any line using a `#` to precede the comment. On lines with switch or `include` directives, the `#` must be white space-separated from any preceding name, IP address, or included file name.

3.3.2.2.2 Explicit Switch Names

When switches are explicitly specified using the `-H` option or the `SWITCHES` environment variable, a space-separated list of names (or IP addresses) may be supplied. For example: `-H` *`switch1 switch2 switch3`* .

3.4 Sample Files

This section describes the files that are installed in the `/usr/share/eth-tools/samples` directory, including the `ethgentopology` sample script.

3.4.1 List of Files

This section describes the files that are installed in the `/usr/share/eth-tools/samples` directory.

3.4.1.1 Configuration and Control Files

Files used by commands that analyze the fabric and perform multi-step initialization and verification operations.

- `allhosts-sample`: All hosts in fabric, including management nodes.

  See `ethhostadmin`.

- `hosts-sample`: All hosts in the fabric.

  See `ethhostadmin`.

3.4.1.2 Topology Files

Files related to topology:

- `README.topology`
- `README.xlat_topology`
- `ethgentopology`: Script to generate topology file.
- `ethtopology_links.txt`: Text CSV values for LinkSummary information.
- `ethtopology_NICs.txt`: Text CSV values for NIC Nodes information.
- `ethtopology_SWs.txt`: Text CSV values for Switch Nodes information.
- `linksum_swd06.csv`, `linksum_swd24.csv`: Sample CSV configurations.

  See `README.xlat_topology` for explanation.
- `detailed_topology.xlsx`, `minimal_topology.xlsx`: Topology files in spreadsheet format.
- `ethmon.conf-sample`, `ethmon.si.conf-sample`: Port counter threshold files for use with `ethreport`.

3.4.1.2.1 ethgentopology

Provides a simple sample of how to generate the topology XML file used for topology verification. If you want to integrate topology XML file generation into your cluster design process, you can create your own script to take information available in other formats and tools and produce the topology XML file directly. The alternative is to use `ethxlattopology` and have tools generate the input files it expects.

This tool uses CSV input files `ethtopology_links.txt`, `ethtopology_NICs.txt`, and `ethtopology_SWs.txt` to generate LinkSummary, Node NICs, and Node SWs information, respectively. These files are samples of what might be produced as part of translating a user-custom file format into temporary intermediate CSV files.

LinkSummary information includes Link, Cable, and Port information. Note that `ethgentopology` (not `ethxmlgenerate`) generates the XML version string as well as the `<Report>` and `<LinkSummary>` lines. Also note that the indent level is at the default value of zero (0). The portions of the script that call `ethxmlgenerate` follow:

```
ethxmlgenerate -X /usr/share/eth-tools/samples/ethtopology_1.txt -d \; -h Link \
-g Rate -g MTU -g Internal -g LinkDetails -h Cable -g CableLength -g CableLabel \
-g CableDetails -e Cable -h Port -g IfAddr -g PortNum -g PortId -g NodeDesc \
-g MgmtIfAddr -g NodeType -g PortDetails -e Port  -h Port -g IfAddr -g PortNum \
-g PortId -g NodeDesc -g MgmtIfAddr -g NodeType -g PortDetails -e Port -e Link
```

```
ethxmlgenerate -X /usr/share/eth-tools/samples/ethtopology_2.txt -d \; \
-h Node -g IfAddr -g NodeDesc -g NodeDetails -e Node
```

Syntax

```
/usr/share/eth-tools/samples/ethgentopology [--help] [plane]
```

NOTE

You must use the full path to access this command.

Options

No option Produces sample output. See Example.

--help Produces full help text.

plane Plane name. Default is 'plane'.

ethtopology_links.txt

This file can be found in `/usr/share/eth-tools/samples/`. For brevity, this sample shows only two links. The second link shows an example of omitting some information. In the second line, the MTU, LinkDetails, and other fields are not present, which is indicated by an empty value for the field (no entry between the semicolon delimiters).

NOTE

The following example exceeds the available width of the page. For readability, a blank line is shown between lines to make it clear where the line ends. In an actual link file, no blank lines are used.

```
25g;2048;0;IO Server Link;11m;S4567;cable model
456;0x0002c9020020e004;1;20e004,bender-eth2;0x0002c9020020e004;NIC;Some info
about port;0x0011750007000df6;7;Eth7;Switch 1234 Leaf 4;;SW;
```

```
25g;;0;;;;;0x0002c9020025a678;1;25a678;mindy2-
eth2;;NIC;;0x0011750007000e6d;4;Eth4;Switch 2345 Leaf 5;;SW;
```

ethtopology_NICs.txt

This file can be found in `/usr/share/eth-tools/samples/`. For brevity, this sample shows only two nodes.

```
0x0002c9020020e004;bender-eth2;More details about node
0x0002c9020025a678;mindy2-eth2;Node details
```

ethtopology_SWs.txt

This file can be found in `/usr/share/eth-tools/samples/`. For brevity, this sample shows only two nodes.

```
0x0011750007000df6;Switch 1234 Leaf 4;
0x0011750007000e6d;Switch 2345 Leaf 5;
```

Example

When run against the supplied topology input files, `/usr/share/eth-tools/samples/ethgentopology` produces:

```
<?xml version="1.0" encoding="utf-8" ?>
<Report plane="plane">
<LinkSummary>
<Link>
<Rate>25g</Rate>
<MTU>2048</MTU>
<Internal>0</Internal>
<LinkDetails>IO Server Link</LinkDetails>
<Cable>
<CableLength>11m</CableLength>
<CableLabel>S4567</CableLabel>
<CableDetails>cable model 456</CableDetails>
</Cable>
<Port>
<IfAddr>0x001175010020e004</IfAddr>
<PortNum>1</PortNum>
<PortId>20e004</PortId>
<NodeDesc>bender-eth2</NodeDesc>
<MgmtIfAddr>0x001175010020e004</MgmtIfAddr>
<NodeType>NIC</NodeType>
<PortDetails>Some info about port</PortDetails>
</Port>
<Port>
<IfAddr>0x0011750107000df6</IfAddr>
<PortNum>7</PortNum>
<PortId>Eth7</PortId>
<NodeDesc>Switch 1234 Leaf 4</NodeDesc>
<NodeType>SW</NodeType>
</Port>
</Link>
<Link>
<Rate>25g</Rate>
<Internal>0</Internal>
<Cable>
</Cable>
<Port>
<IfAddr>0x001175010025a678</IfAddr>
<PortNum>1</PortNum>
<PortId>25a678</PortId>
<NodeDesc>mindy2-eth2</NodeDesc>
<NodeType>NIC</NodeType>
</Port>
<Port>
<IfAddr>0x0011750107000e6d</IfAddr>
<PortNum>4</PortNum>
<PortId>Eth4</PortId>
<NodeDesc>Switch 2345 Leaf 5</NodeDesc>
<NodeType>SW</NodeType>
</Port>
</Link>
</LinkSummary>
<Nodes>
<NICs>
<Node>
<IfAddr>0x0002c9020020e004</IfAddr>
<NodeDesc>bender-eth2</NodeDesc>
<NodeDetails>More details about node</NodeDetails>
</Node>
<Node>
<IfAddr>0x0002c9020025a678</IfAddr>
<NodeDesc>mindy2-eth2</NodeDesc>
<NodeDetails>Node details</NodeDetails>
</Node>
</NICs>
```

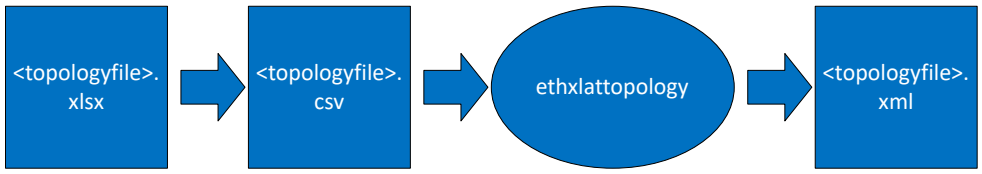
```
<Switches>
<Node>
<IfAddr>0x0011750107000df6</IfAddr>
<NodeDesc>Switch 1234 Leaf 4</NodeDesc>
</Node>
<Node>
<IfAddr>0x0011750107000e6d</IfAddr>
<NodeDesc>Switch 2345 Leaf 5</NodeDesc>
</Node>
</Switches>
</Nodes>
</Report>
```

3.4.1.2.2 Sample Topology Spreadsheet Overview

This section describes two sample topology spreadsheet files: `detailed_topology.xlsx` and `minimal_topology.xlsx`. Both files are installed in the `/usr/share/eth-tools/samples` directory. In this document, `<topologyfile>` indicates the file you customize for your cluster.

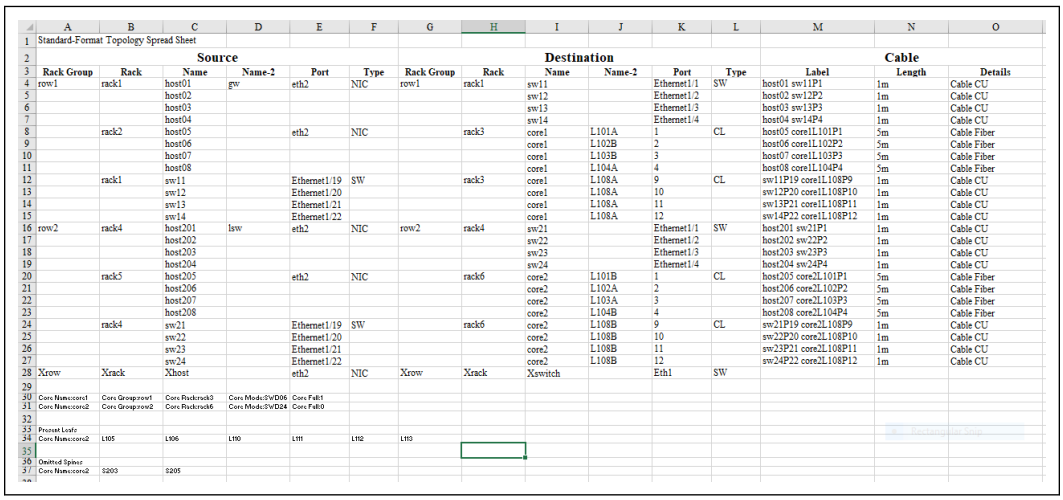
The following figure shows the multi-step process to generate the XML file required by topology verification tools, such as `ethreport`. First, select and edit one of the sample topology XLSX files, save the edited information in CSV format, and run the `ethxlattopology` script, which produces the `<topologyfile>.xml` file.

Figure 4. Topology Workflow



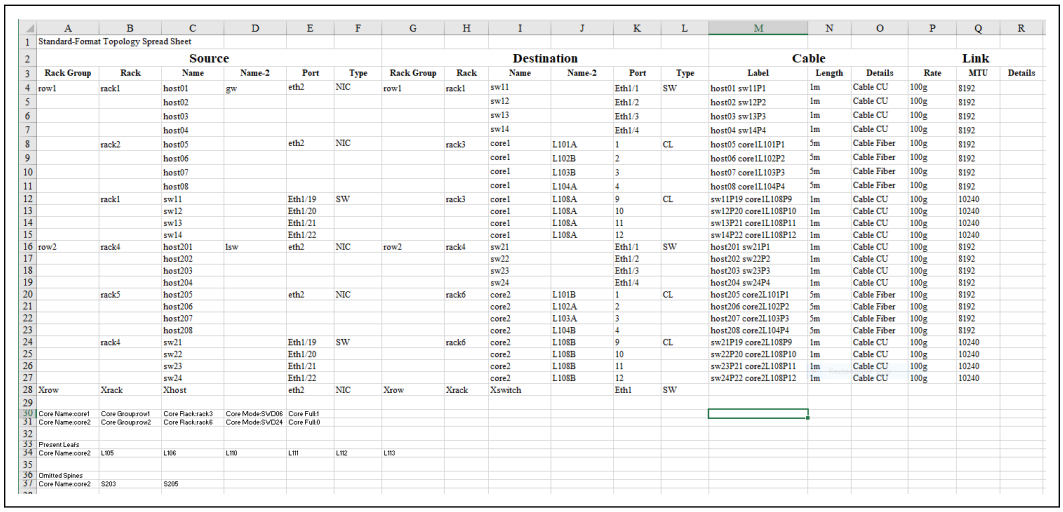
The two sample files, `detailed_topology.xlsx` and `minimal_topology.xlsx`, provide a standard format for representing each external link in a cluster. Each row in the `minimal_topology.xlsx` spreadsheet depicts one link containing **Source**, **Destination**, and **Cable** fields. The `detailed_topology.xlsx` spreadsheet contains an additional **Link** field providing link-specific details. In both spreadsheets, the cells cannot contain commas.

Figure 5. minimal_topology.xlsx Example



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Standard-Format Topology Spread Sheet														
2			Source						Destination					Cable	
3	Rack Group	Rack	Name	Name-2	Port	Type	Rack Group	Rack	Name	Name-2	Port	Type	Label	Length	Details
4	row1	rack1	host01	gw	eth2	NIC	row1	rack1	sw11		Ethernet1/1	SW	host01 sw11P1	1m	Cable CU
5			host02						sw12		Ethernet1/2		host02 sw12P2	1m	Cable CU
6			host03						sw13		Ethernet1/3		host03 sw13P3	1m	Cable CU
7			host04						sw14		Ethernet1/4		host04 sw14P4	1m	Cable CU
8		rack2	host05		eth2	NIC		rack3	core1	L101A	1	CL	host05 core1L101P1	5m	Cable Fiber
9			host06						core1	L102B	2		host06 core1L102P2	5m	Cable Fiber
10			host07						core1	L103B	3		host07 core1L103P3	5m	Cable Fiber
11			host08						core1	L104A	4		host08 core1L104P4	5m	Cable Fiber
12		rack1	sw11		Ethernet1/19	SW		rack3	core1	L108A	9	CL	sw11P19 core1L108P9	1m	Cable CU
13			sw12		Ethernet1/20				core1	L108A	10		sw12P20 core1L108P10	1m	Cable CU
14			sw13		Ethernet1/21				core1	L108A	11		sw13P21 core1L108P11	1m	Cable CU
15			sw14		Ethernet1/22				core1	L108A	12		sw14P22 core1L108P12	1m	Cable CU
16	row2	rack4	host201	lsw	eth2	NIC	row2	rack4	sw21		Ethernet1/1	SW	host201 sw21P1	1m	Cable CU
17			host202						sw22		Ethernet1/2		host202 sw22P2	1m	Cable CU
18			host203						sw23		Ethernet1/3		host203 sw23P3	1m	Cable CU
19			host204						sw24		Ethernet1/4		host204 sw24P4	1m	Cable CU
20		rack5	host205		eth2	NIC		rack6	core2	L101B	1	CL	host205 core2L101P1	5m	Cable Fiber
21			host206						core2	L102A	2		host206 core2L102P2	5m	Cable Fiber
22			host207						core2	L103A	3		host207 core2L103P3	5m	Cable Fiber
23			host208						core2	L104B	4		host208 core2L104P4	5m	Cable Fiber
24		rack4	sw21		Ethernet1/19	SW		rack6	core2	L108B	9	CL	sw21P19 core2L108P9	1m	Cable CU
25			sw22		Ethernet1/20				core2	L108B	10		sw22P20 core2L108P10	1m	Cable CU
26			sw23		Ethernet1/21				core2	L108B	11		sw23P21 core2L108P11	1m	Cable CU
27			sw24		Ethernet1/22				core2	L108B	12		sw24P22 core2L108P12	1m	Cable CU
28	Xrow	Xrack	Xhost		eth2	NIC	Xrow	Xrack	Xswitch		Eth1	SW			
29															
30	Core Name:core1	Core Group:row1	Core Rack:rack3	Core Mode:SVD06	Core Full1										
31	Core Name:core2	Core Group:row2	Core Rack:rack6	Core Mode:SVD24	Core Full0										
32															
33	Present Leafs														
34	Core Name:core2	L105	L106	L110	L111	L112	L113								
35															
36	Omitted Spines														
37	Core Name:core2	S203	S205												

Figure 6. detailed_topology.xlsx Example



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Standard-Format Topology Spread Sheet																	
2			Source						Destination				Cable			Link		
3	Rack Group	Rack	Name	Name-2	Port	Type	Rack Group	Rack	Name	Name-2	Port	Type	Label	Length	Details	Rate	MTU	Details
4	row1	rack1	host01	gw	eth2	NIC	row1	rack1	sw11		Eth1/1	SW	host01 sw11P1	1m	Cable CU	100g	8192	
5			host02						sw12		Eth1/2		host02 sw12P2	1m	Cable CU	100g	8192	
6			host03						sw13		Eth1/3		host03 sw13P3	1m	Cable CU	100g	8192	
7			host04						sw14		Eth1/4		host04 sw14P4	1m	Cable CU	100g	8192	
8		rack2	host05		eth2	NIC		rack3	core1	L101A	1	CL	host05 core1L101P1	5m	Cable Fiber	100g	8192	
9			host06						core1	L102B	2		host06 core1L102P2	5m	Cable Fiber	100g	8192	
10			host07						core1	L103B	3		host07 core1L103P3	5m	Cable Fiber	100g	8192	
11			host08						core1	L104A	4		host08 core1L104P4	5m	Cable Fiber	100g	8192	
12		rack1	sw11		Eth1/19	SW		rack3	core1	L108A	9	CL	sw11P19 core1L108P9	1m	Cable CU	100g	10240	
13			sw12		Eth1/20				core1	L108A	10		sw12P20 core1L108P10	1m	Cable CU	100g	10240	
14			sw13		Eth1/21				core1	L108A	11		sw13P21 core1L108P11	1m	Cable CU	100g	10240	
15			sw14		Eth1/22				core1	L108A	12		sw14P22 core1L108P12	1m	Cable CU	100g	10240	
16	row2	rack4	host201	lsw	eth2	NIC	row2	rack4	sw21		Eth1/1	SW	host201 sw21P1	1m	Cable CU	100g	8192	
17			host202						sw22		Eth1/2		host202 sw22P2	1m	Cable CU	100g	8192	
18			host203						sw23		Eth1/3		host203 sw23P3	1m	Cable CU	100g	8192	
19			host204						sw24		Eth1/4		host204 sw24P4	1m	Cable CU	100g	8192	
20		rack5	host205		eth2	NIC		rack6	core2	L101B	1	CL	host205 core2L101P1	5m	Cable Fiber	100g	8192	
21			host206						core2	L102A	2		host206 core2L102P2	5m	Cable Fiber	100g	8192	
22			host207						core2	L103A	3		host207 core2L103P3	5m	Cable Fiber	100g	8192	
23			host208						core2	L104B	4		host208 core2L104P4	5m	Cable Fiber	100g	8192	
24		rack4	sw21		Eth1/19	SW		rack6	core2	L108B	9	CL	sw21P19 core2L108P9	1m	Cable CU	100g	10240	
25			sw22		Eth1/20				core2	L108B	10		sw22P20 core2L108P10	1m	Cable CU	100g	10240	
26			sw23		Eth1/21				core2	L108B	11		sw23P21 core2L108P11	1m	Cable CU	100g	10240	
27			sw24		Eth1/22				core2	L108B	12		sw24P22 core2L108P12	1m	Cable CU	100g	10240	
28	Xrow	Xrack	Xhost		eth2	NIC	Xrow	Xrack	Xswitch		Eth1	SW						
29																		
30	Core Name:core1	Core Group:row1	Core Rack:rack3	Core Mode:SVD06	Core Full1													
31	Core Name:core2	Core Group:row2	Core Rack:rack6	Core Mode:SVD24	Core Full0													
32																		
33	Present Leafs																	
34	Core Name:core2	L105	L106	L110	L111	L112	L113											
35																		
36	Omitted Spines																	
37	Core Name:core2	S203	S205															

The previous figures show examples of links between NIC and Edge Switch (rows 4-7), NIC and Core Switch (rows 8-11), and Edge Switch and Core Switch (rows 12-15).

Source and **Destination** fields each have the following columns:

- Rack Group (first row required)

  Use this field to specify a Row or location of cluster hardware. **The first row in the spreadsheet must have a value.** If the Rack Group or Rack field is empty on any row, the script defaults the value in that field to the closest previous value.

- Rack (optional)

  Use this field to specify a rack unit number for the device.

- Name (required)

  Specifies the user-defined primary name of host or switch. Intel recommends that host names match the host names configured in `/etc/eth-tools/hosts`.

Hosts use the following information:

— **Host**: Hostname or hostdetails

— **Edge Switch**: Switchname

— **Core Leaf**: Corename or L*nnn*

- Name-2 (optional)

  For hosts, Name-2 is optional and is output as NodeDetails in the topology XML file.

- Port (required)

  Contains the port name of the NIC or Switch. If the Port field is empty, the script defaults to the closest previous value.

- Type

  Contains the device type. When creating the spreadsheet to verify external links, use the following values for type: `NIC`, `SW`, and `CL`.

  The first row must have a value. If the Type field is empty on any row, the script defaults the value to the closest previous value. The type values are:

  — **Host**: NIC

  — **Edge Switch**: SW

  — **Core Leaf**: CL for Director switch core leaf module

The **Cable** field has the following columns:

- Label - Max characters = 57
- Length
- Details

NOTE

Cable values are optional and have no special syntax.

The **Link** fields have following columns:

- Rate
- MTU
- Details

NOTE

Link fields are optional and have no special syntax. An example can be seen on `detailed_topology.xlsx`.

Core Full Statement

At the bottom of each sample topology file, there is a Core Full statement to indicate if the core switch is fully populated with all spine and leaf modules installed. If there are multiple mode core switches in the fabric, each core switch should have an entry in your `<topologyfile>.xlsx` file as shown in the following table.

Table 3. Core Full Statement Definitions

Core Name:Core01	Core Group:row1	Core Rack:rack01	Core Mode:SWD06	Core Full:0
Core Name:Core02	Core Group:row1	Core Rack:rack02	Core Mode:SWD24	Core Full:0
Core Name: Specified in "Name" Column of <topologyfile>.xlsx	Core Group: Specified in "Rack Group" Column of <topologyfile>.xlsx	Core Rack: Specified in "Rack" Column of <topologyfile>.xlsx	Core Mode: Set to core switch mode.	0: Use for partially populated director. 1: Use for fully populated director.

Present Leaf Statement

This section should be used when the Core is partially populated (Core Full:0). Present Leaf Statement is used to specify the list of all present Leafs in the Core. This section can have multiple rows for each partially populated Core in the fabric.

There is no need to list the Leaf names that have already been listed in the external link section as either Source Name or Destination Name.

Table 4. Present Leaf Statement Definitions

Core Name:core2	L105	L106	L110	L111	L112	L113
Core Name: Specified in "Name" Column of <topologyfile>.xlsx	Name of Leaf Present	Name of Leaf Present	Name of Leaf Present			

Omitted Spine Statement

This section should also be used when the Core is partially populated (Core Full:0). The Omitted Spine Statement is used to list all the missing Spines from the Core. This section can have multiple rows for each partially populated Core in the fabric.

Table 5. Omitted Spines Statement Definitions

Core Name:core2	S203	S205
Core Name: Specified in "Name" Column of <topologyfile>.xlsx	Name of Missing Spine	Name of Missing Spine

3.4.1.3 Miscellaneous Files

- `hostverify.sh`: Bash script to help verify configuration and performance of host nodes.
- `mac_to_dhcp`: Script to help generate DHCP stanzas to append to `dhcpd.conf`. Uses host and MAC addresses.
- `ethfastfabric.conf-sample`: Configuration file for `ethfastfabric`. Used in `/etc/eth-tools`.

- `mgt_config.xml-sample`: Configuration file for `ethreport`. Used in `/etc/eth-tools`.

3.5 Configuration Files for FastFabric

The FastFabric configuration files allow you to configure and change the basic settings and variables for the fabric and each of its components. These files are pushed out across the network ensuring that each component is synchronized.

Configuration files are located under the `/etc/eth-tools` directory.

Sample files are installed into `/usr/share/eth-tools/samples` with the suffix –`sample`. These files show the defaults of the given release.

NOTE

Do not edit the sample files.

Configuration files are self-documented as shown in the following example snippet.

```
#!/bin/bash
# [ICS VERSION STRING: @(#) ./fastfabric/samples/ethfastfabric.conf-sample
10_3_0_0_51 [09/20/16 23:52]
# This is a bash sourced config file which defines variables used in
# fast fabric tools.  Command line arguments will override these settings.
# Assignments should be scripted such that this file does not override
# exported environment settings, as shown in the defaults below

if [ "$CONFIG_DIR" = "" ]
then
        if [ -d /etc ]
        then
                CONFIG_DIR=/etc
        else
                CONFIG_DIR=/etc
        fi
        export CONFIG_DIR
fi

# Override default location for HOSTS_FILE
export HOSTS_FILE=${HOSTS_FILE:-$CONFIG_DIR/eth-tools/hosts}

# Override default location for CHASSIS_FILE
export SWITCHES_FILE=${SWITCHES_FILE:-$CONFIG_DIR/eth-tools/switches}
```

You can find more information about the various configuration variables in the "Environment Variables" section for the applicable CLI commands.

3.5.1 Management Configuration File

The Ethernet FastFabric Tools collect fabric data through SNMP. The management configuration file allows you to specify SNMP parameters.

The file is located under `/etc/eth-tools/mgt_config.xml`. It is used by the command `ethreport`.

This file is self-documented as shown in the example below.

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

<!-- Configuration parameters needed to use SNMP API -->
<Config>
        <!-- Common configuration that applies on all Fabric planes -->
        <Common>
                <ConfigDir>/etc/eth-tools</ConfigDir>
                <SnmpPort>161</SnmpPort>

                <!-- Supported: SNMP_VERSION_2c or SNMP_VERSION_3  -->
                <!-- <SnmpVersion>SNMP_VERSION_3</SnmpVersion>     -->
                <SnmpVersion>SNMP_VERSION_2c</SnmpVersion>

                <!-- Community string used when running SNMP_VERSION_2c -->
                <SnmpCommunityString>public</SnmpCommunityString>

                <!-- Identifies user name for SNMP session  -->
                <SnmpSecurityName>EthFastFabricUser</SnmpSecurityName>

                <!-- Supported: NOAUTH, AUTHNOPRIV, AUTHPRIV                      -->
                <!-- NOAUTH:  no authentication or encryption                     -->
                <!-- AUTHNOPRIV: authentication but no encryption                 -->
                <!-- AUTHPRIV: both authentication and encryption will be enforced -->
                <SnmpSecurityLevel>NOAUTH</SnmpSecurityLevel>

                <!-- Supported: MD5 or SHA -->
                <SnmpAuthenticationProtocol>MD5</SnmpAuthenticationProtocol>

                <!-- Supported: AES or DES -->
                <SnmpEncryptionProtocol>DES</SnmpEncryptionProtocol>

                <!-- Assumes all hosts that will be used for SNMP queries will     -->
                <!-- be configured to use the same passphrases.  Passphrases for   -->
                <!-- authentication and encryption can be different, but all hosts -->
                <!-- should use this same authentication passphrase on all hosts   -->
                <!-- that is specified in this file -->

                <SnmpAuthPassphrase>DefaultPassphrase</SnmpAuthPassphrase>
                <SnmpEncrypPassphrase>DefaultPassphrase</SnmpEncrypPassphrase>
        </Common>
        <!-- Default plane -->
        <Plane>
                <!-- Note: value 'ALL' (case sensitive) is reserved to present all enabled planes -->
                <Name>plane</Name>
                <!-- When disabled (0), this plane is ignored -->
                <Enable>1</Enable>
                <HostsFile>allhosts</HostsFile>
                <SwitchesFile>switches</SwitchesFile>
                <!-- input file to augment and verify fabric information -->
                <!-- <TopologyFile>topology.xml</TopologyFile> -->
        </Plane>
        <!-- Example of second plane that is disabled -->
        <Plane>
                <Name>plane2</Name>
                <Enable>0</Enable>
                <HostsFile>allhosts2</HostsFile>
                <SwitchesFile>switches2</SwitchesFile>
                <!-- example of overwriting an attribute defined in Common -->
                <SnmpPort>1234</SnmpPort>
        </Plane>
</Config>
```

- `ConfigDir` specifies the base directory for configuration files. A filepath defined in this file (e.g., `HostsFile`, `SwitchesFile`) is relative to the base directory unless it begins with a slash '/', in which case it is treated as an absolute path.
- `HostsFile` points to a file that defines ALL hosts in a fabric.

  See Host List Files on page 27 for more details.
- `SwitchesFile` points to a file that defines ALL switches in a fabric.

  See Switch List Files on page 28 for more details.
- `TopologyFile` points to a file that defines topology in a fabric.
- `SnmpPort` defines the SNMP query port. All hosts and switches use the same SNMP port.

NOTE

The current implement only supports SNMP v2c. `SnmpVersion` has to be `SNMP_VERSION_2c`

3.5.2 FastFabric Configuration File

The FastFabric configuration file allows you to view the default settings and modify the variables for most of the FastFabric command line options.

The file is located under `/etc/eth-tools/ethfastfabric.conf`.

A sample file is provided, and matches the internal defaults of the FastFabric tools.

NOTE

Command line arguments will override these settings.

Modifying the FastFabric Configuration File

1. To modify the configuration file, refer to the following FastFabric TUI procedures:
   - Editing the Configuration Files for Host Setup on page 48
   - Editing the Configuration Files for Host Verification on page 63
2. Adhere to the following requirements when editing the file:
   - The configuration file is a bash shell script that will be included by each tool. As such, the file should be implemented so that the environment variables defined prior to execution will not be altered.

     The sample code below shows the bash syntax that allows only uninitialized variables to be overwritten by the configuration file:

```
var= "${var:-value}"
```

3.5.3 Switches List Configuration Files

The Switches List configuration files allow you to specify the switches that FastFabric will operate against for some operations.

A sample file is provided, `/etc/eth-tools/switches`, and matches the internal defaults of the FastFabric tools.

Alternate filenames may be specified in `ethfastfabric.conf`, using environment variables or on the command line.

Modifying the Switches List Configuration Files

1. To modify the configuration files, refer to Editing the Configuration Files for Host Verification on page 63.
2. Adhere to the following requirements when editing the file:
   - Each line of the switches list file may specify a single switch, a comment, or another switches list file to include.

- Switches are specified by the switch management network IP address or by a resolvable TCP/IP name.

  NOTE

  Typically, names are used for readability.

- Files to be included may be specified using an `include` directive followed by a file name.

  In general, specified file names should be absolute path names. If relative path names are used, they will be searched for within the current directory, then `/etc/eth-tools` directory. To avoid cycling include, only one level of include is supported.

- Comments may be placed on any line by using a "`#`" to precede the comment.

  On lines with switches or `include` directives, the `#` must be white-space separated from any preceding name, IP address, or included filename.

3.5.4 Hosts List Configuration Files

The Hosts List configuration files allow you to specify the hosts that FastFabric will operate against for many operations.

A sample file is provided, `/etc/eth-tools/allhosts`, which includes `/etc/eth-tools/hosts`, and matches the internal defaults of the FastFabric tools.

Alternate filenames may be specified in `ethfastfabric.conf`, using environment variables or on the command line.

Modifying the Hosts List Configuration Files

1. To modify the configuration file, refer to the following FastFabric TUI procedures:
   - Editing the Configuration Files for Host Setup on page 48
   - Editing the Configuration Files for Host Verification on page 63
2. Adhere to the following requirements when editing the file:
   - Each line of the host list file may specify a single host, a comment, or another host list file to include. You can augment a hostname with a list of network interfaces to specify the interfaces to join the fabric. If no network interfaces are defined for a host, all available interfaces on the host will be considered as part of the fabric.
   - Hosts are specified by resolvable TCP/IP hostnames (without appended domain names). Typically, management network hostnames are specified. However, if desired, IP addresses may be used to accelerate large file transfers and other operations.
   - Files to be included may be specified using an `include` directive followed by a file name.

     In general, specified file names should be absolute path names. If relative path names are used, they will be searched for within the current directory, then `/etc/eth-tools` directory. To avoid cycling include, only one level of include is supported.

- Comments may be placed on any line by using a # to precede the comment.

  On lines with hosts or include directives, the # must be white-space separated from any preceding host name, IP address, or included file name.

3.5.5 Port Statistics Thresholds Configuration File

The `ethmon.conf` configuration file defines the thresholds for each port statistic. Error Counters are specified in absolute number of errors since last cleared. If the threshold for a given statistic is not defined or is set to 0 (disabled), the given statistic will not be checked. This file is use by the following commands:

- `ethreport`

  NOTE

  When used by `ethreport` or fabric health tools, the counts are absolute values and are applied against the counters as found in the system.

- `ethfabricanalysis`
- `ethlinkanalysis`
- `ethextractbadlinks`
- `ethextractstat`
- `ethextractstat2`
- `ethallanalysis`

The file is located under `/etc/eth-tools/ethmon.conf`.

A sample file is provided, and matches the internal defaults of the FastFabric tools.

3.5.6 Signal Integrity Thresholds Configuration File

The `ethmon.si.conf` configuration file defines thresholds for port counter signal integrity. This file allows analysis for any non-zero error counters related to signal integrity (bad cables, etc.) and can be enabled by adding the `-c` option to many FastFabric tools including:

- `ethreport`
- `ethextractbadlinks`
- `ethextractstat`
- `ethextractstat2`
- `ethlinkanalysis`
- `ethcabletest`
- `ethfabricanalysis`

The file is located under `/etc/eth-tools/ethmon.si.conf`.

A sample file is provided, and matches the internal defaults of the FastFabric tools.

3.5.7 Fabric Topology Input File

The Fabric Topology input file (`topology.xml`) allows you to specify the expected fabric topology and augmented fabric information (such as cable labels, types, lengths, node details, link details, etc.). If present, this file will be used by assorted FastFabric commands such as `ethreports`, `ethfabricanalysis`, and `ethallanalysis`.

The file is located under `/etc/eth-tools/topology.xml`.

A sample file is provided, and matches the internal defaults of the FastFabric tools.

Alternate filenames may be specified in `ethfastfabric.conf`, using environment variables or on the command line.

Modifying the Fabric Topology Input File

The following is an example of the topology input file (in XML format):

```
<?xml version="1.0" encoding="utf-8" ?>
<Report plane="plane" date="Wed Oct 14 23:27:10 2020" unixtime="1602732430"
options="-x -o topology -d 3" >
<Nodes>
    <NICs>
        <ConnectedNICCount>2</ConnectedNICCount>
        <Node id="0x000040a6b7190248">
            <IfAddr>0x000040a6b7190248</IfAddr>
            <NodeType>NIC</NodeType>
            <NodeType_Int>1</NodeType_Int>
            <NodeDesc>mindy1-ens785f0</NodeDesc>
            <Port id="0x000040a6b7190248:1">
                <PortNum>1</PortNum>
                <PortId>40a6b7190248</PortId>
                <EndMgmtIfID>0x00a86505</EndMgmtIfID>
                <MgmtIfAddr>0x000040a6b7190248</MgmtIfAddr>
                <LinkSpeedActive>100Gb</LinkSpeedActive>
                <LinkSpeedActive_Int>2</LinkSpeedActive_Int>
            </Port>
        </Node>
        <Node id="0x000040a6b7190330">
            <IfAddr>0x000040a6b7190330</IfAddr>
            <NodeType>NIC</NodeType>
            <NodeType_Int>1</NodeType_Int>
            <NodeDesc>mindy2-ens785f0</NodeDesc>
            <Port id="0x000040a6b7190330:1">
                <PortNum>1</PortNum>
                <PortId>40a6b7190330</PortId>
                <EndMgmtIfID>0x00a86506</EndMgmtIfID>
                <MgmtIfAddr>0x000040a6b7190330</MgmtIfAddr>
                <LinkSpeedActive>100Gb</LinkSpeedActive>
                <LinkSpeedActive_Int>2</LinkSpeedActive_Int>
            </Port>
        </Node>
    </NICs>
    <Switches>
        <ConnectedSwitchCount>1</ConnectedSwitchCount>
        <Node id="0x0000444ca8cbf441">
            <IfAddr>0x0000444ca8cbf441</IfAddr>
            <NodeType>SW</NodeType>
            <NodeType_Int>2</NodeType_Int>
            <NodeDesc>aw-arista-7060-01</NodeDesc>
            <Port id="0x0000444ca8cbf441:0">
                <PortNum>0</PortNum>
                <PortId></PortId>
                <EndMgmtIfID>0x00e4d4e7</EndMgmtIfID>
```

```
                <MgmtIfAddr>0x0000444ca8cbf441</MgmtIfAddr>
                <LinkSpeedActive>None</LinkSpeedActive>
                <LinkSpeedActive_Int>0</LinkSpeedActive_Int>
            </Port>
            <Port id="0x0000444ca8cbf441:21">
                <PortNum>21</PortNum>
                <PortId>Eth21</PortId>
                <LinkSpeedActive>100Gb</LinkSpeedActive>
                <LinkSpeedActive_Int>2</LinkSpeedActive_Int>
            </Port>
            <Port id="0x0000444ca8cbf441:25">
                <PortNum>25</PortNum>
                <PortId>Eth25</PortId>
                <LinkSpeedActive>100Gb</LinkSpeedActive>
                <LinkSpeedActive_Int>2</LinkSpeedActive_Int>
            </Port>
            <Port id="0x0000444ca8cbf441:29">
                <PortNum>29</PortNum>
                <PortId>Eth29</PortId>
                <LinkSpeedActive>100Gb</LinkSpeedActive>
                <LinkSpeedActive_Int>2</LinkSpeedActive_Int>
            </Port>
        </Node>
    </Switches>
</Nodes>
<LinkSummary>
    <LinkCount>2</LinkCount>
    <Link id="0x000040a6b7190248:1">
        <Rate>100g</Rate>
        <Rate_Int>16</Rate_Int>
        <Internal>0</Internal>
        <Port id="0x000040a6b7190248:1">
            <IfAddr>0x000040a6b7190248</IfAddr>
            <MgmtIfAddr>0x000040a6b7190248</MgmtIfAddr>
            <PortNum>1</PortNum>
            <PortId>40a6b7190248</PortId>
            <NodeType>NIC</NodeType>
            <NodeType_Int>1</NodeType_Int>
            <NodeDesc>mindy1-ens785f0</NodeDesc>
        </Port>
        <Port id="0x0000444ca8cbf441:29">
            <IfAddr>0x0000444ca8cbf441</IfAddr>
            <PortNum>29</PortNum>
            <PortId>Eth29</PortId>
            <NodeType>SW</NodeType>
            <NodeType_Int>2</NodeType_Int>
            <NodeDesc>aw-arista-7060-01</NodeDesc>
        </Port>
    </Link>
    <Link id="0x000040a6b7190330:1">
        <Rate>100g</Rate>
        <Rate_Int>16</Rate_Int>
        <Internal>0</Internal>
        <Port id="0x000040a6b7190330:1">
            <IfAddr>0x000040a6b7190330</IfAddr>
            <MgmtIfAddr>0x000040a6b7190330</MgmtIfAddr>
            <PortNum>1</PortNum>
            <PortId>40a6b7190330</PortId>
            <NodeType>NIC</NodeType>
            <NodeType_Int>1</NodeType_Int>
            <NodeDesc>mindy2-ens785f0</NodeDesc>
        </Port>
        <Port id="0x0000444ca8cbf441:21">
            <IfAddr>0x0000444ca8cbf441</IfAddr>
            <PortNum>21</PortNum>
            <PortId>Eth21</PortId>
            <NodeType>SW</NodeType>
            <NodeType_Int>2</NodeType_Int>
            <NodeDesc>aw-arista-7060-01</NodeDesc>
        </Port>
```

```
    </Link>
</LinkSummary>
</Report>
```

Related Links

Sample Topology Spreadsheet Overview on page 33

4.0 FastFabric TUI Menus

This section describes the Intel® Ethernet Fabric Suite FastFabric TUI menus used to perform common fabric management tasks.

The menus guide you through the administration process for each of the following components:

- Managing the Host Configuration
- Verifying the Host

4.1 Managing the Host Configuration

The FastFabric Ethernet Host Setup menu allows you to set up and install the Fabric software on all the hosts.

To access up the FastFabric Ethernet Host Setup Menu, perform the following steps:

1. Log in to the server as root.
2. At the command prompt, enter **`ethfastfabric`**.

   The Ethernet FastFabric Tools menu is displayed.

```
Intel Ethernet FastFabric Tools
Version: X.X.X.X.X

   1) Host Setup
   2) Host Verification/Admin

   X) Exit (or Q)
```

3. Type **1**.

   The FastFabric Ethernet Host Setup menu is displayed.

```
FastFabric Ethernet Host Setup Menu
Host File: /etc/eth-tools/hosts
Setup:
0) Edit Management Config File               [ Skip  ]
1) Edit FF Config and Select/Edit Host File  [ Skip  ]
2) Verify Hosts Pingable                     [ Skip  ]
3) Set Up Password-Less SSH/SCP              [ Skip  ]
4) Copy /etc/hosts to All Hosts              [ Skip  ]
5) Show uname -a for All Hosts               [ Skip  ]
6) Install/Upgrade Intel Ethernet Software   [ Skip  ]
7) Configure SNMP                            [ Skip  ]
8) Build Test Apps and Copy to Hosts         [ Skip  ]
9) Reboot Hosts                              [ Skip  ]
Admin:
a) Refresh SSH Known Hosts                   [ Skip  ]
b) Rebuild MPI Library and Tools             [ Skip  ]
c) Run a Command on All Hosts                [ Skip  ]
d) Copy a File to All Hosts                  [ Skip  ]
Review:
e) View ethhostadmin Result Files            [ Skip  ]
```

```
P) Perform the Selected Actions N) Select None
X) Return to Previous Menu (or ESC or Q)
```

4. Select one or more items by typing the alphanumeric character associated with the item to toggle the selection from `Skip` to `Perform`.
5. Type **`P`** to perform the operations.

NOTE

Each menu item will present you with prompts to complete the operation.

Table 6. FastFabric Ethernet Host Setup Menu Descriptions

Menu Item	Description
`0) Edit Management Config File`	Allows you to edit `/etc/eth-tools/mgt_config.xml`. This Management Config file specifies the planes in a fabric and the SNMP query parameters.
`1) Edit Config and Select/Edit Host File`	Allows you to edit the following configuration files: • `/etc/eth-tools/hosts` The `hosts` file lists the names of the hosts in a cluster except the FastFabric toolset node. • `/etc/eth-tools/ethfastfabric.conf` The `ethfastfabric.conf` file lists the default settings for most of the FastFabric command line options **NOTE:** The `hosts` file selected and created using this menu should not list the FastFabric host itself.
`2) Verify Hosts Pingable`	Allows you to ping all the hosts listed through the Management Network. Associated CLI command: `ethpingall`
`3) Set Up Password-Less SSH/SCP`	**(Linux)** Allows you to set up secure password-less SSH such that the Fabric Management Node can securely log into all the other hosts as root through the management network without requiring a password. Associated CLI command: `ethsetupssh`
`4) Copy /etc/hosts to All Hosts`	**(Linux)** Allow you to copy the `/etc/hosts` file on this host to all the other selected hosts. **NOTE:** This is not necessary when using a DNS server to resolve host names for the cluster. Associated CLI command: `ethscpall`
`5) Show uname -a for All Hosts`	**(Linux)** Allows you to view the OS version on all the hosts. In typical clusters, all hosts are running the same OS and kernel version. Associated CLI command: `ethcmdall`
`6) Install/Upgrade Intel Ethernet Software`	**(Host)** Allows you to install the Intel® Ethernet Fabric Suite software on all the hosts. Associated CLI command: `ethhostadmin`, options: `load` and `update`
`7) Configure SNMP`	**(Host)** Allows you to configure SNMP on all hosts to allow SNMP query from management node. Associated CLI command: `ethsetupsnmp`
`8) Build Test Apps and Copy to Hosts`	**(Host)** Allows you to build the MPI sample benchmarks on the Fabric Management Node and copy the resulting object files to all the hosts. Associated CLI commands: `ethscpall`, `ethuploadall`, and `ethcmdall`

continued...

Menu Item	Description
`9) Reboot Hosts`	**(Linux)** Allows you to reboot all the selected hosts and to ensure they reboot fully (as verified using ping over the management network). When the hosts come back up, they will be running the software installed. Associated CLI command: `ethhostadmin reboot`
`a) Refresh SSH Known Hosts`	**(Linux)** Allows you to refresh the ssh known hosts list on this server for the Management Network. In addition, this option may be used to update security for this host to complete installation of the hosts or if hosts are installed, replaced, reinstalled, renamed, or repaired. Associated CLI command: `ethsetupssh`
`b) Rebuild MPI Library and Tools`	**(Host)** Allows you to rebuild the MPI Library and related tools (such as mpirun). Associated CLI commands: `ethscpall`, `ethuploadall`, and `ethcmdall`
`c) Run a Command on All Hosts`	**(Linux)** Allows you to run a command on all hosts. **NOTE:** A Linux shell command (or sequence of commands separated by semicolons) may be specified to be executed against all selected hosts. Associated CLI command: `ethcmdall`
`d) Copy a File to All Hosts`	**(Linux)** Allow you to copy a file to all hosts. **NOTE:** A file on the local host may be specified to be copied to all selected hosts. Associated CLI command: `ethscpall`
`e) View ethhostadmin Result Files`	Allows you to view the `test.log` and `test.res` files that reflect the results from `ethhostadmin` runs (such as for installing software or rebooting all hosts per menu items above).

4.1.1 Editing Management Config File for Host Setup

The **Edit Management Config File** selection allows you to edit the management config file to specify the planes in a fabric and the SNMP query parameters.

1. From the FastFabric Ethernet Host Setup menu, type **0**.

   The menu item changes from `[Skip]` to `[Perform]`.

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **P** to begin the operation.

```
Performing Host Setup: Edit Management Config File
Using vi (to select a different editor, export EDITOR).
About to: vi /etc/eth-tools/mgt_config.xml
Hit any key to continue (or ESC to abort)...
```

3. Press any key to open the `mgt_config.xml` file or **ESC** to abort the operation.

   The configuration file opens.

4. Review the settings.

   Refer to Management Configuration File on page 37 for more information.

5. Save and close the `mgt_config.xml` file in the editor.

4.1.2 Editing the Configuration Files for Host Setup

The **Edit Config and Select/Edit Host File** selection allows you to edit the hosts and FastFabric configuration files.

1. From the FastFabric Ethernet Host Setup menu, type **1**.

   The menu item changes from `[Skip]` to `[Perform]`.

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **P** to begin the operation.

```
Performing Host Setup: Edit Config and Select/Edit Host File
Using vi (to select a different editor, export EDITOR).
You will now have a chance to edit/review the FastFabric Config File:
/etc/eth-tools/ethfastfabric.conf
The values in this file will control the default operation of the
FastFabric Tools.  With the exception of the host file to use,
the values you specify for defaults will be used for all FastFabric
Operations performed via this menu system
Beware existing environment variables will override the values in this file.

About to: vi /etc/eth-tools/ethfastfabric.conf
Hit any key to continue (or ESC to abort)...
```

3. Press any key to open the `ethfastfabric.conf` file or **ESC** to abort the operation.

   NOTE

   To get to subsequent configuration files, you must access each file.

   The configuration file opens.

4. Review the settings.

   Refer to FastFabric Configuration File on page 39 for more information.

5. After saving and closing the `ethfastfabric.conf` file in the editor, you will be given the opportunity to edit the `hosts` file.

```
The FastFabric operations which follow will require a file
listing the hosts to operate on
You should select a file which OMITS this host
Select Host File to Use/Edit [/etc/eth-tools/hosts]:
```

6. Press any key to open the `hosts` file or **ESC** to abort the operation.

   The configuration file opens.

   Refer to Hosts List Configuration Files on page 40 for more information.

   For further details about the Host Lists file format, refer to Host List Files on page 27.

7. Create the file with a list of the hosts names (the TCP/IP management network names), except the Management Node from which FastFabric is presently being run.

Enter one host's name per line. For example:

```
host1
host2
```

NOTE

Do not list the Management Node itself (the node where FastFabric is currently running).

If additional Management Nodes are to be used, they may be listed at this time, and FastFabric can aid in their initial installation and verification.

8. After saving and closing the `hosts` file in the editor, you will be given the opportunity to review and change the configuration files again.

```
Selected Host File: /etc/eth-tools/hosts
Do you want to edit/review/change the files? [y]:
```

9. Press **Enter** to review and edit the files or type **n** and press **Enter** to end the operation.

4.1.3 Verifying Hosts are Pingable

(All) The **Verify Hosts Pingable** selection pings each selected host over the management network.

1. From the FastFabric Ethernet Host Setup menu, type **2**.

   The menu item changes from `[Skip]` to `[Perform]`.

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **P** to begin the operation.

```
Performing Host Setup: Verify Hosts Pingable
Would you like to verify hosts are ssh-able? [n]:
```

3. Press **Enter** to select the default (n) or enter **y** and press **Enter**.

   The status is displayed.

```
Executing: /usr/sbin/ethfindgood -A -R -f /etc/eth-tools/hosts
1 hosts will be checked
1 hosts are pingable (alive)
1 hosts are alive (good)
0 hosts are bad (bad)
Bad hosts have been added to /root/punchlist.csv
Hit any key to continue (or ESC to abort)...
```

4. If some hosts were not found, press **ESC** and use the following list to assist in troubleshooting:
   - Host powered on and booted?
   - Host connected to management network?

- Host management network IP address and network settings consistent with DNS or `/etc/hosts`?
- Management node connected to the management network?
- Management node IP address and network settings correct?
- Management network itself up (including switches and others)?
- Correct set of hosts listed in the hosts file? You may need to repeat the previous step to review and edit the file.

After fixing the issues, restart this task.

5. If all hosts were found, press any key to continue.

```
Would you like to now use /etc/eth-tools/good as Host File? [y]:
```

6. Press **Enter** to select the default (y) or enter **n** and press **Enter** to end the operation.

4.1.4 Setting Up Password-Less SSH/SCP

(Linux) The **Setup Password-less ssh/scp** selection allows you to set up secure password-less SSH (root password) such that the Management Node can securely log in to all the other hosts as root through the management network without requiring a password.

NOTE

Password-less SSH is required by Intel® Ethernet Fabric Suite FastFabric, MPI test applications, and most versions of MPI (including Open MPI).

1. From the FastFabric Ethernet Host Setup menu, type **3**.

   The menu item changes from `[Skip]` to `[Perform]`.

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **P** to begin the operation.

```
Performing Host Setup: Set Up Password-Less SSH/SCP
Executing: /usr/sbin/ethsetupssh -S -p -i '' -f /etc/eth-tools/hosts
Password for root on all hosts:
```

3. Type the password for root on all hosts and press **Enter**.

4.1.5 Copying /etc/hosts to All Hosts

(Linux) The **Copy /etc/hosts to all hosts** selection allows you to copy the `/etc/hosts` file on this host to all the other selected hosts.

Typically, `/etc/resolv.conf` is set up as part of OS installation for each host. However, if `/etc/resolv.conf` was not set up on all the hosts during OS installation, the **Copy a File to All Hosts** operation could be used at this time to copy `/etc/resolv.conf` from the Management Node to all the other nodes.

NOTE

If DNS is being used, this task is not required.

1. From the FastFabric Ethernet Host Setup menu, type **4**.

   The menu item changes from `[Skip]` to `[Perform]`.

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **P** to begin the operation.

```
Performing Host Setup: Copy /etc/hosts to All Hosts
Executing: /usr/sbin/ethscpall -p -f /etc/eth-tools/hosts /etc/hosts /etc/
hosts
scp  -q  /etc/hosts root@[phgppriv11]:/etc/hosts
Hit any key to continue (or ESC to abort)...
```

3. Press any key to continue or **ESC** and press **y** to cancel the operation.

4.1.6 Showing uname -a for All Hosts

(Linux) The **Show uname -a for All Hosts** selection allows you to show the OS version on all the hosts.

1. From the FastFabric Ethernet Host Setup menu, type **5**.

   The menu item changes from `[Skip]` to `[Perform]`.

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **P** to begin the operation.

```
Performing Host Setup: Show uname -a for All Hosts
Executing: /usr/sbin/ethcmdall -T 60 -f /etc/eth-tools/hosts 'uname -a'
[root@phgppriv11]# uname -a
Linux phgppriv11.ph.intel.com 3.10.0-123.el7.x86_64 #1 SMP Mon May 5 11:16:57
EDT 2014 x86_64 x86_64 x86_64 GNU/Linux
Hit any key to continue (or ESC to abort)...
```

3. Press any key or **ESC** to end the operation.

4. Review the results to verify all the hosts have the expected OS version.
   - In typical clusters, all hosts are running the same OS and kernel version.
   - If any hosts are identified with an incorrect OS version, the OS on those hosts should be corrected at this time.

     After the OS versions have been corrected, perform Copying a File to All Hosts on page 58.

4.1.7 Installing/Upgrading Eth Software

(Host) The **Install/Upgrade Eth Software** selection allows you to install or upgrade the Intel® Ethernet Host Software on all the hosts. By default, it looks in the current directory for the `IntelEth-[Basic|FS].`*`DISTRO`*`.`*`VERSION`*`.tgz` file. If the file is not found in the current directory, the installer application prompts for a directory name where this file can be found.

NOTE

Refer to the *Intel® Ethernet Fabric Suite Software Installation Guide* for performing first-time installations and upgrades.

1. From the FastFabric Ethernet Host Setup menu, type **6**.

   The menu item changes from `[Skip]` to `[Perform]`.

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **P** to begin the operation.
3. For each prompt, provide the required information and press **Enter**:

Prompt	Description
`Enter Directory to get IntelEth-[Basic|FS].`*`DISTRO.VERSION`*`.tgz from (or none):`	Allows you to enter the directory to the software. If `none`, you will be prompted whether you want to proceed: • Select `y` to continue. • Select `n` to abort.
`Do you want to use ./IntelEth-[Basic|FS].`*`DISTRO.VERSION`*`.tgz? [y]:`	Allows you to select the tgz that is required for the installation or upgrade.
`Would you like to do a fresh [i]nstall, an [u]pgrade or [s]kip this step? [u]:`	• Select `i` to install software. • Select `u` to upgrade software. • Select `s` to skip this step.
`Are you sure you want to proceed? [n]:`	

After executing the prompts, the following is displayed:

```
 /usr/sbin/ethhostadmin -f /etc/eth-tools/hosts -d . load
Executing load Test Suite (load) Day Mth DD HH:MM:SS timezone yyyy ...
.
.
.
Hit any key to continue (or ESC to abort)...
```

NOTE

If any hosts fail to be installed, you will see results as shown in the following example:

```
TEST SUITE load: 1 Cases; 0 PASSED; 1 FAILED
TEST SUITE load FAILED
```

Use the Viewing ethhostadmin Result Files on page 79 option to review the result files from the update.

4. Press any key or **ESC** to end the operation.

4.1.8 Configuring SNMP

The **Configuring SNMP** selection allows you to configure SNMP on the hosts.

1. From the FastFabric Ethernet Host Setup menu, type **7**.

   The menu item changes from `[Skip]` to `[Perform]`.

NOTE

More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **P** to begin the operation.
   - Follow the prompts to complete the operation.
   - For FastFabric to work properly, you accept the FastFabric-required MIBs.

```
Performing Host Setup: Configure SNMP
Executing: /usr/sbin/ethsetupsnmp -p -f /etc/eth-tools/allhosts
Configuring SNMP...
Enter space separated list of admin hosts (mindy1):
Enter SNMP community string (public):
Fast Fabric requires the following MIBs:
        1.3.6.1.2.1.1 (SNMPv2-MIB:system)
        1.3.6.1.2.1.2 (IF-MIB:interfaces)
        1.3.6.1.2.1.4 (IP-MIB:ip)
        1.3.6.1.2.1.10.7 (EtherLike-MIB:dot3)
        1.3.6.1.2.1.31.1 (IP-MIB:ifMIBObjects)
Do you accept these MIBs [y/n] (y):
Enter space separated list of extra MIBs to support (NONE):

Will config SNMP with the following settings:
  admin hosts: mindy1
  community: public
  MIBs: 1.3.6.1.2.1.1 1.3.6.1.2.1.2 1.3.6.1.2.1.4 1.3.6.1.2.1.10.7 1.3.6.1.2.1.31.1
Do you accept these settings [y/n] (y):
scp  -q  /usr/sbin/ethsetupsnmp root@[mindy1]:/tmp/ethsetupsnmp
scp  -q  /usr/sbin/ethsetupsnmp root@[mindy2]:/tmp/ethsetupsnmp
[root@phwfstl005]# /tmp/ethsetupsnmp -l -a 'mindy1' -c 'public' -m '1.3.6.1.2.1.1 1.3.6.1.2.1.2
1.3.6.1.2.1.4 1.3.6.1.2.1.10.7 1.3.6.1.2.1.31.1 ';rm -f /tmp/ethsetupsnmp
Configuring SNMP...
SNMP configuration completed
[root@phwfstl006]# /tmp/ethsetupsnmp -l -a 'mindy1' -c 'public' -m '1.3.6.1.2.1.1 1.3.6.1.2.1.2
1.3.6.1.2.1.4 1.3.6.1.2.1.10.7 1.3.6.1.2.1.31.1 ';rm -f /tmp/ethsetupsnmp
Configuring SNMP...
SNMP configuration completed
SNMP configuration completed
Hit any key to continue (or ESC to abort)...
```

3. Press any key or **ESC** to end the operation.

4.1.9 Building Test Applications and Copying to Hosts

(Host) The **Build Test Apps and Copy to Hosts** selection allows you to build the MPI sample applications on the Management Node and copy the resulting object files to all the hosts. This is in preparation for execution of MPI performance tests and benchmarks.

NOTE

Refer to the *Intel® Ethernet Fabric Suite Software Release Notes* for the latest supported MPI Library versions.

1. From the FastFabric Ethernet Host Setup menu, type **8**.

   The menu item changes from `[Skip]` to `[Perform]`.

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **P** to begin the operation.

```
Performing Host Setup: Build Test Apps and Copy to Hosts
Do you want to build MPI Test Apps? [y]:
```

3. Press **Enter**.

```
Enter location (or none) for copy and build MPI Apps [/root/mpi_apps]:
```

4. Press **Enter** to accept the default location, or type in a desired location for MPI Apps, or 'none' to end the operation.

   The MPI Directory Selection TUI is displayed.

```
Host Setup: Build Test Apps and Copy to Hosts
MPI Directory Selection

Please Select MPI Directory:
0) /usr/mpi/gcc/openmpi-4.1.1-cuda-ofi
1) /usr/mpi/gcc/openmpi-4.1.1-ofi
2) /opt/intel/impi/2019.10.000/intel64
3) Enter Other Directory

X) Return to Previous Menu (or ESC or Q)
```

5. Select the target menu item or type **X** to end the operation.
6. If no CUDA directory is detected, go to step 9.

   If only one CUDA directory is detected, go to step 8.

   Otherwise, the CUDA Directory Selection TUI is displayed.

```
Host Setup: Build Test Apps and Copy to Hosts
CUDA Directory Selection

Please Select CUDA Directory:
0) /usr/local/cuda-10.3
```

```
1) /usr/local/cuda-11.4

X) Return to Previous Menu (or ESC or Q)
```

NOTE

FastFabric searches for the CUDA directory in the location defined by the environment variable *FF_CUDA_DIR*. Its default value is defined in FastFabric configuration file `/etc/eth-tools/ethfastfabric.conf`.

7. Select the desired CUDA directory or type **X** to end the operation.

```
Build with CUDA support? [y]:
```

8. Press **Enter** to build MPI APPs with CUDA support, or type `n` to build without CUDA support.
9. Follow the prompts to complete the operation.

4.1.10 Rebooting Hosts

(Linux) The **Reboot Hosts** selection allows you to reboot all the selected hosts and ensure they fully reboot, as verified through ping over the management network.

1. From the FastFabric Ethernet Host Setup menu, type **9**.

   The menu item changes from `[Skip]` to `[Perform]`.

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **P** to begin the operation.

   Reboot begins immediately.

```
Performing Host Setup: Reboot Hosts
Executing: /usr/sbin/ethhostadmin -f /etc/eth-tools/hosts reboot
Executing reboot Test Suite (reboot) Fri Oct 07 11:58:48 EDT 2020 ...
Executing TEST SUITE reboot CASE (reboot.phgppriv11.reboot) phgppriv11
reboot ...
```

4.1.11 Refreshing SSH Known Hosts

(Linux) The **Refresh SSH Known Hosts** selection allows you to refresh the SSH known hosts list on this server for the Management Network. This may be used to update security for this host if hosts are replaced, reinstalled, renamed, or repaired.

1. From the FastFabric Ethernet Host Setup menu, type **a**.

   The menu item changes from `[Skip]` to `[Perform]`.

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **P** to begin the operation.

```
Performing Host Setup: Refresh SSH Known Hosts
Executing: /usr/sbin/ethsetupssh -p -U -f /etc/eth-tools/hosts
Verifying localhost ssh...
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
localhost: Connected
Warning: Permanently added 'phgppriv10,10.10.10.10' (ECDSA) to the list of
known hosts.
phgppriv10: Connected
...
Successfully processed: X
Hit any key to continue (or ESC to abort)...
```

3. Press any key or **ESC** to end the operation.

4.1.12 Rebuilding MPI Library and Tools

(Host) The **Rebuild MPI Library and Tools** allows you to rebuild the MPI Library and related tools (such as `mpirun`), and install the resulting rpms on all the hosts.

This operation is performed using the `do_build` tool supplied with the MPI Source. When rebuilding MPI, `do_build` prompts you for selection of which MPI to rebuild, and provides choices as to which available compiler to use. Refer to *Intel® Ethernet Fabric Suite Software Installation Guide* and *Intel® Ethernet Fabric Suite Host Software User Guide* for more information.

1. From the FastFabric Ethernet Host Setup menu, type **b**.

   The menu item changes from `[Skip]` to `[Perform]`.

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **P** to begin the operation.

```
Performing Host Setup: Rebuild MPI Library and Tools
Executing: cd //usr/src/eth/MPI; ./do_build

IEFS MPI Library/Tools rebuild
1) openmpi
Select MPI to Build:
```

3. Enter the menu item to rebuild and press **Enter**.
4. Rebuild openmpi.

NOTE

Open MPI included in the IEFS package does not include C++ bindings by default (in keeping with MPI standard 3.0). To build Open MPI with C++ bindings, set `CONFIG_OPTIONS` environment variable to `--enable-mpi-cxx` before executing `do_build` script.

```
IEFS OpenMPI MPI Library/Tools rebuild
1) gcc
Select Compiler:
```

a. Enter the menu item and press **Enter**.

   If `libfabric-devel` was installed, below will show on screen.

```
Build for OFI [y]:
```

b. Press **Enter** to continue or **n** and **Enter** to abort.

```
Executing: cd /usr/src/eth/MPI && /usr/sbin/ethscpall -p -f /etc/eth-
tools/hosts  /var/tmp
...
Hit any key to continue (or ESC to abort)...
```

c. Press any key to continue.

```
Executing: /usr/sbin/ethcmdall -p -f /etc/eth-tools/hosts 'cd /var/tmp;
rpm -U --force ; rm -f '
[root@phgppriv11]# cd /var/tmp; rpm -U --force ; rm -f
...
Hit any key to continue (or ESC to abort)...
```

d. Press any key or **ESC** to end operation.

4.1.13 Running a Command on All Hosts

(Linux) The **Run a Command on All Hosts** selection allows you to perform other operations on all hosts. Each time this is executed, a Linux shell command may be specified to be executed against all selected hosts. You can also specify a sequence of commands separated by semicolons.

1. From the FastFabric Ethernet Host Setup menu, type **c**.

   The menu item changes from `[Skip]` to `[Perform]`.

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **P** to begin the operation.

```
Performing Host Setup: Run a Command on All Hosts
Enter Command to run on all hosts (or none):
```

3. Enter a Linux command and press **Enter**.

```
Timelimit in minutes (0=unlimited): [1]:
```

4. Specify a time limit and press **Enter**.

```
Run in parallel on all hosts? [y]:
```

5. Select **y** (yes) or **n** (no) and press **Enter**.

```
About to run: /usr/sbin/ethcmdall  -T 60 -f /etc/eth-tools/hosts 'xxxx'
Are you sure you want to proceed? [n]:
```

6. Type **y** and press **Enter** to proceed with the operation.

   The operation is completed.

4.1.14 Copying a File to All Hosts

(Linux) The **Copy a File to All Hosts** selection allows you to run the `ethscpall` command. A file on the local host may be specified to be copied to all selected hosts.

1. From the FastFabric Ethernet Host Setup menu, type **d**.

   The menu item changes from `[Skip]` to `[Perform]`.

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **P** to begin the operation.

```
Performing Host Setup: Copy a File to All Hosts
Enter File to copy to all hosts (or none):
```

3. Enter the name of the file to copy and press **Enter**.

```
Are you sure you want to proceed? [n]:
```

4. Type **y** and press **Enter** to continue.

```
Executing: /usr/sbin/ethscpall -p -f /etc/eth-tools/hosts /root/xxx /root/xxx
scp  -q  /root/xxx root@[phgppriv11]:/root/xxx
...
Hit any key to continue (or ESC to abort)...
```

5. Press any key or **ESC** to end the operation.

4.1.15 Viewing ethhostadmin Result Files

(All) The **View ethhostadmin Result File** selection allows you to display the `test.log` and `test.res` files that contain the results from prior `ethhostadmin` runs, such as installing Fabric software or rebooting all hosts. You are also given the option to remove these files after viewing them.

If prior files are not removed, subsequent runs of `ethhostadmin` from within the current directory continue to append to these files.

NOTE

For more information on the log files, refer to Interpreting the ethhostadmin log files and `ethhostadmin` Details.

1. From the FastFabric Ethernet Host Setup menu, type **`e`**.

   The menu item changes from `[Skip]` to `[Perform]`.

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **`P`** to begin the operation.

```
Performing Host Setup: View ethhostadmin Result Files
Using vi (to select a different editor, export EDITOR).
About to: vi /root/test.res /root/test.log
Hit any key to continue (or ESC to abort)...
```

3. Press any key to view the `ethhostadmin` results files.
4. After reviewing and closing the log, you are prompted to remove the following files.

```
Would you like to remove test.res test.log test_tmp* and save_tmp
in /root ? [n]:
```

5. Select **`y`** (yes) or **`n`** (no) and press **Enter**.
6. If you chose **`y`** in the step above, press any key or **ESC** to end the operation.

4.2 Verifying the Host

The FastFabric Ethernet Host Verification/Admin Menu allows you to verify hosts and the fabric, as well as manage of all the hosts.

To access up the FastFabric Ethernet Host Setup Menu, perform the following steps:

1. Log in to the server as root.

2. At the command prompt, enter **ethfastfabric**.

   The FastFabric EthernetTools menu is displayed.

```
Intel Ethernet FastFabric Tools
Version: X.X.X.X.X

   1) Host Setup
   2) Host Verification/Admin

   X) Exit
```

3. Type **2**.

   The FastFabric Ethernet Host Verification/Admin Menu is displayed.

```
FastFabric Ethernet Host Verification/Admin Menu
Plane: plane
Host File: /etc/eth-tools/allhosts
Plane:
0) Edit Management Config and Select Plane          [ Skip  ]
Validation:
1) Edit FF Config and Select/Edit Host File         [ Skip  ]
2) Summary of Fabric Components                     [ Skip  ]
3) Verify Hosts Are Pingable, SSHable, and Active   [ Skip  ]
4) Perform Single Host Verification                 [ Skip  ]
5) Verify Eth Fabric Status and Topology            [ Skip  ]
6) Verify Hosts Ping via RDMA                       [ Skip  ]
7) Verify PFC via empirical test                    [ Skip  ]
8) Refresh SSH Known Hosts                          [ Skip  ]
9) Check MPI Performance                            [ Skip  ]
a) Check Overall Fabric Health                      [ Skip  ]
b) Start or Stop Bit Error Rate Cable Test          [ Skip  ]
Admin:
c) Generate All Hosts Problem Report Info           [ Skip  ]
d) Run a Command on All Hosts                       [ Skip  ]
Review:
e) View ethhostadmin Result Files                   [ Skip  ]

P) Perform the Selected Actions              N) Select None
X) Return to Previous Menu (or ESC or Q)
```

4. Select one or more items by typing the alphanumeric character associated with the item to toggle the selection from `Skip` to `Perform`.
5. Type **P** to perform the operations.

NOTE

Each menu item will present you with prompts to complete the operation.

Table 7. FastFabric Ethernet Host Verification/Admin Menu Descriptions

Menu Item	Description
`0) Edit Management Config and Select Plane`	Allows you to edit `/etc/eth-tools/mgt_config.xml` and select the fabric plane to verify. This Management Config file specifies the planes in a fabric and the SNMP query parameters
`1) Edit Config and Select/Edit Host File`	Allows you to edit the following configuration files: • `/etc/eth-tools/allhosts`

continued...

Menu Item	Description
	The `allhosts` file lists of all hosts including the FastFabric toolset node. • `/etc/eth-tools/ethfastfabric.conf` The `ethfastfabric.conf` file lists the default settings for most of the FastFabric command line options.
`2) Summary of Fabric Components`	Allows you to view a brief summary of the components in the fabric including the number of components, how many switch chips, NICs, and links. It also indicates whether any degraded or omitted (quarantined or out of policy) links were found. Associated CLI command: `ethfabricinfo` described in *Intel® Ethernet Fabric Suite Host Software User Guide*
`3) Verify Hosts Are Pingable, SSHable, and Active`	Allows you to ping all the hosts listed through the Management Network. Associated CLI command: `ethpingall`
`4) Perform Single Host Verification`	Allows you to perform verification on all nodes in the selected host file including configuration, performance, and stability using a variety of tools and checks including single node HPL. For additional information on the verification that is performed, refer to the `/usr/share/eth-tools/samples/hostverify.sh` file. Associated CLI commands: `ethcheckload` and `ethverifyhosts`
`5) Verify Eth Fabric Status and Topology`	**(Host or All)** Allows you to review the fabric state and error counts of all ports. Associated CLI commands: `ethshowallports` and `ethreport`
`6) Verify Hosts Ping via RDMA`	**(Host)** Allows you to verify that RDMA is properly configured and running on all the hosts. This is accomplished through the Fabric node pinging each other using rping. Associated CLI command: `ethhostadmin rping`
`7) Verify PFC via empirical test`	**(Host)** Allows you to verify that PFC is properly configured and running on all the hosts and switches via an empirical test. This is accomplished through an RDMA UD stress test on the Fabric nodes. The test verifies the expected pause frames occur and that there are no packet drops under an 8:1 incast traffic pattern. Associated CLI command: `ethhostadmin pfctest`
`8) Refresh SSH Known Hosts`	**(Linux)** Allows you to refresh the ssh known hosts list on this server for the Management Networks. This option may be used to update security for this host to complete installation of the hosts or if hosts are replaced, reinstalled, renamed, or repaired. Associated CLI command: `ethsetupssh`
`9) Check MPI Performance`	**(Host)** Allows you to perform a quick check of PCI and MPI performance using end-to-end latency and bandwidth tests. Associated CLI command: `ethcheckload` and `ethhostadmin`
`a) Check Overall Fabric Health`	**(Host)** Allows you to check the overall fabric health. Associated CLI command: `ethallanalysis`
`b) Start or Stop Bit Error Rate Cable Test`	**(Host)** Allows you to start or stop the Cable Bit Error Rate stress tests for NIC-to-switch links. Associated CLI command: `ethcabletest`

continued...

Menu Item	Description
c) Generate All Hosts Problem Report Info	**(Host)** Allows you to collect configuration and status information from all hosts and generates a single *.tgz file, which can be sent to a support representative. Associated CLI command: ethcaptureall
d) Run a Command on All Hosts	**(Linux)** Allows you to execute a command on all hosts. Associated CLI command: ethhostadmin
e) View ethhostadmin Result Files	Allows you to view the test.log and test.res files that reflect the results from ethhostadmin runs (such as those for installing software or rebooting all hosts per menu items above).

4.2.1 Editing Management Config File and Selecting Plane for Host Verification

The **Edit Management Config File and Select Plane** selection allows you to edit the management config file to specify the planes in a fabric and select the plane for verification.

1. From the FastFabric Ethernet Host Verification/Admin menu, type **0**.

   The menu item changes from [Skip] to [Perform].

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **P** to begin the operation.

```
Performing Host Setup: Edit Management Config File
Using vi (to select a different editor, export EDITOR).
About to: vi /etc/eth-tools/mgt_config.xml
Hit any key to continue (or ESC to abort)...
```

3. Press any key to open the mgt_config.xml file or **ESC** to abort the operation.

   The configuration file opens.

4. Review the settings.

   In particular, review the following:

   - Plane
     - Name
     - Enable
     - HostsFile
     - SwitchesFile
     - TopologyFile

   Refer to Management Configuration File on page 37 for more information.

NOTE

Intel recommends that a FastFabric topology file is created for each plane as `/etc/eth-tools/topology_<plane>.xml` to describe the intended topology. The file can also augment assorted fabric reports with customer-specific information, such as cable labels and additional details about nodes, links, ports, and cables. Refer to Fabric Topology Input File on page 42, Topology Verification on page 181, and ethreport Detailed Information on page 114 for more information about topology verification files.

5. Save and close the `mgt_config.xml` file in the editor.
6. If there are more than one enabled planes defined in `mgt_config.xml`, the Plane Selection TUI is displayed.

```
Host Admin: Edit Management Config and Select Plane
Plane Selection

Please Select Fabric Plane:
0) plane
1) plane2

X) Return to Previous Menu (or ESC or Q)
```

   Otherwise, go to step 8.
7. Select the target menu item or type **X** to end the operation.
8. Press **y** to confirm the selection, or press **Enter** to edit `mgt_config.xml` and select a plane.

   The Host Verification/Admin menu displays the selected plane and plane hosts file (noted in bold) on head panel

```
FastFabric Ethernet Host Verification/Admin Menu
Plane: plane
Host File: /etc/eth-tools/allhosts
```

4.2.2 Editing the Configuration Files for Host Verification

The **Edit Config and Select/Edit Host File** section allows you to select and edit the hosts and FastFabric configuration files.

1. From the FastFabric Ethernet Host Verification/Admin menu, type **1**.

   The menu item changes from `[Skip]` to `[Perform]`.

NOTE

More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **P** to begin the operation.

```
Performing Host Admin: Edit Config and Select/Edit Host File
Using vi (to select a different editor, export EDITOR).
You will now have a chance to edit/review the FastFabric Config File:
/etc/eth-tools/ethfastfabric.conf
The values in this file will control the default operation of the
```

```
FastFabric Tools.  With the exception of the host file to use,
the values you specify for defaults will be used for all FastFabric
Operations performed via this menu system
Beware existing environment variables will override the values in this file.

About to: vi /etc/eth-tools/ethfastfabric.conf
Hit any key to continue (or ESC to abort)...
```

3. Press any key to open the `ethfastfabric.conf` file or `ESC` to abort the operation.

   NOTE

   To get to subsequent configuration files, you must access each file.

   The configuration file opens.

4. Review the settings.

   Especially review the following:

   - `FF_DEVIATION_ARGS`

   Refer to FastFabric Configuration File on page 39 for more information.

5. After saving and closing the `ethfastfabric.conf` file in the editor, you will be given the opportunity to edit the target plane's *HostsFile* file.

```
The FastFabric operations which follow will apply on
plane 'plane' hosts defined in /etc/eth-tools/allhosts
About to: vi /etc/eth-tools/allhosts
Hit any key to continue (or ESC to abort)...
```

6. Press any key to open the `allhosts` file or **ESC** to abort the operation.

   The configuration file opens.

   Refer to Hosts List Configuration Files on page 40 for more information.

   For further details about the Host Lists file format, refer to Host List Files on page 27.

7. Create the file with the Management Node's host name (the TCP/IP management network name, for example `mgmthost`) and the ethernet port names (name of the ethernet interfaces that connect to the fabric plane, for example cvl0), and include a hosts file contains host list for other hosts.

   Enter one host's name per line. For example:

```
mgmthost:cvl0
include /etc/eth-tools/hosts
```

8. After saving and closing the `allhosts` file in the editor, you will be given the opportunity to edit the selected plane's *SwitchesFile* file.

```
The FastFabric operations which follow will apply on
plane switches defined in /etc/eth-tools/switches
About to: vi /etc/eth-tools/switches
Hit any key to continue (or ESC to abort)...
```

9. Press any key to open the `switches` file or **ESC** to abort the operation.

   The configuration file opens.

For further details about the Switches Lists file format, refer to Switch List Files on page 28.

10. After saving and closing the `switches` file in the editor, you will be given the opportunity to review and change the configuration files again.

```
Do you want to edit/review/change the files? [y]:
```

11. Press **Enter** to review and edit the files again or type **n** and press **Enter** to end the operation.

4.2.3 Viewing a Summary of Fabric Components

The **Summary of Fabric Components** selection allows you to generate a brief summary of the counts of components in the fabric plane, including how many switch chips, hosts, and links are in the fabric. The summary also indicates whether any degraded or omitted links were found, which can indicate a poorly seated or bad cable, incorrect fabric configuration, or security issues.

1. From the FastFabric Ethernet Host Verification/Admin menu, type **2**.

   The menu item changes from `[Skip]` to `[Perform]`.

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **P** to begin the operation.

   The summary is generated.

```
Performing Host Admin: Summary of Fabric Components
Executing: /usr/sbin/ethfabricinfo -p plane -f /etc/eth-tools/allhosts
Done Getting All Fabric Records
Number of NICs: 2
Number of Switches: 0
Number of Links: 1
Number of NIC Links: 1              (Internal: 0   External: 1)
Number of ISLs: 0                   (Internal: 0   External: 0)
Number of Slow Links: 0             (NIC Links: 0   ISLs: 0)
Number of Omitted Links: 0          (NIC Links: 0   ISLs: 0)
------------------------------------------------------------------------------
-
Hit any key to continue (or ESC to abort)...
```

3. Press any key or **ESC** to end the operation.

4.2.4 Verifying Hosts Pingable, SSHable, and Active

The **Verify Hosts Pingable, SSHable, and Active** selection allows you to verify each host and provides a concise summary of the bad hosts found.

Interactive prompts allow you to select ping, SSH, and port active verification. After completion of this test, you have the option of using the resulting good hosts file for the remainder of the operations within this TUI session.

1. From the FastFabric Ethernet Host Verification/Admin menu, type **3**.

   The menu item changes from `[Skip]` to `[Perform]`.

NOTE

More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **P** to begin the operation.
3. For each prompt, provide the required information and press **Enter**:

Prompt	Description
`Would you like to verify hosts are ssh-able? [y]:`	Allows you to see which hosts are ssh-able.
`Would you like to verify host RDMA ports are active? [y]:`	Allows you to view which RDMA ports are active.

After executing the prompts, the results are displayed.

```
Executing: /usr/sbin/ethfindgood  -f /etc/eth-tools/allhosts
2 hosts will be checked
2 hosts are pingable (alive)
2 hosts are ssh'able (running)
2 total hosts have RMDA active ports on one or more fabrics (active)
2 hosts are alive, running, active (good)
0 hosts are bad (bad)
Bad hosts have been added to /root/punchlist.csv
Hit any key to continue (or ESC to abort)...
```

The following files are created in `ethsorthosts` with all duplicates removed in the `CONFIG_DIR/` directory:

- good
- alive
- running
- active
- bad

The resulting `good` file can then be used in as input for subsequent verification commands and to create `mpi_hosts` files for running mpi_apps and the NIC-SW cable test.

4. If some hosts were not found, press **ESC** and use the following list to assist in troubleshooting:
   - Host powered on and booted?
   - Host connected to management network?
   - Host management network IP address and network settings consistent with DNS or `/etc/hosts`?
   - Management node connected to the management network?
   - Management node IP address and network settings correct?
   - Management network itself up (including switches and others)?
   - Correct set of hosts listed in the hosts file? You may need to repeat the previous step to review and edit the file.

   After fixing the issues, restart this task.

5. If all hosts were found, press any key to continue.

```
Would you like to now use /etc/eth-tools/good as Host File? [y]:
```

6. Press **Enter** to use the host file or type **n** and press **Enter** to end the operation

4.2.5 Performing Single Host Verification

The **Perform Single Host Verification** selection allows you to perform a single host test on all hosts.

NOTES

- Prior to using this selection, you must have a copy of the `hostverify.sh` in the directory pointed to by `FF_HOSTVERIFY_DIR`.
- If the file does not exist in that directory, copy the sample file `/usr/share/eth-tools/samples/hostverify.sh` to the directory pointed to by `FF_HOSTVERIFY_DIR`. When placed in the editor to review `hostverify.sh`, review the settings near the top and the list of TESTS selected, edit and save as needed.
- This test can be run on a subset of hosts placed in a file created under `/etc/eth-tools`. The test then allows tailoring `hostverify.sh` for that subset. The tailored `hostverify.sh` can be saved with a unique suffix using the subset filename.
- Review the HPL variables in the `hostverify.sh` script to control how much memory pressure is used for single node HPL validation (default is 30%). The goal of the single node HPL test is to check node stability and the consistency of performance between hosts, NOT to optimize performance. For optimizing HPL performance, refer to the *Intel® Ethernet Fabric Performance Tuning Guide*.

1. From the FastFabric Ethernet Host Verification/Admin menu, type **4**.

   The menu item changes from `[Skip]` to `[Perform]`.

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

   When operating on different subsets, select menu items `0` and `3` each time. Then select the desired host file while following the flow for menu item `0`.

2. Type **P** to begin the operation.
3. For each prompt, provide the required information and press **Enter**:

Prompt	Description
`FastFabric needs to create a working file for verification` `Would you like to use /root/hostverify_allhosts.sh? [y]:`	Defines the generated hostverify script name. The name can be specific to the hosts file you are using with a filename as

continued...

Prompt	Description
	hostverify_<hosts_filename>, or default filename hostverify_default.
Would you like to copy /usr/share/eth-tools/samples/hostverify.sh to /root/hostverify_allhosts.sh? [n]:	Allows you to copy the hostverify.sh file from the sample directory in order to edit for use. *Note:* The copy location is dependent on the file name under /etc/eth-tools used for listing the hosts to operate on using FastFabric Ethernet Host Verification/Admin menu Step 0. (/etc/hosts/allhosts is used in this example.)
In below file /root/hostverify_allhosts.sh, leaving NIC_IFS empty will use the interfaces defined in /etc/eth-tools/allhosts. Would you like to edit /root/hostverify_allhosts.sh? [y]:	Allows you to edit the hostverify_*.sh file. Follow the settings explanations in the file to review and edit the settings near the top. The next prompt will appear after you close the file.
Would you like to copy /root/hostverify_allhosts.sh to hosts? [y]:	Allows you to copy the local hostverify.sh to the destination host. Choose n only if /root/hostverify_allhosts.sh on hosts has not changed.
Would you like to specify tests to run? [n]:	Allows you to run specific tests.
Enter filename for upload destination file [hostverify.res]:	Allows you to enter a file name for the results file or use the default hostverify.res.
Timelimit in minutes: [1]:	Allows you to set the time limit for the tests.
View Load on hosts prior to verification? [y]:	Allows you to view the load on the hosts before verification begins.

After executing the prompts, the average loads per host are displayed.

```
Executing: /usr/sbin/ethcheckload  -f /etc/eth-tools/allhosts
loadavg                              host
0.00 0.01 0.05 2/1161 3044      phkpstl085
0.00 0.01 0.05 2/1161 3044      phkpstl085
0.00 0.01 0.05 2/1117 25477     phkpstl087
0.00 0.01 0.05 1/1118 25164     phkpstl086
Hit any key to continue (or ESC to abort)...
```

4. Press any key to start the tests.

```
Executing: /usr/sbin/ethverifyhosts -k -c -u hostverify.res -T 60 -f /etc/eth-
tools/allhosts
-F /root/hostverify_allhosts.sh
Killing hostverify and xhpl on hosts...
[root@phkpstl085]# pkill -9 -f -x 'host[v]erify.*.sh'; pkill -9 '[x]hpl';
echo -n
[root@phkpstl086]# pkill -9 -f -x 'host[v]erify.*.sh'; pkill -9 '[x]hpl';
echo -n
[root@phkpstl087]# pkill -9 -f -x 'host[v]erify.*.sh'; pkill -9 '[x]hpl';
echo -n
[root@phkpstl085]# pkill -9 -f -x 'host[v]erify.*.sh'; pkill -9 '[x]hpl';
echo -n
3 hosts will be verified
SCPing /root/hostverify_allhosts.sh to /root/hostverify.sh ...
scp -q /root/hostverify_allhosts.sh root@[phkpstl086]:/root/hostverify.sh
scp -q /root/hostverify_allhosts.sh root@[phkpstl087]:/root/hostverify.sh
scp -q /root/hostverify_allhosts.sh root@[phkpstl085]:/root/hostverify.sh
scp -q /root/hostverify_allhosts.sh root@[phkpstl085]:/root/hostverify.sh
Running /root/hostverify.sh -d /root ...
Killing hostverify and xhpl on hosts...
[root@phkpstl087]# pkill -9 -f -x 'host[v]erify.*.sh'; pkill -9 '[x]hpl';
```

```
echo -n
[root@phkpstl086]# pkill -9 -f -x 'host[v]erify.*.sh'; pkill -9 '[x]hpl';
echo -n
[root@phkpstl085]# pkill -9 -f -x 'host[v]erify.*.sh'; pkill -9 '[x]hpl';
echo -n
[root@phkpstl085]# pkill -9 -f -x 'host[v]erify.*.sh'; pkill -9 '[x]hpl';
echo -n
Uploading /root/hostverify.res to ./uploads/hostverify.res ...
scp -q root@[phkpstl086]:/root/hostverify.res ./uploads/phkpstl086/
hostverify.res
scp -q root@[phkpstl087]:/root/hostverify.res ./uploads/phkpstl087/
hostverify.res
scp -q root@[phkpstl085]:/root/hostverify.res ./uploads/phkpstl085/
hostverify.res
scp -q root@[phkpstl085]:/root/hostverify.res ./uploads/phkpstl085/
hostverify.res
About to: vi /root/verifyhosts.res
Hit any key to continue (or ESC to abort)...
```

5. Press any key to view the results file.

   The results of the test are shown in the editor.

6. Close the results file to end the operation.

4.2.6 Verifying Eth Fabric Status and Topology

The **Verify Eth Fabric Status and Topology** selection allows you to run various checks on the fabric plane and topology.

1. From the FastFabric Ethernet Host Verification/Admin menu, type **5**.

   The menu item changes from `[Skip]` to `[Perform]`.

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **P** to begin the operation.
3. For each prompt, provide the required information and press **Enter**:

Prompt	Description
`Would you like to perform fabric error analysis? [y]:`	Allows you to perform fabric error analysis.
`Would you like to perform fabric link speed error analysis? [y]:`	Allows you to perform link speed error analysis.
`Check for links configured to run slower than supported? [n]:`	Allow you to look for links that are configured to run slower than supported.
`Check for links connected with mismatched speed potential? [n]:`	Allows you to look for connected links with mismatch speed potential.
`Would you like to verify fabric topology? [y]:`	Allows you to verify the fabric topology.

continued...

Prompt	Description
	Note: The fabric deployment can be verified against the planned topology. Typically, the planned topology will have been converted to an XML topology file using `ethxlattopology` or a customized variation. If this step has been done and a topology file has been placed in the location specified by the `FF_TOPOLOGY_FILE` in ethfastfabric.conf file, then a topology verification can be performed. Refer to Topology Verification on page 181 and ethreport Detailed Information on page 114 for more information.
`Verify all aspects of topology (links, nodes)? [y]:`	Allows you to verify all links and nodes in the topology.
`Include unexpected devices in punchlist? [y]:`	Allows you to include unexpected devices in the punchlist.
`Enter filename for results [/root/linkanalysis.res]:`	Allows you to enter a file name for the result file or accept the default `linkanalysis.res`.

After executing the prompts, the average loads per host are displayed.

```
Executing: /usr/sbin/ethcheckload  -f /etc/eth-tools/allhosts
loadavg                                host
0.66 0.41 0.20 2/880 193597     phgppriv10
0.00 0.01 0.05 1/813 4001       phgppriv11
Hit any key to continue (or ESC to abort)...
```

The following items are verified:

- Perform a fabric error analysis.
- Perform a fabric link speed error analysis.
- Check for links that are configured to run slower than supported.
- Check links that are connected with mismatched speed potential.
- Verify the fabric topology.
- Verify all aspects of the topology including links and nodes.
- Include unexpected devices in the punchlist.

The results can be seen in the `$FF_RESULT_DIR/linkanalysis.res` file. A punch list of issues is appended to the `$FF_RESULT_DIR/punchlist.csv` file.

4. Press any key or **ESC** to end the operation.

4.2.7 Verifying Hosts Ping via RDMA

(Host) The **Verify Hosts Ping via RDMA** selection allows you to confirm that RDMA is properly configured and running on all the hosts. This is accomplished through the Fabric node pinging each other through rping.

1. From the FastFabric Ethernet Host Verification/Admin menu, type **6**.

   The menu item changes from `[Skip]` to `[Perform]`.

NOTE

More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **p** to begin the operation.

   The status is displayed.

```
Performing Host Admin: Verify Hosts Ping via RDMA
Executing: /usr/sbin/ethhostadmin -f /etc/eth-tools/allhosts rping
Executing rdma ping Test Suite (rping) Thu Oct 15 14:50:11 EDT 2020 ...
Executing TEST SUITE rdma ping CASE (rping.phwfstl014) phwfstl014 rping ...
Executing TEST SUITE rdma ping CASE (rping.phwfstl015) phwfstl015 rping ...
TEST SUITE rdma ping CASE (rping.phwfstl014) phwfstl014 rping ...
TEST SUITE rdma ping ITEM (rping.phwfstl014.eth3-phwfstl015) phwfstl015 to
phwfstl014:eth3 PASSED
TEST CASE phwfstl014 rping: 1 Items; 1 PASSED
TEST SUITE rdma ping CASE (rping.phwfstl014) phwfstl014 rping PASSED
TEST SUITE rdma ping CASE (rping.phwfstl015) phwfstl015 rping ...
TEST SUITE rdma ping ITEM (rping.phwfstl015.eth3-phwfstl014) phwfstl014 to
phwfstl015:eth3 PASSED
TEST CASE phwfstl015 rping: 1 Items; 1 PASSED
TEST SUITE rdma ping CASE (rping.phwfstl015) phwfstl015 rping PASSED
TEST SUITE rdma ping: 2 Cases; 2 PASSED
TEST SUITE rdma ping PASSED
Done rdma ping Test Suite Thu Oct 15 14:50:17 EDT 2020

Hit any key to continue (or ESC to abort)...
```

3. Press any key or **ESC** to end the operation.

4.2.8 Verifying PFC via Empirical Test

(Host) The **Verify PFC via empirical test** selection allows you to verify that PFC is properly configured and running on all the hosts and switches via an empirical test. This is accomplished through an RDMA UD stress test on the Fabric nodes. The test verifies that the expected pause frames occur, and there are no packet drops under an 8:1 incast traffic pattern.

1. From the FastFabric Ethernet Host Verification/Admin menu, type **7**.

   The menu item changes from `[Skip]` to `[Perform]`.

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **p** to begin the operation.

   The status is displayed.

```
Performing Host Admin: Verify PFC via empirical test
Executing: /usr/sbin/ethhostadmin -f /etc/eth-tools/allhosts pfctest
Executing Empirical PFC Test Test Suite (pfctest) Mon Jul 12 19:08:57 EDT
2021 ...
Executing TEST SUITE Empirical PFC Test CASE
(pfctest.hdtsfnb1011.hd.intel.com) In-cast hdtsfnb1011.hd.intel.com<--
(hdtsfnb2271.hd.intel.com hdtsfnb2281.hd.intel.com
hdtsfnb1031.hd.intel.com) ...

...
TEST SUITE Empirical PFC Test CASE (pfctest.hdtsfnb1011.hd.intel.com) In-cast
hdtsfnb1011.hd.intel.com<--(hdtsfnb2271.hd.intel.com hdtsfnb2281.hd.intel.com
hdtsfnb1031.hd.intel.com) ...
TEST SUITE Empirical PFC Test ITEM
(pfctest.hdtsfnb1011.hd.intel.com.:ens785f0<--(hdtsfnb2271.hd.intel.com
hdtsfnb2281.hd.intel.com hdtsfnb1031.hd.intel.com)) In-cast
```

```
hdtsfnb1011.hd.intel.com:ens785f0<--(hdtsfnb2271.hd.intel.com
hdtsfnb2281.hd.intel.com hdtsfnb1031.hd.intel.com) PASSED
TEST CASE In-cast hdtsfnb1011.hd.intel.com<--(hdtsfnb2271.hd.intel.com
hdtsfnb2281.hd.intel.com hdtsfnb1031.hd.intel.com): 1 Items; 1 PASSED
TEST SUITE Empirical PFC Test CASE (pfctest.hdtsfnb1011.hd.intel.com) In-cast
hdtsfnb1011.hd.intel.com<--(hdtsfnb2271.hd.intel.com hdtsfnb2281.hd.intel.com
hdtsfnb1031.hd.intel.com) PASSED
...
TEST SUITE Empirical PFC Test: 4 Cases; 4 PASSED
TEST SUITE Empirical PFC Test PASSED
Done Empirical PFC Test Test Suite Mon Jul 12 19:09:04 EDT 2021

Hit any key to continue (or ESC to abort)...
```

3. Press any key or **ESC** to end the operation.

If any hosts fail, carefully examine the failing hosts and switches in the route path to verify the PFC configuration. Refer to switch manual and *Intel® Ethernet Fabric Performance Tuning Guide* for detailed information.

4.2.9 Refreshing SSH Known Hosts

(Linux) The **Refresh SSH Known Hosts** selection allows you to refresh the SSH `known_hosts` file on the Management Node to include all the hosts.

1. From the FastFabric Ethernet Host Verification/Admin menu, type **8**.

   The menu item changes from `[Skip]` to `[Perform]`.

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **P** to begin the operation.

   The status is displayed.

```
Performing Host Admin: Refresh SSH Known Hosts
Executing: /usr/sbin/ethsetupssh -p -U -f /etc/eth-tools/allhosts
Verifying localhost ssh...
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
localhost: Connected
Warning: Permanently added 'phgppriv10,10.228.209.74' (ECDSA) to the list of
known hosts.
phgppriv10: Connected
...
Successfully processed: X
Hit any key to continue (or ESC to abort)...
```

3. Press any key or **ESC** to end the operation.

4.2.10 Checking MPI Performance

(Host) The **MPI Performance** selection allows you to perform a quick check of PCIe and MPI performance through end-to-end latency and bandwidth tests.

NOTE

This test identifies nodes whose performance is not consistent with others in the fabric. It is not intended as a benchmark of fabric latency and bandwidth. This test purposely uses techniques to reduce test runtime.

1. From the FastFabric Ethernet Host Verification/Admin menu, type **9**.

   The menu item changes from `[Skip]` to `[Perform]`.

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **P** to begin the operation.

   The status is displayed.

```
Performing Host Admin: Check MPI Performance
Test Latency and Bandwidth deviation between all hosts? [y]:
```

3. Press **Enter** to select default (y).

```
View Load on hosts prior to test? [y]:
```

4. Press **Enter** to select default (y).

```
Performing Host Admin: Check MPI Performance
Test Latency and Bandwidth deviation between all hosts? [y]:
View Load on hosts prior to test? [y]:
Executing: /usr/sbin/ethcheckload  -f /etc/eth-tools/allhosts
loadavg                                  host
1.00 1.00 1.00 2/778 27345      hdtsfnb2271.hd.intel.com
1.00 1.00 1.00 2/777 27203      hdtsfnb2281.hd.intel.com
0.02 0.01 0.00 1/786 44108      hdtsfnb1011.hd.intel.com
0.00 0.00 0.00 1/775 27745      hdtsfnb1031.hd.intel.com
Hit any key to continue (or ESC to abort)...
```

5. Press any key to continue.

```
Executing: /usr/sbin/ethhostadmin -f /etc/eth-tools/allhosts mpiperfdeviation
Executing mpi lat/bw deviation Test Suite (mpiperfdeviation) Fri Dec 04
12:51:54 EST 2020 ...
Executing TEST SUITE mpi lat/bw deviation CASE
(mpiperfdeviation.localhost.deviation) localhost starts openmpi deviation ...
TEST SUITE mpi lat/bw deviation CASE (mpiperfdeviation.localhost.deviation)
localhost starts openmpi deviation PASSED
PERF openmpi deviation for hdtsfnb1031.hd.intel.com hdtsfnb2271.hd.intel.com
hdtsfnb2281.hd.intel.com hdtsfnb1011.hd.intel.com:
PERF /usr/mpi/gcc/openmpi-4.0.5-ofi/bin/mpirun  -np 4 -allow-run-as-root --
map-by node -machinefile mpi_hosts
$MPI_CMD_ARGS  ./deviation -bwtol 20 -lattol 50 -c; echo DONE
PERF
PERF Trial runs of 4 hosts are being performed to find
PERF the best host since no baseline host was specified.
```

```
PERF
PERF Baseline host is hdtsfnb1011 (0)
PERF
PERF Running Concurrent MPI Latency Tests - Pairs 2   Testing     2
PERF Running Concurrent MPI Bandwidth Tests - Pairs 2   Testing     2
PERF
PERF Concurrent MPI Performance Test Results
PERF   Latency Summary:
PERF     Min: 6.80 usec, Max: 6.81 usec, Avg: 6.80 usec
PERF     Range: +0.2% of Min, Worst: +0.1% of Avg
PERF     Cfg: Tolerance: +50% of Avg, Delta: 0.80 usec, Threshold: 10.20 usec
PERF          Message Size: 0, Loops: 4000
PERF
PERF   Bandwidth Summary:
PERF     Min: 3486.9 MB/s, Max: 3515.6 MB/s, Avg: 3501.2 MB/s
PERF     Range: -0.8% of Max, Worst: -0.4% of Avg
PERF     Cfg: Tolerance: -20% of Avg, Delta: 150.0 MB/s, Threshold: 2801.0
MB/s
PERF          Message Size: 2097152, Loops: 30 BiDir: no
PERF
PERF Latency: PASSED
PERF Bandwidth: PASSED
PERF DONE
PERF
--------------------------------------------------------------------------------

TEST SUITE mpi lat/bw deviation: 1 Cases; 1 PASSED
TEST SUITE mpi lat/bw deviation PASSED
Done mpi lat/bw deviation Test Suite Fri Dec 04 12:51:57 EST 2020
Hit any key to continue (or ESC to abort)...
```

The results display the pair-wise analysis of latency and bandwidth for the selected hosts, and report pairs outside an acceptable tolerance range. By default, performance is compared relative to other hosts in the fabric. It is assumed that all hosts selected for a given run have comparable fabric performance. Failing hosts are clearly indicated.

Intel recommends that you review the `FF_DEVIATION_ARGS` parameter in `ethfastfabric.conf` and adjust it as appropriate for the cluster. The default can accommodate a wide range of cluster designs.

The results are also written to the `test.res` file, which may be viewed through Viewing ethhostadmin Result Files on page 79.

6. Press any key or **ESC** to end the operation.

Additional Details

If any hosts fail, carefully examine the failing hosts to verify the NIC models, PCIe slot used, BIOS settings, and any motherboard or BIOS settings related to devices on PCIe buses or slot speeds. Also verify the NIC and any riser cards are properly seated.

The bandwidth that is reported should also be checked against the PCIe speeds in the Performance Impact table below. If all pairs are not in the expected performance range, carefully examine all hosts to verify the NIC models, PCIe slot used, BIOS settings, and any motherboard or BIOS settings related to devices on PCIe buses or slot speeds. Also verify the NIC and any riser cards are properly seated.

Table 8. Performance Impact

PCIe Speed	Fabric Speed	Typical Bandwidth
PCIe 8GT/s x16 (Gen3)	100 Gbps	12.0 - 12.4 GBps
PCIe 8GT/s x8 (Gen3)	100 Gbps	6.4 - 6.8 GBps
PCIe 5GT/s x16 (Gen2)	100 Gbps	6.4 - 6.8 GBps
PCIe 5GT/s x8 (Gen2)	100 Gbps	3.2 - 3.4 GBps
Note: 1 GBps = 1,000,000,000 bytes/second		

4.2.11 Checking Overall Fabric Health

The **Check Overall Fabric Health** selection allows you to baseline the present fabric configuration for use in future fabric health checks. Perform this check after configuring any additional Management Nodes and establishing a healthy fabric via successful execution of all the other tests. If desired, a baseline of an incomplete or unhealthy fabric may be taken for future comparison after making additions or corrections to the fabric.

Refer to Configure and Initialize Health Check Tools in the *Intel® Ethernet Fabric Suite Software Installation Guide* for more information.

1. From the FastFabric Ethernet Host Verification/Admin menu, type **a**.

   The menu item changes from `[Skip]` to `[Perform]`.

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **p** to begin the operation.

   The status is displayed.

```
Performing Host Admin: Check Overall Fabric Health
Baseline present configuration? [n]:
```

3. Press **Enter** (n) to analyze the configuration without baselining it.

```
Executing: /usr/sbin/ethallanalysis -p plane -f /etc/eth-tools/allhosts
ethfabricanalysis: Fabric(s) OK
ethallanalysis: All OK
Hit any key to continue (or ESC to abort)...
```

4. Type **y** and press **Enter** to baseline the configuration.

   The configuration is baselined.

```
Executing: /usr/sbin/ethallanalysis -b -p plane -f /etc/eth-tools/allhosts
ethfabricanalysis: Baselined
ethallanalysis: Baselined
Hit any key to continue (or ESC to abort)...
```

5. Press any key or **ESC** to end the operation.

4.2.12 Starting or Stopping Bit Error Rate Cable Test

The **Start or Stop Bit Error Rate Cable Test** selection allows you to perform host cable testing. The test allows for starting and stopping an extended Bit Error Rate test.

Intel recommends that you run this test for 20-60 minutes for a thorough test. While the test is running, monitor the fabric for signal integrity or stability errors using `ethreport`. Once the desired test time has elapsed, return to this item in the menu and stop the test.

1. From the FastFabric Ethernet Host Verification/Admin menu, type **b**.

   The menu item changes from `[Skip]` to `[Perform]`.

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **P** to begin the operation.
3. For each prompt, provide the required information and press **Enter**:

Prompt	Description
`Stop or cleanup any already running Cable Test? [y]:`	Allows you to stop and clean up any cable tests in process.
`Stop Cable Test? [y]:`	Allows you to stop cable test.
`Start Cable Test? [y]:`	Allows you to start a new cable test.

   After executing the prompts, the following is displayed.

```
About to run: /usr/sbin/ethcabletest -f /etc/eth-tools/allhosts
Hit any key to continue (or ESC to abort)...
```

4. Press any key to execute the cabletest or **ESC** to end the operation.

4.2.13 Generating All Hosts Problem Report Information

(Host) The **Generate all Hosts Problem Report Info** selection allows you to collect configuration and status information from all hosts and generate a single `*.tgz` file that can be sent to an Intel support representative.

1. From the FastFabric Ethernet Host Verification/Admin menu, type **c**.

   The menu item changes from `[Skip]` to `[Perform]`.

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **P** to begin the operation.

The status is displayed.

```
Performing Host Admin: Generate All Hosts Problem Report Info
Capture detail level (1-Normal 2-Fabric 3-Analysis): [3]:
```

The detail levels are cumulative and shown below:

Detail Level	Description
`1-Normal`	Obtains local information from each host.
`2-Fabric`	In addition to "Normal", obtains basic fabric information by queries to the fabric and fabric error analysis using `ethreport`.
`3-Analysis`	In addition to "Fabric", obtains `all_analysis` results. If `all_analysis` has not yet been run, it is run as part of the capture.
Notes: • Detail levels 2-3 can be used when fabric operational problems occur. If the problem appears to be node-specific, detail level 1 should be sufficient. • Detail levels 2-3 require an operational Fabric Manager. Typically, your support representative requests a given detail level. If a given detail level takes excessively long or fails to be gathered, try a lower detail level. • For detail levels 2-3, the additional information is only gathered on the node running the `ethcaptureall` command.	

3. Type the menu item for the level of details required for the report and press **Enter**.

   `ethcaptureall` is initiated and results gathered in a `hostcapture.all.tgz`.

   A sample of a "Normal" analysis is shown below.

```
Executing: /usr/sbin/ethcaptureall -p -D 1 -f /etc/eth-tools/allhosts
Running capture on all non-local hosts ...
[root@phwfstl006]# rm -f ~root/hostcapture.tgz; ethcapture ~root/
hostcapture.tgz
Getting software and firmware version information ...
Capturing Ethernet NIC devices
Obtaining OS configuration ...
Obtaining dmesg logs ...
Obtaining present process and module list ...
Obtaining module info for ice ...
Obtaining module info for irdma ...
Obtaining PCI device list ...
Obtaining processor information ...
Obtaining environment variables ...
Obtaining network interfaces ...
Obtaining DMI information ...
Obtaining Shared Memory information ...
Obtaining device statistics
Obtaining MPI configuration ...
Copying configuration and statistics from /proc ...
Obtaining additional CPU info...
Copying kernel debug information from /sys/kernel/debug/ice...
Copying kernel debug information from /sys/kernel/debug/irdma...
Obtaining side channel security issue mitigation information from /sys/
devices/system/cpu/vulnerabilities
Copying side channel security issue mitigation information from /sys/devices/
system/cpu/vulnerabilities...
Copying configuration and statistics for irdma from /sys ...
Copying network interface information
Copying configuration and statistics data from /sys/module ...
Copying all Cable Health Reports
Creating tar file /root/hostcapture.tgz ...
Done.

Please include /root/hostcapture.tgz with any problem reports to Customer
```

```
Support
Uploading capture from each host ...
Running capture on local host ...
scp   root@[phwfstl006]:hostcapture.tgz ./uploads/phwfstl006/.
Getting software and firmware version information ...
hostcapture.tgz
100%   17MB  55.1MB/s   00:00
Capturing Ethernet NIC devices
Obtaining OS configuration ...
Obtaining dmesg logs ...
Obtaining present process and module list ...
Obtaining module info for ice ...
Obtaining module info for irdma ...
Obtaining PCI device list ...
Obtaining processor information ...
Obtaining environment variables ...
Obtaining network interfaces ...
Obtaining DMI information ...
Obtaining Shared Memory information ...
Obtaining device statistics
Obtaining MPI configuration ...
Copying configuration and statistics from /proc ...
Obtaining additional CPU info...
Copying kernel debug information from /sys/kernel/debug/ice...
Copying kernel debug information from /sys/kernel/debug/irdma...
Obtaining side channel security issue mitigation information from /sys/
devices/system/cpu/vulnerabilities
Copying side channel security issue mitigation information from /sys/devices/
system/cpu/vulnerabilities...
Copying configuration and statistics for irdma from /sys ...
Copying network interface information
Copying configuration and statistics data from /sys/module ...
Copying all Cable Health Reports
Creating tar file /root/./uploads/phwfstl005/hostcapture.tgz ...
Done.

Please include /root/./uploads/phwfstl005/hostcapture.tgz with any problem
reports to Customer Support
Combining captured files into ./uploads/hostcapture.all.tgz ...
Done.
Hit any key to continue (or ESC to abort)...
```

4. Press any key or **ESC** to end the operation.

4.2.14 Running a Command on All Hosts

(Linux) The **Run a command on all hosts** selection allows you to perform other operations on all hosts. Each time this is executed, a Linux shell command may be specified to be executed against all selected hosts. You can also specify a sequence of commands separated by semicolons.

1. From the FastFabric Ethernet Host Verification/Admin menu, type **d**.

   The menu item changes from `[Skip]` to `[Perform]`.

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **P** to begin the operation.

```
Performing Host Admin: Run a Command on All Hosts
Enter Command to run on all hosts (or none):
```

3. Enter a Linux command and press **Enter**.

```
Timelimit in minutes (0=unlimited): [1]:
```

4. Specify a time limit and press **Enter**.

```
Run in parallel on all hosts? [y]:
```

5. Select **y** (yes) or **n** (no) and press **Enter**.

```
About to run: /usr/sbin/ethcmdall  -T 60 -f /etc/eth-tools/hosts 'xxxx'
Are you sure you want to proceed? [n]:
```

6. Type **y** and press **Enter** to proceed with the operation.

   The operation is completed.

4.2.15 Viewing ethhostadmin Result Files

The **View ethhostadmin result files** allows you to display the `test.log` and `test.res` files that contain the results from prior `ethhostadmin` runs, such as installing fabric software or rebooting all hosts. You are also given the option to remove these files after viewing them.

If prior files are not removed, subsequent runs of `ethhostadmin` from within the current directory continue to append to these files.

NOTE

For more information on the log files, refer to Interpreting the ethhostadmin log files and `ethhostadmin` Details.

1. From the FastFabric Ethernet Host Verification/Admin menu, type **e**.

   The menu item changes from `[Skip]` to `[Perform]`.

   NOTE

   More than one menu item may be selected. The operations will perform individually and in sequence with the menu.

2. Type **P** to begin the operation.

```
Performing Host Admin: View ethhostadmin Result Files
Using vi (to select a different editor, export EDITOR).
About to: vi /root/punchlist.csv /root/verifyhosts.res /root/test.res /root/
test.log
Hit any key to continue (or ESC to abort)...
```

3. Press any key to view the `ethhostadmin` results files.

4. After reviewing and closing the log, you are prompted to remove the following files.

```
4 files to edit
Would you like to remove verifyhosts.res test.res test.log test_tmp* and
save_tmp
in /root ? [n]:
```

5. Select **y** (yes) or **n** (no) and press **Enter**.
6. If you chose y in the step above, press any key or **ESC** to end the operation.

5.0 Descriptions of Command Line Tools

This section provides a complete description of each Intel® Ethernet Fabric Suite FastFabric Toolset command line tool and its parameters.

Whereas the TUI menus are presented sequentially showing you how to perform common fabric management tasks, the CLI tools provide more functional granularity and are organized by categories.

NOTE

Basic CLI tools are described in the *Intel® Ethernet Fabric Suite Host Software User Guide*.

Related Links

High-Level TUIs on page 81

Health Check and Baselining Tools on page 82

Verification, Analysis, and Control CLIs on page 96

Detailed Fabric Data Gathering on page 135

Configuration and Control for Host on page 144

Basic Setup and Administration Tools on page 151

File Management Tools on page 160

FastFabric Utilities on page 167

5.1 High-Level TUIs

The tools described in this section are used for fabric monitoring, deployment verification, and analysis.

5.1.1 ethfastfabric

Starts the top-level Intel® Ethernet Fabric Suite FastFabric Text-based User Interface (TUI) menu to enable setup and configuration.

Syntax

```
ethfastfabric
```

Options

None.

Example

```
#ethfastfabric
 Intel Ethernet FastFabric Tools
 Version: X.X.X.X.X

     1) Host Setup
     2) Host Verification/Admin

     X) Exit (or Q)
```

5.2 Health Check and Baselining Tools

The software includes tools to rapidly identify if the fabric has a problem or if its configuration has changed since the last baseline. Analysis includes hardware, software, and fabric topology. The tools are designed to permit easy manual execution or automated execution using `cron` or other mechanisms. The health check tools include:

- `ethfabricanalysis` – Performs fabric topology and error counters analysis.
- `ethallanalysis` – Performs analysis on all components or a subset of components. Intel recommends this as the primary tool for general analysis.

5.2.1 Usage Model

The health check tools support three modes of operation: health check only mode, baseline mode, and check mode. The typical usage model for the tools is:

- Perform initial fabric install and verification:
  - Optionally run tools in *health check only* mode
  - Performs quick health check
  - Duplicates some of steps already done during verification
- Run tools in *baseline* mode:
  - Takes a baseline of present hardware and software configuration
- Periodically run tools in *check* mode:
  - Performs quick health check
  - Compares present hardware and software configuration to baseline
  - Can be scheduled in hourly `cron` jobs
- As needed, rerun *baseline* when expected changes occur, including:
  - Fabric upgrades
  - Hardware replacements and changes
  - Software configuration changes

5.2.2 Common Operations and Options

The Health Check and Baselining tool supports the following options:

- `-b` - Performs a baseline snapshot of the configuration.
- `-e` - Performs an error check/health analysis only.

If no option is specified, the tool performs a snapshot of the present configuration, compares it to the baseline, and performs an error check/health analysis.

Using both `-b` and `-e` on a given run is not permitted.

A typical use case is:

- Perform an initial error check by running the `-e` option.
- Review and correct the errors reported in the files indicated by the tools.
- Once all the errors are corrected, perform a baseline of the configuration using the `-b` option. The baseline configuration is saved to files in *`FF_ANALYSIS_DIR`*`/baseline`. The default `/var/usr/lib/eth-tools/analysis/baseline` is set through `/etc/eth-tools/ethfastfabric.conf`. This baseline configuration should be carefully reviewed to make sure it matches the intended configuration. If it does not, correct the configuration and run a new baseline.

Example

```
ethfabricanalysis -e
```

Errors reported could include links with high error rates, unexpected low speeds, etc. Correct any errors, then rerun `ethfabricanalysis -e` to make sure there is a good fabric.

```
ethfabricanalysis -b
```

The baseline configuration is saved to *`FF_ANALYSIS_DIR`*`/baseline`. This includes files starting with `links` and `comps`, which are the results of `ethreport -o links` and `ethreport -o comps` reports, respectively. Review these files and make sure all the expected links and components are present. For example, make sure all the switches and servers in the cluster are present. Also, verify the appropriate links between servers and switches are present. If the fabric is not correctly configured, correct the configuration and rerun the baseline.

NOTE

Alternatively, the advanced topology verification capabilities of `ethreport` can be used to verify the fabric deployment against the intended design.

Once a good baseline has been established, use the tools to compare the present fabric against the baseline and check its health.

```
ethfabricanalysis
```

Checks the present fabric links and components against the previous baseline. If there have been changes, it reports a failure and indicates which files hold the resulting snapshot and differences. It also checks error counters and link speeds for the fabric, similar to `ethfabricanalysis -e`. If either of these checks fail, it returns a non-zero exit status, permitting higher level scripts to detect a failed condition.

The differences files are generated using the Linux command specified by `FF_DIFF_CMD` in `ethfastfabric.conf`. By default, this is the `diff -C 1` command. It is run against the baseline and new snapshot. Therefore, lines after each

*** #,# **** heading in the diff are from the baseline and lines after each --- #,# ---- heading are from the new snapshot. If FF_DIFF_CMD is simply set to diff, lines indicated by "<" in the diff are from the baseline and lines indicated by ">" in the diff are from the new snapshot.

Another useful command is the Linux sdiff command. For more information about the diff output format, consult the Linux man page for diff.

If the configuration is intentionally changed, Intel recommends that you obtain a new error analysis and baseline using the same sequence as the initial installation to establish a new baseline for future comparisons.

In addition, all of the tools support the following options:

- -s

  Saves the history of failures.

  When the -s option is used, each failed run also creates a directory whose name is the date and time the analysis tool was started. The directory contains the failing snapshot information and diffs, allowing you to track a history of failures. Note that every run of the tools also creates a latest directory with the latest snapshot. The latest files are overwritten by each subsequent run of the tool, which means the most recent run results are always available.

  CAUTION

  Frequent use of the health check tools in conjunction with -s can consume a large amount of disk space. The space requirements depend greatly on the size of the cluster. For example, it could be > 10 megabytes per run on a 1000 node cluster.

- -d *dir*

  Specifies the top-level directory for saving baseline, snapshots, and history.

  Runs using -d must use the same directory as any previous baseline to be compared to (except when the -e option is used). Default is FF_ANALYSIS_DIR which is set in ethfastfabric.conf.

  The FF_ANALYSIS_DIR option can be changed to provide a customer-specific alternate directory to be used whenever the -d option is not specified. Subdirectories under FF_ANALYSIS_DIR are created as follows:

  — baseline - Baseline snapshot from each analysis tool.

  — latest - Latest snapshot from each analysis tool.

  — *YYYY-MM-DD-HH:MM:SS* - Failed analysis from analysis run with -s.

5.2.3 ethfabricanalysis

Performs analysis of the fabric.

Syntax

```
ethfabricanalysis [-b|-e] [-s] [-d dir] [-c file]
 [-E file] [-p planes] [-T topology_inputs] [-f host_files]
```

Options

`--help`	Produces full help text.
`-b`	Specifies the baseline mode. Default is compare/check mode.
`-e`	Evaluates health only. Default is compare/check mode.
`-s`	Saves history of failures (errors/differences).
`-d` *dir*	Specifies the top-level directory for saving baseline and history of failed checks. Default is `/var/usr/lib/eth-tools/analysis`
`-c` *file*	Specifies the error thresholds config file. Default is `/etc/eth-tools/ethmon.conf`
`-E` *file*	Specifies Ethernet Mgt configuration file. The default is `/etc/eth-tools/mgt_config.xml`.
`-p` *planes*	Specifies Fabric planes separated by space. The default is the first enabled plane defined in config file. Value 'ALL' will use all enabled planes.
`-f` *host_files*	Hosts files separated by space. It overrides the HostsFiles defined in Mgt config file for the corresponding planes. Value 'DEFAULT' will use the HostFile defined in Mgt config file for the corresponding plane
`-T` *topology_inputs*	Specifies the name of topology input filenames separated by space. See Details and `ethreport` `on page 108` for more information.

Example

```
ethfabricanalysis
ethfabricanalysis -p 'p1 p2' -f 'hosts1 DEFAULT'
```

The fabric analysis tool checks the following:

- Fabric links (both internal to switch and external cables)
- Fabric components (nodes, links, systems, and their configuration)
- Fabric error counters and link speed mismatches

NOTE

The comparison includes components on the fabric. Therefore, operations such as shutting down a server cause the server to no longer appear on the fabric and are flagged as a fabric change or failure by `ethfabricanalysis`.

Environment Variables

The following environment variables are also used by this command:

`FF_ANALYSIS_DIR` Top-level directory for baselines and failed health checks.

Details

You can specify the `topology_input` file to be used with one of the following methods:

- On the command line using the `-T` option.
- Using the `TopologyFile` specified in Ethernet Mgt config file.

If the specified file does not exist, no `topology_input` file is used.

For more information on `topology_input`, refer to `ethreport` on page 108.

By default, the error analysis includes counters and slow links (that is, links running below enabled speeds). You can change this using the `FF_FABRIC_HEALTH` configuration parameter in `ethfastfabric.conf`. This parameter specifies the `ethreport` options and reports to be used for the health analysis.

When a `topology_input` file is used, it can also be useful to extend `FF_FABRIC_HEALTH` to include fabric topology verification options such as `-o verifylinks`.

The thresholds for counter analysis default to `/etc/eth-tools/ethmon.conf`. However, you can specify an alternate configuration file for thresholds using the `-c` option. The `ethmon.si.conf` file can also be used to check for any non-zero values for signal integrity counters.

All files generated by `ethfabricanalysis` start with `fabric` in their file name.

The `ethfabricanalysis` tool generates files such as the following within `FF_ANALYSIS_DIR`:

Health Check

- `latest/fabric.<plane_name>.errors`

  `stdout` of `ethreport` for errors encountered during fabric error analysis.
- `latest/fabric.<plane_name>.errors.stderr`

  `stderr` of `ethreport` during fabric error analysis.

Baseline

During a baseline run, the following files are also created in `FF_ANALYSIS_DIR/latest`.

- `baseline/fabric.<plane_name>.snapshot.xml`

  `ethreport` snapshot of complete fabric components and configuration.
- `baseline/fabric.<plane_name>.comps`

  `ethreport` summary of fabric components and basic configuration.

- `baseline/fabric.<plane_name>.links`

  `ethreport` summary of internal and external links.

Full Analysis

- `latest/fabric.<plane_name>.snapshot.xml`

  `ethreport` snapshot of complete fabric components and configuration.
- `latest/fabric.<plane_name>.snapshot.stderr`

  `stderr` of `ethreport` during snapshot.
- `latest/fabric.<plane_name>.errors`

  `stdout` of `ethreport` for errors encountered during fabric error analysis.
- `latest/fabric.<plane_name>.errors.stderr`

  `stderr` of `ethreport` during fabric error analysis.
- `latest/fabric.<plane_name>.comps`

  `stdout` of `ethreport` for fabric components and configuration.
- `latest/fabric.<plane_name>.comps.stderr`

  `stderr` of `ethreport` for fabric components.
- `latest/fabric.<plane_name>.comps.diff`

  `diff` of baseline and latest fabric components.
- `latest/fabric.<plane_name>.links`

  `stdout` of `ethreport` summary of internal and external links.
- `latest/fabric.<plane_name>.links.stderr`

  `stderr` of `ethreport` summary of internal and external links.
- `latest/fabric.<plane_name>.links.diff`

  `diff` of baseline and latest fabric internal and external links.
- `latest/fabric.<plane_name>.links.changes.stderr`

  `stderr` of `ethreport` comparison of links.
- `latest/fabric.<plane_name>.links.changes`

  `ethreport` comparison of links against baseline. This is typically easier to read than the `links.diff` file and contains the same information.
- `latest/fabric.<plane_name>.comps.changes.stderr`

  `stderr` of `ethreport` comparison of components.
- `latest/fabric.<plane_name>.comps.changes`

  `ethreport` comparison of components against baseline. This is typically easier to read than the `comps.diff` file and contains the same information.

The `.diff` and `.changes` files are only created if differences are detected.

If the `-s` option is used and failures are detected, files related to the checks that failed are also copied to the time-stamped directory name under `FF_ANALYSIS_DIR`.

Fabric Items Checked Against the Baseline

Based on `ethreport -o links`:

- Unconnected/down/missing cables
- Added/moved cables
- Changes in link width and speed
- Changes to IfAddr in fabric (replacement of NIC or Switch hardware)
- Adding/Removing Nodes (NIC, Virtual NICs, Virtual Switches, Physical Switches, Physical Switch internal switching cards (leaf/spine))
- Changes to server or switch names

Based on `ethreport -o comps`:

- Overlap with items from links report
- Changes in port MTU
- Changes in port speed/width enabled or supported
- Changes in NIC or switch device IDs/revisions/VendorID (for example, ASIC hardware changes)
- Changes in port Capability mask (which features/agents run on port/server)
- Changes to I/O Units (IOUs), I/O Controllers (IOCs), and I/O Controller Services Services provided

NOTE

Only applicable if IOUs in fabric (such as Virtual IO cards, native storage, and others).

Fabric Items Also Checked During Health Check

Based on `ethreport -s -o errors -o slowlinks`:

- Error counters on all Intel® Ethernet Fabric ports (NIC, switch external, and switch internal) checked against configurable thresholds.
  - Typically identifies potential fabric errors, such as symbol errors.
  - May also identify transient congestion, depending on the counters that are monitored.
- Link active speed/width as compared to Enabled speed.
  - Identifies links whose active speed/width is < min (enabled speed/width on each side of link).
  - This typically reflects bad cables or bad ports or poor connections.
- Side effect is the verification of fabric health.

5.2.4 ethallanalysis

`ethallanalysis` command performs the set of analysis specified in `FF_ALL_ANALYSIS` and can be specified for fabric or hosts.

Syntax

```
ethallanalysis [-b|-e] [-s] [-d dir] [-c file] [-T topology_input]
 [-E file] [-p planes] [-f host_files]
```

Options

`--help`	Produces full help text.
`-b`	Sets the baseline mode. Default is compare/check mode.
`-e`	Evaluates health only. Default is compare/check mode.
`-s`	Saves the history of failures (errors/differences).
`-d dir`	Identifies the top-level directory for saving baseline and history of failed checks. Default is `/var/usr/lib/eth-tools/analysis`
`-c file`	Specifies the error thresholds configuration file. Default is `/etc/eth-tools/ethmon.conf`
`-E file`	Ethernet Mgt configuration file. The default is `/etc/eth-tools/mgt_config.xml`.
`-p planes`	Fabric planes separated by space. The default is the first enabled plane defined in config file. Value 'ALL' will use all enabled planes.
`-f host_files`	Hosts files separated by space. It overrides the HostsFiles defined in Mgt config file for the corresponding planes. Value 'DEFAULT' will use the HostFile defined in Mgt config file for the corresponding plane
`-T topology_inputs`	Specifies the name of topology input filenames separated by space. See ethreport on page 108 for more information on `topology_input` files.

Example

```
ethallanalysis
ethallanalysis -p 'p1 p2' -f 'hosts1 DEFAULT'
```

Environment Variables

The following environment variables are also used by this command:

`FF_ANALYSIS_DIR` Top-level directory for baselines and failed health checks.

Details

The `ethallanalysis` command performs the set of analysis specified in `FF_ALL_ANALYSIS`, which must be a space-separated list. This can be provided by the environment or using `/etc/eth-tools/ethfastfabric.conf`. The analysis set includes the options: fabric.

Note that the `ethallanalysis` command has options that are a superset of the options for all other analysis commands. The options are passed along to the respective tools as needed. For example, the `-c` file option is passed on to `ethfabricanalysis` if it is specified in `FF_ALL_ANALYSIS`.

The output files are all the output files for the `FF_ALL_ANALYSIS` selected set of analysis. See the previous sections for the specific output files.

5.2.5 Manual and Automated Usage

There are two basic ways to use the tools:

- Manual

  Run the tools manually when trying to diagnose problems, or when you want to validate the fabric configuration and health.

- Automated

  Run `ethallanalysis` or a specific tool in an automated script (such as a `cron` job). When run in this mode, the `-s` option may prove useful, but care must be taken to avoid excessive saved failures. When run in automated mode, Intel recommends you use a frequency of no faster than hourly. For many fabrics, a daily run or perhaps every few hours is sufficient. Because the exit code from each of the tools indicates the overall success/failure, an automated script can easily check the exit status. If failure occurs, an e-mail of the output can be sent from the analysis tool to the appropriate administrators for further analysis and corrective action.

NOTE

Running these tools too often can have negative impacts. Among the potential risks:

- Each run adds a potential burden to the fabric and switches. For infrequent runs (hourly or daily), the impact is negligible. However, if this were to be run frequently, the impacts to fabric performance can be noticeable.
- Runs with the `-s` option consume additional disk space for each run that identifies an error. The amount of disk space varies depending on fabric size. For a larger fabric, this can be on the order of 1-40 MB. Therefore, care must be taken not to run the tools too often and to visit and clean out the `FF_ANALYSIS_DIR` periodically. If the `-s` option is used during automated execution of the health check tools, it may be helpful to also schedule automated disk space checks (for example, as a `cron` job).
- Runs coinciding with downtime for selected components (such as servers that are offline or rebooting) are considered failures and generate the resulting failure information. If the runs are not carefully scheduled, this data could be misleading and also waste disk space.

5.2.6 Re-Establishing Health Check Baseline

Intel recommends you establish a baseline after you change the fabric configuration. The following activities are examples of ways in which the fabric configuration may be changed:

- Repair a faulty board, which leads to a new serial number for that component.
- Update switch firmware.
- Change time zones in a switch.
- Add or delete a new device or link to a fabric.
- Remove a failed link and its devices from the fabric manager database.

Perform the following procedure to re-establish the health check baseline:

1. Make sure that you have fixed all problems with the fabric, including inadvertent configuration changes, before proceeding.
2. Verify that the fabric configured is as expected. The simplest way to do this is to run `ethfabricinfo`, which returns information for each subnet to which the fabric management server is connected. The following is an example output for a single subnet.

```
# ethfabricinfo
Getting All Fabric Records...
Done Getting All Fabric
Records
Number of NICs: 7
Number of Switches: 1
Number of Links: 7
Number of NIC Links: 7              (Internal: 0   External: 7)
Number of ISLs Links: 0             (Internal: 0   External: 0)
Number of Slow Links: 0             (NIC Links: 0   ISLs: 0)
Number of Omitted Links: 0          (NIC Links: 0   ISLs: 0)
```

3. Save the old baseline because it may be required for future debug. The old baseline is a group of files in `/var/usr/lib/eth-tools/analysis/baseline`.
4. Run `ethallanalysis -b`.
5. Check the new output files in `/var/usr/lib/analysis/baseline` to verify that the configuration is as you expect it.

5.2.7 Interpreting the Health Check Results

When any of the health check tools are run, the overall success or failure is indicated in the output of the tool and its exit status. The tool also indicates which areas had problems and which files should be reviewed. The results from the latest run can be found in `FF_ANALYSIS_DIR/latest/`. This directory includes the latest configuration of the fabric and any errors/differences found during the health check.

If the `-s` option was used when running the health check, a directory whose name is the date and time of the failing run is created under `FF_ANALYSIS_DIR`. In this case, refer to that directory instead of the `latest` directory shown in the following examples.

For fabric, review the results of the fabric analysis for each configured fabric. If nodes or links are missing, the fabric analysis detects them. Missing links or nodes can cause other health checks to fail. If such failures are expected (for example, a node or switch is offline), you can perform further review of result files. However, be aware that the loss of the node or link can cause other analysis to also fail.

The following discussion presents the analysis order for `fabric.<plane_name>`. If other or additional fabrics are configured for analysis, review the files in the order shown for each fabric. There is no specific order recommended for which fabric to review first.

1. `latest/fabric.<plane_name>.errors.stderr`

   If this file is not empty, it can indicate problems with `ethreport`. This may result in unexpected problems or inaccuracies in the related errors file. Correct problems reported in this file first. Once corrected, rerun the health checks to look for further errors.

2. `latest/fabric.<plane_name>.errors`

   If any links with excessive error rates or incorrect link speeds are reported, correct them. If there are links with errors, be aware that the same links may also be detected in other reports such as the links and comps files.

3. `latest/fabric.<plane_name>.snapshot.stderr`

   If this file is not empty, it can indicate problems with `ethreport`. This may result in unexpected problems or inaccuracies in the related links and comps files. Correct problems reported in this file first. Once corrected, rerun the health checks to look for further errors.

4. `latest/fabric.<plane_name>.links.stderr` and `latest/fabric.<plane_name>.links.changes.stderr`

If these files are not empty, it can indicate problems with `ethreport`, which can result in unexpected problems or inaccuracies in the related links files. Correct problems reported in this file first. Once corrected, rerun the health checks to look for further errors.

5. `latest/fabric.<plane_name>.links.diff` and `latest/fabric.<plane_name>.links.changes`

   These indicate that the links between components in the fabric have changed, been removed/added, or that components in the fabric have disappeared. If both files are available, use the `fabric.<plane_name>.links.changes` file since it has a more concise and precise description of the fabric link changes. Compare the `latest/fabric.<plane_name>.links` file to `baseline/fabric.<plane_name>.links`. If components have disappeared, review the `latest/fabric.<plane_name>.comps.diff` and `latest/fabric.<plane_name>.comps.changes` files. Correct missing nodes and links, if necessary. Once corrected, rerun the health checks to look for further errors. If the change was expected and is permanent, rerun a baseline once all other health check errors have been corrected.

6. `latest/fabric.<plane_name>.comps.stderr` and `latest/fabric.<plane_name>.comps.changes.stderr`

   If these files are not empty, it can indicate problems with `ethreport`, which can result in unexpected problems or inaccuracies in the related comps file. Correct problems reported in these files first. Once corrected, rerun the health checks to look for further errors.

7. `latest/fabric.<plane_name>.comps.diff` and `latest/fabric.<plane_name>.comps.changes`

   These indicate that the components in the fabric have changed. If both files are available, use the `fabric.<plane_name>.comps.changes` file since it has a more concise and precise description of the fabric component changes. Compare the `latest/fabric.<plane_name>.comps` file to `baseline/fabric.<plane_name>.comps`. Correct missing nodes, ports that are down, and port misconfigurations, if necessary. Once corrected, rerun the health checks to look for further errors. If the change was expected and permanent, rerun a baseline once all other health check errors have been corrected.

Related Links

Interpreting Health Check *.changes Files on page 93

5.2.8 Interpreting Health Check *.changes Files

Files with the extension `.changes` summarize what has changed in a configuration based on the queries done by the health check.

This type of file uses the following format:

- [What is being verified]
- [Indication that something is not correct]
- [Items that are not correct and what is incorrect about them]
- [How many items were checked]
- [Total number of incorrect items]

- [Summary of how many items had particular issues]

The following example of `fabric.*.links.changes` only shows links that were "Unexpected". That means that the link was not found in the previous baseline.

```
# cat latest/fabric.plane.links.changes
Links Topology Verification

Links Found with incorrect configuration:
Rate IfAddr             Port PortId          Type Name
100g 0x00006805caa382c0   1  6805caa382c0    NIC   phs1fnivd13u07n3-eth2
<->  0x0000fcbd6762d279  11  Eth11           SW   phs1swivd13u21
Unexpected Link

4 of 4 Fabric Links Checked

Links Expected but Missing, Duplicate in input or Incorrect:
3 of 3 Input Links Checked

Total of 1 Incorrect Links found
0 Missing, 1 Unexpected, 0 Misconnected, 0 Duplicate, 0 Different
------------------------------------------------------------------------
```

The following table summarizes possible issues found in `.changes` files.

Table 9. Possible Issues Found in Health Check .changes Files

Issue	Description and Possible Actions
Missing	This issue indicates an item that is in the baseline but is not in this instance of health check output. This may indicate a broken item or a configuration change that has removed the item from the configuration. If you have intentionally removed this item from the configuration, save the original baseline and rerun the baseline. For example, if you removed a NIC connection, the NIC and the link to it are shown as *Missing* in `fabric.*.links.changes` and `fabric.*.comps.changes` files. If the item is still part of the configuration, check for faulty connections or unintended changes to configuration files on the fabric management server. You should also look for any *Unexpected* or *Different* items that may correspond to this item. In some cases, the configuration of an item has changed in a way that makes it difficult to determine precisely how it has changed.
Unexpected	This issue indicates that an item is in this instance of health check output but it is not in the baseline. This may indicate that an item was broken when the baseline was taken or a configuration change has added the item to the configuration. If you have added this item to the configuration, save the original baseline and rerun the baseline. For example, if you added a NIC connection, it is shown as *Unexpected* in `fabric.*.links.changes` and `fabric.*.comps.changes` files. You should also look for any *Missing* or *Different* items that may correspond to this item. In some cases, the configuration of an item has changed in a way that makes it difficult to determine precisely how it has changed.
Misconnected	This issue only applies to links and indicates that a link is not connected properly. This should be fixed. It is possible to find miswires by examining all of the *Misconnected* links in the fabric. However, you must look at all of the `fabric.*.links.changes` files to find miswires between subnets. You should also look for any *Missing* or *Different* items that may correspond to this item. In some cases, the configuration of an item has changed in a way that makes it difficult to determine precisely how it has changed. Individual links that are *Misconnected* are reported as *Incorrect Link* and are added into the Misconnected summary count.

continued...

Issue	Description and Possible Actions
Duplicate	This issue indicates that an item has a duplicate in the fabric. This situation should be resolved so there is only one instance of any particular item being discovered in the fabric. This error can occur: • If there are changes in the fabric such as the addition of parallel links. • When there are enough changes to the fabric that it is difficult to properly resolve and report all the changes. • When `ethreport` is run with manually-generated topology input files that may have duplicate items or incomplete specifications.
Different	This issue indicates that an item still exists in the current health check but it is different from the baseline configuration. If the configuration has changed purposely since the most recent baseline, and the expected difference is reflected here, save the original baseline and rerun the baseline. If this difference was not intended, you must rectify the difference to prevent future health checks from reporting the same difference from the baseline. You should also look for any *Missing* or *Unexpected* items that may correspond to this item. In some cases, the configuration of an item has changed in a way that makes it difficult to determine precisely how it has changed. Individual items that are *Different* are reported as *Mismatched* or *Inconsistent* and are added into the Different summary count.
Port Attributes Inconsistent	This issue indicates that the attributes of a port on one side of a link have changed, such as MgmtIfAddr, Port Number, Device Type, or others. The inconsistency is caused by connecting a different type of device or a different instance of the same device type. This may also occur after replacing a faulty device. If the configuration has changed purposely since the most recent baseline, and this difference is reflected here, save the original baseline and rerun the baseline. If a faulty device was replaced, it is important to re-establish the baseline. If this difference was not intended, you must rectify the difference to prevent future health checks from reporting the same difference from the baseline. This is a specific case of *Different*.
Node Attributes Inconsistent	This issue indicates that the attributes of a node in the fabric have changed, such as IfAddr, Node Description, Device Type, or others. The inconsistency is caused by connecting a different type of device or a different instance of the same device type. This may also occur after replacing a faulty device. If the configuration has changed purposely since the most recent baseline, and this difference is reflected here, save the original baseline and rerun the baseline. If a faulty device was replaced, it is important to re-establish the baseline. If this difference was not intended, you must rectify the difference to prevent future health checks from reporting the same difference from the baseline. This is a specific case of *Different*.
X mismatch: expected … found: ….	This issue indicates an aspect of an item has changed as compared to the baseline configuration. The aspect that changed and the expected and found values are shown. This typically indicates configuration differences such as MTU, Speed, and Node Description. It can also indicate that IfAddr have changed, such as replacing a faulty device with a comparable device. If the configuration has changed purposely since the most recent baseline, and this difference is reflected here, save the original baseline and rerun the baseline. If a faulty device was replaced, it is important to re-establish the baseline. If this difference was not intended, you must rectify the difference to prevent future health checks from reporting the same difference from the baseline.

continued...

Issue	Description and Possible Actions
	This is a specific case of *Different*.
Incorrect Link	This issue only applies to links and indicates that a link is not connected properly. This should be fixed. It is possible to find miswires by examining all of the *Misconnected* links in the fabric. However, you must look at all of the `fabric.*.links.changes` files to find miswires between subnets. You should also look for any *Missing* or *Different* items that may correspond to this item. In some cases, the configuration of an item has changed in a way that makes it difficult to determine precisely how it has changed. This is a specific case of *Misconnected*.

5.3 Verification, Analysis, and Control CLIs

The CLIs described in this section are used for fabric deployment verification, analysis, and control.

5.3.1 ethcabletest

Initiates or stops Cable Bit Error Rate stress tests for Network Interface Card to switch (NIC-SW) links.

Syntax

```
ethcabletest [-p plane] [-f hostfile] [-h 'hosts'] [-n numprocs] [start|stop] ...
```

Options

`--help` Produces full help text.

`-p plane` Specifies the fabric plane the test will run on. The specified plane needs to be defined and enabled in the Mgt config file. Default is the first enabled plane.

`-f hostfile` Specifies the file with hosts to include in NIC-SW test. It overrides the HostsFiles defined in Mgt config file for the corresponding plane.

`-h hosts` Specifies the list of hosts to include in NIC-SW test.

`-n numprocs` Number of processes per host for NIC-SW test. Default is 3.

`start` Starts the NIC-SW tests.

`stop` Stops the NIC-SW tests.

The NIC-SW cable test requires that the `FF_MPI_APPS_DIR` is set, and it contains a pre-built copy of the Intel® `mpi_apps` for an appropriate message passing interface (MPI).

Examples

```
ethcabletest -p plane1 start
ethcabletest -f good stop
ethcabletest -h 'arwen elrond' start
HOSTS='arwen elrond' ethcabletest stop
```

Environment Variables

The following environment variables are also used by this command:

`HOSTS`	List of hosts, used if `-h` option not supplied.
`HOSTS_FILE`	File containing list of hosts, used in absence of `-f` and `-h`.
`FABRIC_PLANE`	Name of fabric plane used in absence of `-p`, `-f`, and `-h`.
`FF_MAX_PARALLEL`	Maximum concurrent operations.

5.3.2 ethextractbadlinks

Produces a CSV file listing all or some of the links that exceed `ethreport -o error` thresholds. `ethextractbadlinks` is a front end to the `ethreport` tool. The output from this tool can be imported into a spreadsheet or parsed by other scripts. This script can be used as a sample for creating custom per link reports.

Syntax

```
ethextractbadlinks [ethreport options]
```

Options

`--help`	Produces full help text.
`ethreport` *`options`*	The following options are passed to `ethreport`. This subset is considered typical and useful for this command. By design, the tool ignores `-o/--output` report option.

`-X/--infile` *`snapshot_input`*	Generates a report using the data in the `snapshot_input` file. `snapshot_input` must have been generated during a previous `-o` *`snapshot`* run. `'-'` may be used as the `snapshot_input` to specify `stdin`.
`-T/--topology` *`topology_input`*	Uses `topology_input` file to augment and verify fabric information. When used, various reports can be augmented with information not available electronically. `'-'` may be used to specify `stdin`.

-c/--config *file*	Specifies the error thresholds configuration file. Default is /etc/eth-tools/ethmon.conf file.
-E/--eth *config_file*	Specifies the Ethernet management configuration file. Default is /etc/eth-tools/mgt_config.xml file.
-p *plane*	Name of the enabled plane defined in Mgt config file. Default is the first enabled plane.
-L/--limit	Limits operation to exact specified focus with -F for port error counters check (-o *errors*). Normally, the neighbor of each selected port is also checked. Does not affect other reports.
-F/--focus *point*	Specifies the focus area for report. Used to limit the scope of report. Refer to Point Syntax on page 111 for details.

Examples

```
# List all the bad links in the fabric:
ethextractbadlinks

# List all the bad links to a switch named "coresw1":
ethextractbadlinks -F "node:coresw1"

# List all the bad links to end-nodes:
ethextractbadlinks -F "nodetype:NIC"
```

5.3.3 ethextractlink

Produces a CSV file listing all or some of the links in the fabric. ethextractlink is a front end to the ethreport tool. The output from this tool can be imported into a spreadsheet or parsed by other scripts. This script can be used as a sample for creating custom per link reports.

Syntax

```
ethextractlink [ethreport options]
```

Options

--help	Produces full help text.
ethreport *options*	The following options are passed to ethreport. This subset is considered typical and useful for this command. By design, the tool ignores -o/--output report option.

`-X/--infile` *`snapshot_input`*	Generates a report using the data in the `snapshot_input` file. `snapshot_input` must have been generated during a previous `-o` *`snapshot`* run. `'-'` may be used as the `snapshot_input` to specify `stdin`.
`-T/--topology` *`topology_input`*	Uses `topology_input` file to augment and verify fabric information. When used, various reports can be augmented with information not available electronically. `'-'` may be used to specify `stdin`.
`-E/--eth` *`config_file`*	Specifies the Ethernet management configuration file. Default is `/etc/eth-tools/mgt_config.xml` file.
`-p` *`plane`*	Name of the enabled plane defined in Mgt config file. Default is the first enabled plane.
`-F/--focus` *`point`*	Specifies the focus area for report. Used to limit scope of report. Refer to Point Syntax on page 111 for details.

Examples

```
# List all the links in the fabric:
ethextractlink

# List all the links to a switch named "coresw1":
ethextractlink -F "node:coresw1"

# List all the links to end-nodes:
ethextractlink -F "nodetype:NIC"
```

5.3.4 ethextractmissinglinks

Produces a CSV file listing all or some of the links in the fabric. `ethextractmissinglinks` is a front end to the `ethreport` tool that generates a report listing all or some of the links that are present in the supplied topology file, but are missing in the fabric. The output from this tool can be imported into a spreadsheet or parsed by other scripts.

Syntax

```
ethextractmissinglinks [-T topology_input] [-o report] [ethreport options]
```

Options

`--help`	Produces full help text.

`-T/--topology` *`topology_input`*	Specifies the topology file to verify against. Default is `/etc/eth-tools/topology.xml`
`-o/--output` *`report`*	Specifies the report type for output. Default is `verifylinks` report. Refer to Report Types for details.
`ethreport` *`options`*	The following options are passed to `ethreport`. This subset is considered typical and useful for this command.

`-X/--infile` *`snapshot_input`*	Generates a report using the data in the `snapshot_input` file. `snapshot_input` must have been generated during a previous `-o` *`snapshot`* run. `'-'` may be used as the `snapshot_input` to specify `stdin`.
`-E/--eth` *`config_file`*	Specifies the Ethernet Mgt configuration file. Default is `/etc/eth-tools/mgt_config.xml` file.
`-p` *`plane`*	Name of the enabled plane defined in Mgt config file, default is the first enabled plane.
`-F/--focus` *`point`*	Specifies the focus area for report. Used to limit scope of report. Refer to Point Syntax on page 111 for details.

Report Types

`verifylinks`	Compares fabric (or snapshot) links to supplied topology and identifies differences and omissions.
`verifyextlinks`	Compares fabric (or snapshot) links to supplied topology and identifies differences and omissions. Limits analysis to links external to systems.
`verifyniclinks`	Compares fabric (or snapshot) links to supplied topology and identify differences and omissions. Limits analysis to links to NICs.
`verifyislinks`	Compares fabric (or snapshot) links to supplied topology and identify differences and omissions. Limits analysis to inter-switch links.
`verifyextislinks`	Compares fabric (or snapshot) links to supplied topology and identify differences and omissions. Limits analysis to inter-switch links external to systems.

Examples

```
List all the missing links in the fabric:
      ethextractmissinglinks

List all the missing links to a switch named "coresw1":
      ethextractmissinglinks -T topology.plane.xml -F "node:coresw1"

List all the missing connections to end-nodes:
      ethextractmissinglinks -o verifyniclinks

List all the missing links between two switches:
      ethextractmissinglinks -o verifyislinks -T topology.plane.xml
```

5.3.5 ethextractsellinks

Produces a CSV file listing all or some of the links in the fabric. `ethextractsellinks` is a front end to the `ethreport` tool. The output from this tool can be imported into a spreadsheet or parsed by other scripts. This script can be used as a sample for creating custom per link reports.

Syntax

```
ethextractsellinks [ethreport options]
```

Options

`--help` — Produces full help text.

`ethreport` *`options`* — The following options are passed to `ethreport`. This subset is considered typical and useful for this command. By design, the tool ignores `-o/--output` report option.

- `-X/--infile` *`snapshot_input`* — Generates a report using the data in the `snapshot_input` file. `snapshot_input` must have been generated during a previous `-o` *`snapshot`* run. `'-'` may be used as the `snapshot_input` to specify `stdin`.
- `-T/--topology` *`topology_input`* — Uses `topology_input` file to augment and verify fabric information. When used, various reports can be augmented with information not available electronically. `'-'` may be used to specify `stdin`.
- `-E/--eth` *`config_file`* — Specifies the Ethernet management configuration file. Default is `/etc/eth-tools/mgt_config.xml` file.
- `-p` *`plane`* — Name of the enabled plane defined in Mgt config file, Default is the first enabled plane.

`-F/--focus` *`point`*	Specifies the focus area for report. Used to limit scope of report. Refer to Point Syntax on page 111 for details.

Examples

```
# List all the links in the fabric:
ethextractsellinks

# List all the links to a switch named "coresw1":
ethextractsellinks -F "node:coresw1"

# List all the connections to end-nodes:
ethextractsellinks -F "nodetype:NIC"
```

5.3.6 ethextractstat2

Performs a per link error analysis of a fabric and provides augmented information from a `topology_file` including all error counters. The output is in a CSV format suitable for importing into a spreadsheet or parsed by other scripts. `ethextractstat2` is a front end to the `ethreport` and `ethxmlextract` tools. This script can be used as a sample for creating custom reports.

Syntax

```
ethextractstat2 topology_file [ethreport options]
```

Options

`--help`	Produces full help text.
`topology_file`	Specifies topology_file to use.
`ethreport` *`options`*	The following options are passed to `ethreport`. This subset is considered typical and useful for this command. By design, the tool ignores `-o/--output` report option.

`-X/--infile` *`snapshot_input`*	Generates a report using the data in the `snapshot_input` file. `snapshot_input` must have been generated during a previous `-o` *`snapshot`* run. `'-'` may be used as the `snapshot_input` to specify `stdin`.
`-c/--config` *`file`*	Specifies the error thresholds configuration file. Default is `/etc/eth-tools/ethmon.conf` file.

`-E/--eth` *`config_file`*	Specifies the Ethernet management configuration file. Default is `/etc/eth-tools/mgt_config.xml` file.
`-p` *`plane`*	Name of the enabled plane defined in Mgt config file. Default is the first enabled plane.
`-L/--limit`	Limits operation to exact specified focus with `-F` for port error counters check (`-o` *`errors`*). Normally, the neighbor of each selected port is also checked. Does not affect other reports.
`-F/--focus` *`point`*	Specifies the focus area for report. Used to limit scope of report. Refer to Point Syntax on page 111 for details.

The portion of the script that calls `ethreport` and `ethxmlextract` follows:

```
ethreport -x -d 10 -s -o errors -T $@ | ethxmlextract -d \;
-e Rate -e MTU -e Internal -e LinkDetails -e CableLength -e CableLabel
-e CableDetails -e Port.NodeGUID -e Port.PortGUID -e Port.PortNum
-e Port.PortId -e Port.PortType -e Port.NodeDesc -e Port.PortDetails
-e PortXmitData.Value -e PortXmitPkts.Value -e PortRcvData.Value
-e PortRcvPkts.Value -e SymbolErrors.Value -e LinkErrorRecovery.Value
-e LinkDowned.Value -e PortRcvErrors.Value
-e PortRcvRemotePhysicalErrors.Value -e PortRcvSwitchRelayErrors.Value
-e PortXmitConstraintErrors.Value -e PortRcvConstraintErrors.Value
-e LocalLinkIntegrityErrors.Value -e ExcessiveBufferOverrunErrors.Value
```

Examples

```
ethextractstat2 topology_file
ethextractstat2 topology_file -c my_ethmon.conf
```

5.3.7 ethfabricinfo

Provides a brief summary of the components in the fabric.

`ethfabricinfo` can be useful as a quick assessment of the fabric state. It can be run against a known-good fabric to identify its components, and then later run to see if anything has changed about the fabric configuration or state.

For more extensive fabric analysis, use `ethreport`. These tools can be found in the *Intel® Ethernet Fabric Suite FastFabric User Guide*.

Syntax

```
ethfabricinfo [-v] [-q] [-E file] [-p planes] [-f host_files] [-X snapshot_input]
```

Options

`--help`	Produces full help text.
`-v`	Returns verbose output.
`-q`	Disables progress reports.
`-E` *`file`*	Specifies the Ethernet Mgt config file. Default is `/etc/eth-tools/mgt_config.xml`.
`-p` *`planes`*	Fabric planes separated by space. Default is the first enabled plane defined in config file. Value 'ALL' will use all enabled planes.
`-f` *`host_files`*	Hosts files separated by space. It overrides the HostsFiles defined in Mgt config file for the corresponding planes. Value 'DEFAULT' will use the HostFile defined in Mgt config file for the corresponding plane
`-X` *`snapshot_input`*	Generates a report using data in *`snapshot_input`* file.

Example

```
ethfabricinfo
ethfabricinfo -X snapshot.xml
ethfabricinfo -p 'p1 p2' -f 'hosts1 DEFAULT'
```

5.3.8 ethfindgood

Checks for hosts that are able to be pinged, accessed via SSH, and active on the Intel® Ethernet Fabric. Produces a list of good hosts meeting all criteria. Typically used to identify good hosts to undergo further testing and benchmarking during initial cluster staging and startup.

The resulting `good` file lists each good host exactly once and can be used as input to create *`mpi_hosts`* files for running `mpi_apps` and the NIC-SW cable test. The files `alive`, `running`, `active`, `good`, and `bad` are created in the selected directory listing hosts passing each criteria. If a plane name is provided, filename will be `xxx_<plane>`, e.g., `good_plane1`

This command automatically generates the file `FF_RESULT_DIR/punchlist.csv`. This file provides a concise summary of the bad hosts found. This can be imported into Excel directly as a `*.csv` file. Alternatively, it can be cut/pasted into Excel, and the **Data/Text to Columns** toolbar can be used to separate the information into multiple columns at the semicolons.

A sample generated output is:

```
# ethfindgood
3 hosts will be checked
2 hosts are pingable (alive)
2 hosts are ssh'able (running)
2 total hosts have RDMA active on one or more fabrics (active)
```

```
1 hosts are alive, running, active (good)
2 hosts are bad (bad)
Bad hosts have been added to /root/punchlist.csv
# cat /root/punchlist.csv
2015/10/09 14:36:48;phs1fnivd13u07n4;Doesn't ping
2015/10/09 14:36:48;phs1fnivd13u07n4;Can't ssh
2015/10/09 14:36:48;phs1fnivd13u07n3;No active RDMA port
```

For a given run, a line is generated for each failing host. Hosts are reported exactly once for a given run. Therefore, a host that does not ping is NOT listed as `can't ssh` nor `No active RDMA port`. There may be cases where ports could be active for hosts that do not ping. However, the lack of ping often implies there are other fundamental issues, such as PXE boot or inability to access DNS or DHCP to get proper host name and IP address. Therefore, reporting hosts that do not ping is typically of limited value.

Syntax

```
ethfindgood [-R|-A] [-d dir] [-p plane] [-f hostfile] [-h 'hosts'] [-T timelimit]
```

Options

`--help`	Produces full help text.
`-R`	Skips the running test (SSH). Recommended if password-less SSH is not set up.
`-A`	Skips the active test. Recommended if Intel® Ethernet Fabric Suite software or fabric is not up.
`-p` *plane*	Specifies the name of the plane to use.
`-d` *dir*	Specifies the directory in which to create alive, active, running, good, and bad files. Default is `/etc/eth-tools` directory.
`-f` *hostfile*	Specifies the file with hosts in cluster. Default is `/etc/eth-tools/hosts` file.
`-h` *hosts*	Specifies the list of hosts to ping.
`-T` *timelimit*	Specifies the time limit in seconds for host to respond to SSH. Default is 20 seconds.

Environment Variables

The following environment variables are also used by this command:

`HOSTS`	List of hosts, used if `-h` option not supplied.
`HOSTS_FILE`	File containing list of hosts, used in absence of `-f` and `-h`.
`FF_MAX_PARALLEL`	Maximum concurrent operations.

Examples

```
ethfindgood
ethfindgood -f allhosts
ethfindgood -h 'arwen elrond'
HOSTS='arwen elrond' ethfindgood
HOSTS_FILE=allhosts ethfindgood
```

5.3.9 ethlinkanalysis

Encapsulates the capabilities for link analysis. Additionally, this tool includes cable and fabric topology verification capabilities. This tool is built on top of `ethreport` (and its analysis capabilities), and accepts the same syntax for input topology and snapshot files.

In addition to being able to run assorted `ethreport` link analysis reports and generate human-readable output, this tool additionally analyzes the results and appends a concise summary of issues found to the `FF_RESULT_DIR/punchlist.csv` file.

Syntax

```
ethlinkanalysis [-U] [-T topology_inputs] [-X snapshot_input] [-x snapshot_suffix]
 [-c file] [-E file] [-p planes] [-f host_files] reports ...
```

Options

`--help`	Produces full help text.
`-U`	Omits unexpected devices and links in `punchlist` file from verify reports.
`-T` *topology_inputs*	Specifies the name of topology input filenames separated by space. See ethreport on page 108 for more information on `topology_input` files.
`-X` *snapshot_input*	Performs analysis using data in *snapshot_input*. *snapshot_input* must have been generated via a previous `ethreport -o snapshot` run.
`-x` *snapshot_suffix*	Creates a snapshot file per selected plane. The files are created in `FF_RESULT_DIR` with names of the form: `snapshotSUFFIX.<plane_name>.xml`.
`-c` *file*	Specifies the error thresholds configuration file. The default is `/etc/eth-tools/ethmon.si.conf`.
`-E` *file*	Ethernet Mgt configuration file. The default is `/etc/eth-tools/mgt_config.xml`.

`-p` *`planes`*	Fabric planes separated by space. The default is the first enabled plane defined in config file. Value 'ALL' will use all enabled planes.
`-f` *`host_files`*	Hosts files separated by space. It overrides the HostsFiles defined in Mgt config file for the corresponding planes. Value 'DEFAULT' will use the HostFile defined in Mgt config file for the corresponding plane.
`reports`	Supports the following reports:

`errors`	Specifies link error analysis.
`slowlinks`	Specifies links running slower than expected.
`misconfiglinks`	Specifies links configured to run slower than supported.
`misconnlinks`	Specifies links connected with mismatched speed potential.
`all`	Includes the reports `errors`, `slowlinks`, `misconfiglinks`, and `misconnlinks`.
`verifylinks`	Verifies links against topology input.
`verifyextlinks`	Verifies links against topology input. Limits analysis to links external to systems.
`verifyniclinks`	Verifies links against topology input. Limits analysis to NIC links.
`verifyislinks`	Verifies links against topology input. Limits analysis to inter-switch links.
`verifyextislinks`	Verifies links against topology input. Limits analysis to inter-switch links external to systems.
`verifynics`	Verifies NICs against topology input.
`verifysws`	Verifies switches against topology input.
`verifynodes`	Verifies NICs and switches against topology input.

`verifyall`	Verifies links, NICs, and switches against topology input.

A punchlist of bad links is also appended to the file: `FF_RESULT_DIR/punchlist.csv`

Examples

```
ethlinkanalysis errors
ethlinkanalysis slowlinks
ethlinkanalysis -p 'p1 p2' -f 'hosts1 DEFAULT' errors
```

5.3.10 ethreport

Provides powerful fabric analysis and reporting capabilities. Must be run on a host connected to the Intel® Ethernet Fabric with the Intel® Ethernet Fabric Suite FastFabric Toolset installed.

Syntax

```
ethreport [-v][-q] [--timeout] [-o report] [-d detail] [-P|-H]
          [-N] [-x] [-X snapshot_input] [-T topology_input] [-s]
          [-A] [-c file] [-L] [-F point] [-Q] [-E file] [-p plane] [-f hostfile]
```

Options

`--help`	Produces full help text.
`-v/--verbose`	Returns verbose output.
`-q/--quiet`	Disables progress reports.
`--timeout`	Specifies the timeout (wait time for response) in ms. Default is 1000 ms.
`-o/--output` *report*	Specifies the report type for output. Refer to Report Types on page 109 for details.
`-d/--detail` *level*	Specifies the level of detail `0-n` for output. Default is 2.
`-P/--persist`	Only include data persistent across reboots.
`-H/--hard`	Only include permanent hardware data.
`-N/--noname`	Omits node.
`-x/--xml`	Produces output in XML.

`-X/--infile` *`snapshot_input`*	Generates a report using the data in the `snapshot_input` file. `snapshot_input` must have been generated during a previous `-o` *`snapshot`* run. `'-'` may be used as the `snapshot_input` to specify `stdin`.
`-T/--topology` *`topology_input`*	Uses `topology_input` file to augment and verify fabric information. When used, various reports can be augmented with information not available electronically. `'-'` may be used to specify `stdin`.
`-s/--stats`	Get performance stats for all ports.
`-A/--allports`	Includes PortInfo for down switch ports.
`-c/--config` *`file`*	Specifies the error thresholds configuration file. Default is `/etc/eth-tools/ethmon.conf` file.
`-E/--ethconfig` *`file`*	Specifies the Ethernet Mgt config file. Default is `/etc/eth-tools/mgt_config.xml` file.
`-p` *`plane`*	Specifies the name of the enabled plane defined in Mgt config file. Default is the first enabled plane.
`-f/--hostfile` *`file`*	Specifies the file with hosts in cluster. It overrides the HostsFile for the selected plane that is defined in Mgt config file.
`-L/--limit`	Limits operation to exact specified focus with `-F` for port error counters check (`-o` *`errors`*). Normally, the neighbor of each selected port is also checked. Does not affect other reports.
`-F/--focus` *`point`*	Specifies the focus area for report. Limits output to reflect a subsection of the fabric. May not work with all reports. (For example, the `verify*` reports may ignore the option or not generate useful results.)
`-Q/--quietfocus`	Excludes focus description from report.

Report Types

`comps`	Summary of all systems in fabric.
`brcomps`	Brief summary of all systems in fabric.
`nodes`	Summary of all node types in fabric.
`brnodes`	Brief summary of all node types in fabric.

`ifids`	Summary of all ifids in the fabric.
`linkinfo`	Summary of all links with ifids in the fabric.
`links`	Summary of all links.
`extlinks`	Summary of links external to systems.
`niclinks`	Summary of links to NICs.
`islinks`	Summary of inter-switch links.
`extislinks`	Summary of inter-switch links external to systems.
`slowlinks`	Summary of links running slower than expected.
`slowconfiglinks`	Summary of links configured to run slower than supported, includes `slowlinks`.
`slowconnlinks`	Summary of links connected with mismatched speed potential, includes `slowconfiglinks`.
`misconfiglinks`	Summary of links configured to run slower than supported.
`misconnlinks`	Summary of links connected with mismatched speed potential.
`errors`	Summary of links whose errors exceed counts in the configuration file.
`otherports`	Summary of ports not connected to this fabric.
`verifynics`	Compares fabric (or snapshot) NICs to supplied topology and identifies differences and omissions.
`verifysws`	Compares fabric (or snapshot) switches to supplied topology and identifies differences and omissions.
`verifynodes`	Returns `verifynics` and `verifysws` reports.
`verifylinks`	Compares fabric (or snapshot) links to supplied topology and identifies differences and omissions.
`verifyextlinks`	Compares fabric (or snapshot) links to supplied topology and identifies differences and omissions. Limits analysis to links external to systems.
`verifyniclinks`	Compares fabric (or snapshot) links to supplied topology and identify differences and omissions. Limits analysis to links to NICs.

`verifyislinks`	Compares fabric (or snapshot) links to supplied topology and identify differences and omissions. Limits analysis to inter-switch links.
`verifyextislinks`	Compares fabric (or snapshot) links to supplied topology and identify differences and omissions. Limits analysis to inter-switch links external to systems.
`verifyall`	Returns `verifynics`, `verifysws`, and `verifylinks` reports.
`all`	Returns `comps`, `nodes`, `links`, `extlinks`, `slowconnlinks`, and `errors` reports.
`snapshot`	Outputs snapshot of the fabric state for later use as *`snapshot_input`*. This implies `-x`. May not be combined with other reports. When selected, `-F` and `-N` options are ignored.
`topology`	Outputs the topology of the fabric for later use as *`topology_input`*. This implies `-x`. May not be combined with other reports. Use with detail level 3 or more to get Port element under Node in output xml.
`fabricinfo`	Outputs fabric information.
`none`	Outputs no report.

Point Syntax

`ifid:`*`value`*	*`value`* is numeric ifid.
`ifid:`*`value`*`:node`	*`value`* is numeric ifid, selects `node` with given ifid.
`ifid:`*`value`*`:port:`*`value2`*	*`value`* is numeric ifid of node, *`value2`* is port number.
`ifid:`*`value`*`:portid:`*`value2`*	*`value`* is numeric ifid of node, *`value2`* is port id.
`mgmtifaddr:`*`value`*	*`value`* is numeric port mgmtifaddr.
`ifaddr:`*`value`*	*`value`* is numeric node ifaddr.
`ifaddr:`*`value1`*`:port:`*`value2`*	*`value1`* is numeric node ifaddr, *`value2`* is port number.
`ifaddr:`*`value1`*`:portid:`*`value2`*	*`value1`* is numeric node ifaddr, *`value2`* is port id.
`chassisid:`*`value`*	*`value`* is numeric chassisid.

`chassisid:value1:port:value2`	*value1* is numeric chassisid, *value2* is port number.
`chassisid:value1:portid:value2`	*value1* is numeric chassisid, *value2* is port id.
`node:value`	*value* is node description (node name).
`node:value1:port:value2`	*value1* is node description (node name), *value2* is port number.
`node:value1:portid:value2`	*value1* is node description (node name), *value2* is port id.
`nodepat:value`	*value* is `glob` pattern for node description (node name).
`nodepat:value1:port:value2`	*value1* is the `glob` pattern for the node description (node name), *value2* is port number.
`nodepat:value1:portid:value2`	*value1* is the `glob` pattern for the node description (node name), *value2* is port id.
`nodedetpat:value`	*value* is `glob` pattern for node details.
`nodedetpat:value1:port:value2`	*value1* is the `glob` pattern for the node details, *value2* is port number.
`nodedetpat:value1:portid:value2`	*value1* is the `glob` pattern for the node details, *value2* is port id.
`nodetype:value`	*value* is node type (`SW` or `NIC`).
`nodetype:value1:port:value2`	*value1* is node type (`SW` or `NIC`), *value2* is port number.
`nodetype:value1:portid:value2`	*value1* is node type (`SW` or `NIC`), *value2* is port id.
`rate:value`	*value* is string for rate (`25g, 50g, 75g, 100g`), omits switch mgmt port 0.
`portstate:value`	*value* is a string for state (`up, down, testing, unknown, dormant, notactive`).

`portphysstate:`*`value`*	*`value`* is a string for PHYs state (`other`, `unknown`, `operational`, `standby`, `shutdown`, `reset`).
`mtucap:`*`value`*	*`value`* is MTU size (maximum size 65535), omits switch mgmt port 0.
`linkdetpat:`*`value`*	*`value`* is `glob` pattern for link details.
`portdetpat:`*`value`*	*`value`* is `glob` pattern for port details.
`nodepatfile:`*`FILENAME`*	Specifies the name of file with the list of nodepats or node descriptions.
`nodepairpatfile:`*`FILENAME`*	Specifies the name of file with the list of node pairs, separated by a colon.
`ldr`	Specifies the ports with a non-zero link down reason or neighbor link down reason.
`ldr:`*`value`*	Specifies the ports with a link down reason or neighbor link down reason equal to *`value`*.

Examples

`ethreport` can generate hundreds of different reports. Commonly-generated reports include the following:

```
ethreport -o comps -d 3
ethreport -o errors -o slowlinks
ethreport -o nodes -F mgmtifaddr:0x00066a00a000447b
ethreport -o nodes -F ifaddr:0x001175019800447b:port:1
ethreport -o nodes -F ifaddr:0x001175019800447b
ethreport -o nodes -F 'node:duster-eth2'
ethreport -o nodes -F 'node:duster-eth2:port:1'
ethreport -o nodes -F 'nodepat:d*'
ethreport -o nodes -F 'nodepat:d*:port:1'
ethreport -o nodes -F 'nodedetpat:compute*'
ethreport -o nodes -F 'nodedetpat:compute*:port:1'
ethreport -o nodes -F nodetype:NIC
ethreport -o nodes -F nodetype:NIC:port:1
ethreport -o nodes -F ifid:1
ethreport -o nodes -F ifid:1:node
ethreport -o nodes -F ifid:1:port:2
ethreport -o nodes -F chassisid:0x001175019800447b
ethreport -o nodes -F chassisid:0x001175019800447b:port:1
ethreport -o extlinks -F rate:100g
ethreport -o extlinks -F portstate:up
ethreport -o extlinks -F portphysstate:operational
ethreport -o extlinks -F 'portdetpat:*mgmt*'
ethreport -o links -F mtucap:2048
ethreport -o snapshot > file
ethreport -o topology > topology.xml
ethreport -o errors -X file
```

Related Links

Management Configuration File on page 37

ethreport Detailed Information on page 114

5.3.11 ethreport Detailed Information

This section provides additional information about using `ethreport`.

5.3.11.1 ethreport Basics

`ethreport` can be run with no options at all. In this mode, it provides a brief list of the nodes in the fabric, the `brnodes` report.

A sample of an `ethreport` for a small fabric follows:

```
# ethreport
Getting All Node Records...
Done Getting All Fabric Records
Node Type Brief Summary

4 Connected NICs in Fabric:
IfAddr           Type Name
    Port IfID      PortId         MgmtIfAddr          Speed
0x00006805caa382c0 NIC coyote-ens785f0
      1 0xa86501 6805caa382c0     0x00006805caa382c0   100Gb
0x00006805caa382d0 NIC goblin-ens785f0
      1 0xa86502 6805caa382d0     0x00006805caa382d0   100Gb
0x00006805caa38370 NIC ogre-ens785f0
      1 0xa86504 6805caa38370     0x00006805caa38370   100Gb
0x00006805caa383c8 NIC duster-ens785f0
      1 0xa86503 6805caa383c8     0x00006805caa383c8   100Gb

1 Connected Switches in Fabric:
IfAddr           Type Name
    Port IfID      PortId         MgmtIfAddr          Speed
0x0000fcbd6762d279 SW edge1
      0 0x7f9f6c                 0x0000fcbd6762d279    None
      1          Ethernet1/1                          100Gb
      2          Ethernet2/1                          100Gb
      3          Ethernet3/1                          100Gb
      4          Ethernet4/1                          100Gb
      5          Ethernet5/1                          100Gb
      6          Ethernet6/1                          100Gb
      7          Ethernet7/1                          100Gb
      8          Ethernet8/1                          100Gb
     67          Management1                         <100Gb
```

Each `ethreport` allows for various levels of detail. Increasing detail is shown as further indentation of the additional information. The `-d` option to `ethreport` controls the detail level. The default is `2`. Values from `0-n` are permitted. The maximum detail per report varies, but most have less than five detail levels.

NOTE

Several report types can include port counters if both the counters are available (via the use of the stats flag or input from a snapshot file) and a high enough detail level is used. Usually a detail level between `5` and `8` is high enough to include per-port counters in report outputs. In addition to options already described in `ethreport` on page 108, some reports such as `errors` or the use of flags such as `-F linkqual` already imply the use of `-d 8`.

For example, when the previous report is run at detail level 0, the output is as follows:

```
# ethreport -d 0
Getting All Fabric Records...
Done Getting All Fabric Records
Node Type Brief Summary

4 Connected NICs in Fabric
1 Connected Switches in Fabric
```

A summary of fabric components is shown in the following example. This report is very similar to `ethfabricinfo`. At the next level of detail, the report has more information:

```
# ethreport -d 1
Getting All Fabric Records...
Done Getting All Fabric Records
Node Type Brief Summary

4 Connected NICs in Fabric:
IfAddr           Type Name
0x00006805caa38370 NIC ogre-eth2
0x00006805caa382d0 NIC goblin-eth2
0x00006805caa382c0 NIC coyote-eth2
0x00006805caa383c8 NIC duster-eth2

1 Connected Switches in Fabric:
IfAddr           Type Name
0x0000fcbd6762d279 SW edge1
```

The previous examples were all performed with a single report: the `brnodes` (Brief Nodes) report. This is just one of the many topology reports that `ethreport` can generate.

Other reports summarize the present state of the fabric. Use these reports to analyze the configuration of the fabric and verify that the installation is consistent with the desired design and configuration. These reports include:

`nodes`	Provides a more verbose form of `brnode` that provides much greater levels of detail to drill down into all the details of every node, including all the port state, IOUs/IOCs/Services, and Port counters.
`comps` and `brcomps`	Provides information similar to brnodes and nodes, except the reports are organized around systems. The grouping into systems is based on Chassis IDs for each node. This report presents more complex systems (such as servers with multiple NICs or large switches composed of multiple switch chips).

NOTE

Some devices do not implement the Chassis ID and may report a value of 0. In such a case, `ethreport` treats each component as an independent system.

`links`	Presents all the links in the fabric. The output is very concise and helps to identify the connectivity between nodes in the fabric. This includes both internal (inside a large switch or system) and external ports (cables).
`extlinks`	Lists all the external links in the fabric, for example, those between different systems. This report omits links internal to a single system. Identification of a system is through `ChassisID`.
`ifids`	Provides information similar to `brnodes`, however it is organized and sorted by IfID. The output is very concise and provides a simple cross reference of IfIDs assigned to each NIC and Switch in the fabric.

Additionally, `ethreport` has reports that analyze the operational characteristics of the fabric and identify bottlenecks and faulty components in the fabric. These reports include:

`slowlinks`	Identifies links that are running slower than expected, that pinpoints bad cables or components in the fabric.
`slowconfiglinks`	Extends the `slowlinks` report to also report links that have been configured (typically by software) to run at a speed below their potential.
`slowconnlinks`	Extends on the `slowconfiglinks` report to also report links that are cabled such that one of the ends of the link can never run to its potential.
`misconfiglinks`	Provides information similar to `slowconfiglinks` in that it reports links that have been configured to run below their potential. However, the report does **not** include links that are running slower than expected.
`misconnlinks`	Provides information similar to `slowconnlinks` in that it reports links that have been connected between ports of different speed potential. However, the report does **not** include links that are running slower than expected, nor links that have been configured to run slower than their potential.
`errors`	Performs a single point in time analysis of the port counters for every node and port in the fabric. All the counters are compared against configured thresholds. Defaults are listed in the `ethmon.conf` file. Any link whose counters exceed these thresholds are listed. Depending on the detail level, the exact counter and threshold are reported. This is a powerful way to

identify marginal links in the fabric such as bad or loose cables or damaged components. The `ethmon.si.conf` file can also be used to check for any non-zero values for signal integrity counters.

5.3.11.2 Simple Topology Verification

`ethreport` provides a flexible way to identify changes to the fabric or the appropriate reassembly of the fabric after a move. For example, run `ethreport` after staging and testing the fabric in a remote location before final installation at a customer site.

This type of report can be saved for later comparison to a future report. Since `ethreport` produces simple text reports, standard tools such as `sdiff` (side-by-side diff) can be used for comparison and analysis of the changes.

In this mode of operation, all previous reports are available, however, you can filter the information that is output. Use the `all` report to include all reports of general interest.

Use the `-N` option to omit all the node and IOC names from the report. If changes are anticipated in this area, this option can be used so future differences do not report changes in names.

5.3.11.3 Advanced Topology Verification

You can use the `-T` option for `ethreport` to compare the state of the fabric against a previous state or a user-generated configuration for the fabric.

The XML description used by the `-T` option is the same as the XML format generated by the `-o links` or `-o extlinks` and/or `-o brnodes` reports when they are run with the `-x` option. The `ethreport -o topology` argument is an easy way to generate such a report and is equivalent to specifying all three of these reports.

A simple way to perform topology verification against a previous configuration is to generate the previous topology using a command such as:

```
ethreport -o topology -x > topology.xml
```

Later, the fabric can be compared against that topology using a command such as:

```
ethreport -T topology.xml -o verifyall
```

Unlike simple `diff` comparisons, this method of topology verification performs a more context-sensitive comparison and presents information in terms of links, nodes that are missing, unexpected, or incorrectly configured.

All the other capabilities of `ethreport` are fully available when using a `topology_input` file. For example, `snapshot_input` files can also be used to generate or compare topologies based on previous fabric snapshots. In addition, the `-F` option may be used to focus the analysis.

NOTE

`verify*` reports may still report missing links, nodes outside the scope of the desired focus.

There are multiple variations of advanced topology verification: `verifysws`, `verifylinks`, and `verifyextlinks`. In addition, `verifynodes` and `verifyall` can be used to generate combined reports.

`verifylinks` and `verifyextlinks` perform the same analysis, however, they differ in the scope of the analysis. `verifylinks` checks all links in the fabric. In contrast, `verifyextlinks` performs the following:

- Limits its verification to links outside of a system.
- Does not analyze links between nodes with the same ChassisID, such as within a large Director Chassis.
- Ignores links from the `topology_input` file that specify a non-zero value for the XML tag `<Internal>` within the `<Link>` tag.

The XML format of `topology_input` file is shown in the following example. The example is intended to be brief and omits many links, nodes.

```
<?xml version="1.0" encoding="utf-8" ?>
<Topology>
<LinkSummary>
<Link>
<Rate>25g</Rate>
<MTU>2048</MTU>
<Internal>0</Internal>
<LinkDetails>IO Server Link</LinkDetails>
<Cable>
<CableLength>11m</CableLength>
<CableLabel>S4567</CableLabel>
<CableDetails>cable model 456</CableDetails>
</Cable>
<Port>
<IfAddr>0x001175010020e004</IfAddr>
<PortNum>1</PortNum>
<PortId>75010020e004</PortId>
<NodeDesc>bender-eth2</NodeDesc>
<MgmtIfAddr>0x001175010020e004</MgmtIfAddr>
<NodeType>NIC</NodeType>
<PortDetails>Some info about port</PortDetails>
</Port>
<Port>
<IfAddr>0x0011750107000df6</IfAddr>
<PortNum>7</PortNum>
<PortId>Eth7</PortId>
<NodeDesc>Switch 1234 Leaf 4</NodeDesc>
<NodeType>SW</NodeType>
</Port>
</Link>
<Link>
<Rate>25g</Rate>
<Internal>0</Internal>
<Cable>
</Cable>
<Port>
<IfAddr>0x001175010025a678</IfAddr>
<PortNum>1</PortNum>
<PortId>75010025a678</PortId>
<NodeDesc>mindy2-eth2</NodeDesc>
```

```
<NodeType>NIC</NodeType>
</Port>
<Port>
<IfAddr>0x0011750107000e6d</IfAddr>
<PortNum>4</PortNum>
<PortId>Eth4</PortId>
<NodeDesc>Switch 2345 Leaf 5</NodeDesc>
<NodeType>SW</NodeType>
</Port>
</Link>
</LinkSummary>
<Nodes>
<NICs>
<Node>
<IfAddr>0x0002c9020020e004</IfAddr>
<NodeDesc>bender-eth2</NodeDesc>
<NodeDetails>More details about node</NodeDetails>
</Node>
<Node>
<IfAddr>0x0002c9020025a678</IfAddr>
<NodeDesc>mindy2-eth2</NodeDesc>
<NodeDetails>Node details</NodeDetails>
</Node>
</NICs>
<Switches>
<Node>
<IfAddr>0x0011750107000df6</IfAddr>
<NodeDesc>Switch 1234 Leaf 4</NodeDesc>
</Node>
<Node>
<IfAddr>0x0011750107000e6d</IfAddr>
<NodeDesc>Switch 2345 Leaf 5</NodeDesc>
</Node>
</Switches>
</Nodes>
</Topology>
```

The XML tags have the following meanings:

`<Report>` Primary top-level tag. Exactly one such tag is permitted per file. Alternatively, this may be `<Topology>`.

`<LinkSummary>` Container tag describing all the links expected in the fabric. Alternatively, `<ExternalLinkSummary>` may be used. `<ExternalLinkSummary>` should be used if the file only describes external links. If both external and internal links are described, `<LinkSummary>` should be used. Only one of these two choices is permitted per file.

`<Link>` Container tag describing a single link. Many instances of this tag can occur per `<LinkSummary>` or `<ExternalLinkSummary>`.

`<Link>` allows the following tags:

`<Rate>` String describing the expected rate of the link. The value is case-insensitive but must contain no extra whitespace. Alternatively, an integer value `<Rate_Int>` may be provided based on the values for Rate. If both `<Rate>` and `<Rate_Int>` are specified, whichever value

appears later within the given link is used. If neither is specified, the rate of the link is not verified.

`<MTU>` An integer describing the expected MTU of the link. If not specified, the MTU of the link is not verified.

`<Internal>` A flag indicating if the link is internal or external. A value of 0 indicates external links that are processed by both verifylinks and verifyextlinks. A value of 1 indicates an internal link that is only processed by verifylinks. If omitted, the actual fabric link attributes or the attributes of the port are used to determine if the link should be processed. The value for this field is not verified against the actual fabric.

`<LinkDetails>` A free-form text field of up to 64 characters. This field is optional. When provided, this is output as a link attribute in all reports that show link details, such as links, extlinks, route, verifylinks, and verifyextlinks reports. Intel recommends you use this field to describe the purpose of the link. This field can also be used by the linkdetpat focus option to select the link.

`<Cable>` A container tag providing additional information about the cable.

`<Cable>` allows the following tags:

`<CableLength>` A free-form text field up to 10 characters. This field is optional. When provided, this is output as a link cable attribute in all reports that show link details, such as links, extlinks, verifylinks, and verifyextlinks reports. Intel recommends you use this field to describe the length of the cable using text such as `11m`. This field can also be used by the lengthpat focus option to select the link.

`<CableLabel>` A free-form text field up to 20 characters. This field is optional. When provided, this is output as a link cable attribute in all reports that show link details, such as links, extlinks, verifylinks, and verifyextlinks reports. Intel recommends you use this field to describe the identifying label attached to the cable using text such as `S4576`. Using this field to match the actual unique physical labels placed on the cables during installation can greatly help cross-

		referencing the reports to the physical cluster, such as when needing to identify or replace cables.
	<CableDetails>	A free-form text field of up to 64 characters. This field is optional. When provided, this is output as a link attribute in all reports that show link details, such as links, extlinks, verifylinks, and verifyextlinks reports. Intel recommends you use this field to describe the type, model, and/or manufacturer of the cable.
<Port>		A container tag providing additional information about the two ports that make up the link. <Port> allows the following tags:
	<IfAddr>	IfAddr reported by the given NIC or switch, such as Mac address.
	<MgmtIfAddr>	IfAddr of management interface for the given NIC or switch.
	<PortNum>	Port Number within the NIC or switch.
	<PortId>	Port ID within the NIC or switch.
	<NodeDesc>	Node Description reported by the NIC or switch. Intel recommends that you configure a unique value for this field in each node in your fabric.
	<NodeType>	Node type reported by the node. Values include: NIC or SW. Alternatively, an integer value <NodeType_Int> may be provided based on the values for NodeType. If both <NodeType> and <NodeType_Int> are specified, whichever appears later within the given Port is used. If neither is specified, the node type of the port is not verified.
	<PortDetails>	Free-form text field of up to 64 characters. This field is optional. When provided, this is output as a port attribute in all reports that show port details, such as links, extlinks, comps, verifylinks, and verifyextlinks reports. Intel recommends you use this field to describe the purpose of the port. This field can also be used by the portdetpat focus option to select the port.

The previous fields are used to associate a port in the `topology_input` file with an actual port in the fabric, also called *resolving the port*. You need not provide all of the information. Association to an actual port in the fabric is performed using the following order of checks based on the tags that are specified:

- `IfAddr`, `PortNum`
- `IfAddr`, `PortId`
- `IfAddr`, `MgmtIfAddr`
- `IfAddr` – If given, NIC has exactly 1 port.
- `NodeDesc`, `PortNum`
- `NodeDesc`, `PortId`
- `NodeDesc`, `MgmtIfAddr`
- `NodeDesc` – If given, NIC has exactly 1 port.
- `MgmtIfAddr`, `PortNum` – Useful to select ports other than 0 on a switch.
- `MgmtIfAddr`, `PortId` – Useful to select ports other than 0 on a switch.
- `MgmtIfAddr`

If `NodeDesc` is used to specify ports, it is important that the fabric is configured such that each `NodeDesc` is unique. Otherwise, the `<Port>` may resolve to a different port than desired, which could result in incorrect results or errors during topology verification.

When redundant information is provided, the extra information is ignored while resolving the port. However, during `verifylinks` or `verifyextlinks` all the input provided is verified against the actual fabric and any discrepancies are reported.

Some examples of redundant information:

- `IfAddr`, `NodeDesc` – `NodeDesc` is not used to resolve port.
- `PortNum`, `PortId` – `PortId` is not used to resolve port.
- `IfAddr`, `PortNum`, `MgmtIfAddr` – `MgmtIfAddr` is not used to resolve port.
- `NodeDesc`, `PortNum`, `MgmtIfAddr` – `MgmtIfAddr` is not used to resolve port.

The `<NodeType>` field is never used during resolution; it is only used during verification.

`<Nodes>` Container tag describing all the nodes expected in the fabric.

`<NICs>` Container tag describing all the NICs expected in the fabric. Many instances of this tag can occur per `<Nodes>`.

<Switches>	Container tag describing all the Switches expected in the fabric. Many instances of this tag can occur per <Nodes>.
<Node>	Container tag describing a single node (NIC or Switch). Many instances of this tag can occur per <NICs> or <Switches>.

<Node> allows the following tags:

<IfAddr>	IfAddr reported by the given NIC or Switch.
<NodeDesc>	Node Description reported by the NIC or Switch. Intel recommends that you configure a unique value for this field in each node in your fabric.
<NodeDetails>	Free form text field of up to 64 characters. This field is optional. When provided, this is output as a node attribute in all reports that show node details, such as links, extlinks, comps, verifysws, verifylinks, and verifyextlinks reports. Intel recommends you use this field to describe the purpose and/or model of the node. This field can also be used by the nodedetpat focus option to select the node.

The previous fields are used to associate a Node (NIC or Switch) in the topology_input file with an actual node in the fabric, also called resolving the node. You need not provide all of the information. Association to an actual node in the fabric is performed using the following order of checks based on the tags that are specified:

- IfAddr
- NodeDesc

If NodeDesc is used to specify nodes, the fabric must be configured such that each NodeDesc is unique. Otherwise, the <Node> may resolve to a different node than desired, which could result in incorrect results or errors during topology verification.

When redundant information is provided, the extra information is ignored while resolving the node. However, during verifysws, all the input provided is verified against the actual fabric and any discrepancies are reported.

An example of redundant information:

- IfAddr, NodeDesc - NodeDesc is not used to resolve node.

The node type (as implied by the container tag for the <Node>) is never used during resolution, it is only used during verification.

5.3.11.4 Augmented Report Information

A `topology_input` file includes additional information including cable (length, label, details), links (details), ports (details), and nodes (details). The file can be used during any report to provide information about the fabric that is not electronically available. This can help cross-reference the output of the report against the physical fabric. For example, if the cable length field is supplied, reports can be focused on all cables of a given length. Similarly, if cable labels are supplied, the report output includes the labels, making it much easier to locate the actual cables for tasks such as rerouting or replacement.

5.3.11.5 Focused Reports

One of the more powerful features of `ethreport` is the ability to focus a report on a subset of the fabric. Using the `-F` option, you can specify a node name, node name pattern, IfAddr, node type, MgmtIfAddr, chassis ID, port state, port physical state, MTU capability, or IfID.

The subsequent report indicates the total components in the fabric, but only reports on those that relate to the focus area. For example, in a nodes report, if a port is specified for focus, only the node containing that port is reported. In a links report, if a port is specified for focus, only the link using that port is reported.

When a focus is used for fabric analysis, `-o errors`, the analysis includes all the ports selected by the focus as well as their neighbors. If desired, the `-L` option limits the operation to exactly the selected ports.

You may choose a focus level that is different from the orientation of the report. For example, if a node name is specified as the focus for a links report, a report of all the links to that node is provided. This includes multiple switch ports or NIC ports.

You can perform reverse lookups by carefully using this feature of report focus. For example, requesting a `brnodes` report with a focus on a IfID performs reverse lookup on that IfID and indicates what node it is for.

When focusing a report, you can also specify a detail level. For detail `0`, the report shows only a count of number of matches. For detail `1`, the report shows only the highest level of the entity that matches.

5.3.11.6 Advanced Focus

As mentioned previously, you can focus a report on a subset of the fabric. In addition, you can further limit the report focus using the following methods.

The beginning of a focused report includes a summary of the items focused on. When the focus has a large scope, this list can be quite long. To omit the summary section from the report, use the `-Q` option.

- Port number specifier

  The node name, node name pattern, IfAddr, node type, and chassis ID also allow for a port number specifier. This limits the focus to the given port number. If the selection resolves to multiple switches or NICs, all ports on the present fabric matching the given port number are selected for the report. For example, in a system composed of multiple nodes, there may be multiple ports with the same port number.

- Port ID specifier

  The node name, node name pattern, IfAddr, node type, and chassis ID also allow for a port id specifier. This limits the focus to the given port id. If the selection resolves to multiple switches or NICs, all ports on the present fabric matching the given port id are selected for the report. For example, in a system composed of multiple nodes, there may be multiple ports with the same port id.

- Glob-style patterns

  You can use a wildcard focus for the node name, node details, link details, or port details. If a consistent naming convention is used for fabric components, this method provides a powerful way to focus reports on nodes. If the host names are prefixed with an indication of their purpose, searches can be performed based on the purpose of the node.

  For example, if you use a naming convention such as the following: `l###` = login node ###, `n###` = compute node ###, `s###` = storage node ###, then you can create a report using one of the following patterns: `'l*'`, `'n*'`, or `'s*'`.

  NOTE

  A glob-style pattern is a shell-style wildcard pattern as used by `bash` and other tools. If you use this style of pattern, you must also use single quotes so the shell does not try to expand them to match local file names.

5.3.11.7 Focus Examples

Examples of using the focus options are shown in the following list:

```
ethreport -o nodes -F mgmtifaddr:0x00117500a000447b
ethreport -o nodes -F ifaddr:0x001175009800447b:port:1
ethreport -o nodes -F ifaddr:0x001175009800447b
ethreport -o nodes -F node:duster
ethreport -o nodes -F node:duster:port:1
ethreport -o nodes -F 'nodepat:d*'
ethreport -o nodes -F 'nodepat:d*:port:1'
ethreport -o nodes -F nodetype:NIC
ethreport -o nodes -F nodetype:NIC:port:1
ethreport -o nodes -F ifid:1
ethreport -o nodes -F ifid:1:node
ethreport -o nodes -F chassisid:0x001175009800447b
ethreport -o nodes -F chassisid:0x001175009800447b:port:1
ethreport -o nodes -F chassisid:0x001175009800447b:portid:Eth4
```

5.3.11.8 Scriptable Output

`ethreport` permits custom scripting. As previously mentioned, options like `-N` generate reports that can be compared to each other. The `-x` option permits output reports to be generated in XML format. The XML hierarchy is similar to the text-based reports. Using XML permits other XML tools (such as PERL XML extensions) to easily parse `ethreport` output, enabling you to create scripts to further search and refine report output formats.

The `ethxmlextract` tool easily converts between XML files and delimited text files.

You can integrate `ethreport` into custom scripts. You can also generate customer-specific, new report formats and cross-reference `ethreport` with other site-specific information.

Related Links

ethxmlextract on page 172

5.3.11.9 Monitor for Fabric Changes Using ethreport

`ethreport` can easily be used in other scripts. For example, the following simple script can be run as a `cron` job to identify if the fabric has changed from the initial design.

```
#!/bin/bash
# specify some filenames to use
expected_config=/usr/local/report.master # master copy of config previously
created
config=/tmp/report$$ # where we will generate new report
diffs=/tmp/report.diff$$ # where we will generate diffs

ethreport -o all -d 5 > $config 2>/dev/null
if ! diff $config $expected_config > $diffs 2>/dev/null
then
# notify admin, for example mail the new report to the admin
 cat $diffs $expected_config $config |
mail -s "fabric change detected" admin@somewhere
fi
rm -f $config $diffs
```

5.3.11.10 Sample Outputs

Analyze all ports in fabric for errors, inconsistent connections, bad cables

```
[root@duster root]# ethreport -o errors -o slowlinks
Links running slower than expected Summary

Links running slower than expected:
2 of 2 Links Checked, 0 Errors found
-------------------------------------------------------------------------------
Links with errors > threshold Summary

Configured Thresholds:
    Dot3HCStatsInternalMacTransmitErrors 1
    Dot3HCStatsInternalMacReceiveErrors 1
    Dot3HCStatsSymbolErrors         1
    IfOutErrors                     1
    IfInErrors                      1
    IfInUnknownProtos               1
    Dot3HCStatsAlignmentErrors      1
    Dot3HCStatsFCSErrors            1
    Dot3HCStatsFrameTooLongs        1
    IfOutDiscards                   1
    IfInDiscards                    1
    Dot3StatsCarrierSenseErrors     1
    Dot3StatsSingleCollisionFrames 1
    Dot3StatsMultipleCollisionFrames 1
    Dot3StatsSQETestErrors          1
    Dot3StatsDeferredTransmissions 1
    Dot3StatsLateCollisions         1
    Dot3StatsExcessiveCollisions    1
Rate IfAddr             Port PortId           Type Name
100g 0x000040a6b7190248   1  40a6b7190248     NIC  phwfstl013.ph.intel.com-
ens785f0
<->  0x0000444ca8cbf441   7  Ethernet7/1      SW   aw-arista-7060-01
    Dot3HCStatsInternalMacReceiveErrors: 5 Exceeds Threshold: 1
    IfInErrors: 5 Exceeds Threshold: 1
    IfOutDiscards: 22977378 Exceeds Threshold: 1
```

```
100g 0x000040a6b7190330   1  40a6b7190330     NIC  phwfstl012.ph.intel.com-
ens785f0
<->  0x0000444ca8cbf441   5  Ethernet5/1      SW   aw-arista-7060-01
    IfOutDiscards: 1973173 Exceeds Threshold: 1
2 of 2 Links Checked, 2 Errors found
-------------------------------------------------------------------------------
```

Obtain detailed information about nodes

NOTE

To shorten the length of the output, the following example focuses on only one node.

```
[root@phwtpriv27 ~]$ ethreport -o nodes -F node:"hds1fnc7081-eth2" -d 7
Node Type Summary
Focused on:
    Node: 0x000040a6b7190390 NIC hds1fnc7081-eth2

8 Connected NICs in Fabric:
    Name: hds1fnc7081-eth2
        IfAddr: 0x000040a6b7190390 Type: NIC
        Ports: 1 ChassisID: 0x0000a4bf01554175
        VendorID: 0x0 MfgName:
        HardwareRev:  FirmwareRev:
        DevName:  PartNum:  SerialNum:
    1 Connected Ports:
        PortNum:   1 EndMgmtIfID: 0x00a86504 MgmtIfAddr: 0x000040a6b7190390
LclMgmtIfID: 1
            Neighbor:  Name: hdarei001
                IfAddr: 0x0000444ca8e9ddd5 Type: SW PortNum:   7 PortId:
Ethernet7/1
            LclMgmtIfID: 1   PortState: Up                      PhysState:
Operational
            IsAutoNegEnabled: False
            PortType: (6) ethernetCsmacd       IPAddr IPv4: 192.168.101.4
            IfID:    0x0004
            RespTimeout:    4 us
            MTU Supported: 1500 bytes
            LinkSpeed: Active:   100Gb (100000 mbit/s)  Supported: -
            IPAddr Prim/Sec: :: / 192.168.101.4
            NeighborNodeType: NIC
            NeighborPortNum: 57        NeighborPortId:
            Capability Supported 0x00000039: StationOnly Router WLANAccessPoint
Bridge
            Capability Enabled 0x0001: StationOnly
            Performance: Transmit
                If HC Out Octets
0 MB (19 Octets)
                If HC Out Ucast Pkts                                        10
                If HC Out Multicast Pkts                                     9
            Performance: Receive
                If HC In Octets
0 MB (23 Octets)
                If HC In Ucast Pkts                                         12
                If HC In Multicast Pkts                                     13
            Error: Packet Discards
                If Out Discards                                             17
                If In Discards                                              18
            Errors: Signal Integrity
                Dot3 HC Stats Internal Mac Transmit Errors                  10
                Dot3 HC Stats Internal Mac Receive Errors                   21
                Dot3 HC Stats Symbol Errors                                 16
            Errors: Packet Integrity
                If Out Errors                                               12
                If In Errors                                                 5
                If In Unknown Protos                                         7
```

```
                Dot3 HC Stats Alignment Errors                                  10
                Dot3 HC Stats FCS Errors                                         7
                Dot3 Stats Frame Too Longs                                      17
            Errors: Half-Duplex Detection
                Dot3 Stats Carrier Sense Errors                                 14
                Dot3 Stats Single Collision Frames                              20
                Dot3 Stats Multiple Collision Frames                             5
                Dot3 Stats SQE Test Errors                                       5
                Dot3 Stats Deferred Transmissions                               13
                Dot3 Stats Late Collisions                                      18
                Dot3 Stats Excessive Collisions                                  5
1 Matching NICs Found

1 Connected Switches in Fabric:
0 Matching Switches Found

-------------------------------------------------------------------------------
```

Identify connections and links composing the fabric

```
[goblin1 root@goblin1]# ethreport -o links
Link Summary

2 Links in Fabric:
Rate IfAddr             Port PortId           Type Name
100g 0x000040a6b7190388   1  40a6b7190388     NIC  hds1fnc7061-eth2
<->  0x0000444ca8e9ddd5   7  Ethernet7/1      SW   hdarei001
100g 0x000040a6b7190390   1  40a6b7190390     NIC  hds1fnc7081-eth2
<->  0x0000444ca8e9ddd5   5  Ethernet5/1      SW   hdarei001
-------------------------------------------------------------------------------
```

Reverse lookup

The following example translates a IfID or IfAddr into the information about the node or port represented.

```
[root@duster duster]# ethreport -o nodes -F
ifaddr:0x000040a6b7190388
Node Type Summary
Focused on:
    Node: 0x000040a6b7190388 NIC hds1fnc7061-eth2

8 Connected NICs in Fabric:
    Name: hds1fnc7061-eth2
        IfAddr: 0x000040a6b7190388 Type: NIC
        Ports: 1 ChassisID: 0x0000a4bf015540f0
        VendorID: 0x0 MfgName:
        HardwareRev:  FirmwareRev:
        DevName:  PartNum:  SerialNum:
    1 Connected Ports:
        PortNum:   1 EndMgmtIfID: 0x00a86503 MgmtIfAddr: 0x000040a6b7190388
LclMgmtIfID: 1
            Neighbor:  Name: hdarei001
                IfAddr: 0x0000444ca8e9ddd5 Type: SW PortNum:   7 PortId:
Ethernet7/1
            Speed:   100Gb
1 Matching NICs Found

1 Connected Switches in Fabric:
0 Matching Switches Found

-------------------------------------------------------------------------------
```

Forward lookup

The following example returns information about nodes listed by name.

```
[root@duster root]# ethreport -o nodes -F "node:hds1fnc7061-eth2"
Node Type Summary
Focused on:
    Node: 0x000040a6b7190388 NIC hds1fnc7061-eth2

8 Connected NICs in Fabric:
    Name: hds1fnc7061-eth2
        IfAddr: 0x000040a6b7190388 Type: NIC
        Ports: 1 ChassisID: 0x0000a4bf015540f0
        VendorID: 0x0 MfgName:
        HardwareRev:  FirmwareRev:
        DevName:  PartNum:  SerialNum:
    1 Connected Ports:
        PortNum:   1 EndMgmtIfID: 0x00a86503 MgmtIfAddr: 0x000040a6b7190388
LclMgmtIfID: 1
            Neighbor:  Name: hdarei001
                IfAddr: 0x0000444ca8e9ddd5 Type: SW PortNum:  7 PortId:
Ethernet7/1
            Speed:   100Gb
1 Matching NICs Found

1 Connected Switches in Fabric:
0 Matching Switches Found

-------------------------------------------------------------------------------
```

Generate report for comparison

The following example generates a report so topology verification can be performed against a known-good configuration.

NOTE

To shorten the length of the output, the following example focuses on only one node.

```
[root@phwtpriv27 ~]$ ethreport -o nodes -F "node:hds1fnc7061-eth2" -d 5
Node Type Summary
Focused on:
    Node: 0x000040a6b7190388 NIC hds1fnc7061-eth2

8 Connected NICs in Fabric:
    Name: hds1fnc7061-eth2
        IfAddr: 0x000040a6b7190388 Type: NIC
        Ports: 1 ChassisID: 0x0000a4bf015540f0
        VendorID: 0x0 MfgName:
        HardwareRev:  FirmwareRev:
        DevName:  PartNum:  SerialNum:
    1 Connected Ports:
        PortNum:   1 EndMgmtIfID: 0x00a86503 MgmtIfAddr: 0x000040a6b7190388
LclMgmtIfID: 1
            Neighbor:  Name: hdarei001
                IfAddr: 0x0000444ca8e9ddd5 Type: SW PortNum:   7 PortId:
Ethernet7/1
            LclMgmtIfID: 1   PortState: Up                     PhysState:
Operational
            IsAutoNegEnabled: False
            PortType: (6) ethernetCsmacd       IPAddr IPv4: 192.168.101.3
            IfID:    0x0004
            RespTimeout:    4 us
            MTU Supported: 9000 bytes
            LinkSpeed: Active:   100Gb (100000 mbit/s)  Supported: -
```

```
            IPAddr Prim/Sec: :: / 192.168.101.3
            NeighborNodeType: NIC
            NeighborPortNum: 41        NeighborPortId:
            Capability Supported 0x00000039: StationOnly Router WLANAccessPoint
Bridge
            Capability Enabled 0x0001: StationOnly
            PortStatus:
                If HC Out Octets                                0 MB | If HC Out
Ucast Pkts                            10
                If HC In Octets                                 0 MB | If HC In
Ucast Pkts                             12
1 Matching NICs Found

1 Connected Switches in Fabric:
0 Matching Switches Found

-------------------------------------------------------------------------------
```

5.3.11.11 Snapshots

You can take a *snapshot* of the fabric state for later offline analysis using the `-o snapshot` report. This report generates an XML snapshot of the present fabric status in a format that `ethreport` can parse.

NOTE

Intel recommends that you do **not** develop your own tools against this format because it may change in future versions of `ethreport`.

The snapshot capability can be used to provide powerful analysis capabilities. Multiple reports can be run against the exact same fabric snapshot, which saves time by not requiring the subsequent reports to query the fabric. Also, historic snapshots can be retained for later offline analysis or historical tracking of the fabric.

When a snapshot is generated, no additional `-o` options are allowed during the run and certain `ethreport` options are ignored. These include: `-F` and `-N`. However, `-L` is valid.

After a snapshot has been generated, it may then be used as input to generate many types of `ethreport` reports. To do this, use the `-X snapshot_input` option, where the `snapshot_input` file is the output from a previous `snapshot` run. When using a snapshot as input, the fabric is not accessed and the node running `ethreport` does not need to be attached to the fabric.

The report generated from the snapshot includes port counters **only** if the original snapshot was run with detail level larger than 5. If not, reports such as `-o` errors are not permitted against the snapshot.

If you want to use standard input (`stdin`) for the snapshot file, then specify `-X`. This can be helpful if snapshots are piped through gzip/gunzip to conserve disk space.

5.3.12 ethverifyhosts

Verifies basic node configuration and performance by running `FF_HOSTVERIFY_DIR/hostverify.sh` on all specified hosts.

NOTE

Prior to using `ethverifyhosts`, copy the sample file `/usr/share/eth-tools/samples/hostverify.sh` to `FF_HOSTVERIFY_DIR` and edit it to set the appropriate configuration and performance expectations and select which tests to run by default. On the first run for a given node, use the `-c` option so that `hostverify.sh` gets copied to each node.

`FF_HOSTVERIFY_DIR` defines both the location of `hostverify.sh` and the destination of the `hostverify.res` output file. `FF_HOSTVERIFY_DIR` is configured in the `/etc/eth-tools/ethfastfabric.conf` file.

A summary of results is appended to the `FF_RESULT_DIR/verifyhosts.res` file. A punchlist of failures is also appended to the `FF_RESULT_DIR/punchlist.csv` file. Only failures are shown on stdout.

Syntax

```
ethverifyhosts [-kc] [-f hostfile] [-u upload_file] [-d upload_dir]
[-h hosts] [-T timelimit] [-F filename] [test ...]
```

Options

`--help`	Produces full help text.
`-k`	At start and end of verification, kills any existing hostverify or xhpl jobs on the hosts.
`-c`	Copies `hostverify.sh` to hosts first, useful if you have edited it.
`-f` *hostfile*	Specifies the file with hosts in cluster. Default is `/etc/eth-tools/hosts`.
`-h` *hosts*	Specifies the list of hosts to ping.
`-u` *upload_file*	Specifies the filename to upload `hostverify.res` to after verification to allow backup and review of the detailed results for each node. The default upload destination file is `hostverify.res`. If `-u ''` is specified, no upload occurs.
`-d` *upload_dir*	Specifies the directory to upload result from each host to. Default is `uploads`.
`-T` *timelimit*	Specifies the time limit in seconds for host to complete tests. Default is 300 seconds (5 minutes).

`-F filename`	Specifies the filename of hostverify script to use. Default is `/root/hostverify.sh`.
test	Specifies one or more specific tests to run. See `/usr/share/eth-tools/samples/hostverify.sh` for a list of available tests.

NOTE

Intel® Xeon Phi™ Processors operate in X2Apic Mode, which requires that the Intel® VT for Directed I/O (VT-d) remain enabled. As a result, the vtd test that checks if VT-D is disabled is not applicable.

Examples

```
ethverifyhosts -c
ethverifyhosts -h 'arwen elrond'
HOSTS='arwen elrond' ethverifyhosts
```

Environment Variables

`HOSTS`	List of hosts, used if `-h` option not supplied.
`HOSTS_FILE`	File containing list of hosts, used in absence of `-f` and `-h`.
`UPLOADS_DIR`	Directory to upload to, used in absence of `-d`.
`FF_MAX_PARALLEL`	Maximum concurrent operations.

5.3.13 ethxlattopology

Generates a topology XML file of a cluster using your customized `<topologyfile>.csv` and core switch linksum files, e.g. `linksum_swd06.csv`, and `linksum_swd24.csv` as input. The output topology XML file can be used to bring up and verify the cluster.

Syntax

```
ethxlattopology [-d level] [-v level] [-i level] [-K] [-N] [-f linkfiles]
[-o report] [-p plane] [source [dest]]
```

Options

`--help`	Produces full help text.
`-d level`	Specifies the output detail level. Default is 0. Levels are additive.

By default, the top level is always produced. Switch, rack, and rack group topology files can be added to the output by choosing the appropriate level. If the output at the group or rack level is specified, then group or rack names must be provided in the spreadsheet. Detailed output can be specified in any combination. A directory for each topology XML file is created hierarchically, with group directories (if specified) at the highest level, followed by rack and switch directories (if specified).

`1` Core switch topology files.

`2` Rack topology files.

`4` Rack group topology files.

`-v` *`level`* Specifies the verbose level. Range is 0 – 8. Default is 2. Levels are additive.

`0` No output.

`1` Progress output.

`2` Reserved.

`4` Time stamps.

`8` Reserved.

`-i` *`level`* Specifies the output indent level. Default is 4.

`-K` Specifies DO NOT clean temporary files.

Prevents temporary files in each topology directory from being removed. Temporary files contain CSV formatted lists of links, NICs, and switches used to create a topology XML file. Temporary files are not typically needed after a topology file is created, or can be retained for subsequent inspection or processing.

`-N` Specifies DO NOT generate Port Numbers from Port IDs.

This will introduce slightly poorer topology-loading performance. Useful when it is difficult to generate Port Numbers, such as complicated Port ID formats, or not enough Port IDs to train the Port Number generator.

`-f` *`linkfiles`* Specifies the space separated core switch linksum files.

`-o report` Specifies the report type for output. By default, all the sections are generated.

Report Types:

brnodes	Creates the <Node> section xml for the csv input.
links	Creates the <LinkSummary> section xml for the csv input.
-p plane	Specifies the plane name. Default is plane.
source	Specifies the source csv file. Default is topology.csv.
dest	Specifies the output xml file. Default is topology.xml. The default output file name can be used to specify destination folder.

Description

The ethxlattopology script reads your customized <topologyfile>.csv file from the local directory, and reads the core switch linksum files specified by -f argument. Two sample topology XLSX files, detailed_topology.xlsx and minimal_topology.xlsx, are located in the /usr/share/eth-tools/samples/ directory. You must create your <topologyfile>.csv file by editing one of the sample spreadsheets and saving the Fabric Links tab as a CSV file. Inspect your <topologyfile>.csv file to ensure that each row contains the correct and same number of comma separators. Any extraneous entries in the spreadsheet can cause the CSV output to have extra fields. Do the same thing on the "Internal xxx Links" tab to create your own internal core switch linksum csv files.

Example

```
ethxlattopology -f "/usr/share/eth-tools/samples/linksum_swd06.csv /usr/share/eth-
tools/samples/linksum_swd24.csv" /tmp/detailed_topology.csv
Parsing linksum file: /usr/share/eth-tools/samples/linksum_swd06.csv
Parsing linksum file: /usr/share/eth-tools/samples/linksum_swd24.csv
Parsing /tmp/detailed_topology.csv
Generating links for Core:core1
Generating links for Core:core2
Processing Leaves of partially populated Core:core2
Processing spines of partially populated Core:core2
Generating topology.xml file(s)
Done
```

Both sample files contain examples of links between NIC and Edge SW (rows 4-7), NIC and Core SW (rows 8-11), and Edge SW and Core SW (rows 12-15).

Environment Variables

The following environment variables allow user-specified MTU.

MTU_SW_SW	If set, it overrides default MTU on switch-to-switch links. Default is 10240.
MTU_SW_NIC	If set, it overrides default MTU on switch-to-NIC links. Default is 10240.

Multi-Rail and Multi-Plane Fabrics

NOTE

Planes can also be referred to as *subnets* or *fabrics*.

For Multi-Rail/Multi-Plane fabrics, you have the following options:

- For Multi-Rail fabrics or for a Single Plane fabric with some multi-ported hosts, you can create multiple rows for a host with different Port names, and then follow the standard procedure to generate `<topologyfile>.xml`.
- For a Multi-Plane fabric with identical planes, the tool can be run multiple times on the same `<topologyfile>.csv` modified with different port names.

  For example, if there are two identical fabrics (fabric_1 and fabric_2) connected to a single host with two NICs (eth2 and eth3), the tool can be run twice like this:

  — For fabric_1:

  In `<topologyfile>.csv`, set port name to be `eth2` for hosts.

  — For fabric_2:

  In `<topologyfile>.csv`, set port name to be `eth3` for hosts.
- For a fabric with both Multi-Rail and Multi-Plane segments, you can use a combination of the above techniques to generate the desired `<topologyfile>.xml` file.

Related Links

Sample Files on page 29

Sample Topology Spreadsheet Overview on page 33

5.4 Detailed Fabric Data Gathering

The CLIs described in this section are used for gathering general fabric data for further analysis. Some commands produce text files while others produce files in CSV format that may be imported into Microsoft Excel.

5.4.1 ethbw

`ethbw` reports the total data moved per RDMA NIC over each interval (default of 1 second). The bandwidth reported for each interval is in units of MB (1,000,000 bytes) over the interval. Both transmit (xmt) and receive (rcv) bandwidth counters are monitored. `ethbw` also monitors Intel NICs for any RDMA retransmit or input packet discards, in which case, the xmt or rcv, respectively, is shown as red. The data is gathered via data movement counters in `/sys/class/infiniband`.

The following cases may present the need to improve PFC tuning:

1. Retransmits can represent packet loss or corruption in the network and may indicate opportunities to improve PFC tuning or high bit error rates (BER) on some cables or devices.

2. Input packet discards indicate packets the NIC itself dropped upon receipt. This can represent opportunities to improve PFC tuning but can also be normal for some environments. Retransmits at the remote NICs that are communicating with this NIC are a more powerful indicator of PFC or BER causes for packet loss.

Syntax

```
ethbw [-i seconds] [-d seconds] [nic ... ]
```

Options

`--help`	Produces full help text.
`-i/--interval` *seconds*	Specifies the interval at which bandwidth will be shown. Values of 1-60 allowed. Defaults to 1.
`-d/--duration` *seconds*	Specifies the duration to monitor. Default is `infinite`.
nic	Specifies an RDMA NIC name. If no NICs are specified, all RDMA NICs will be monitored.

Examples

```
ethbw
ethbw irdma1 irdma3
ethbw -i 2 -d 300 irdma1 irdma3
```

5.4.2 ethextracterror

Produces a CSV file listing all or some of the per port errors in the current fabric. `ethextracterror` is a front end to the `ethreport` tool. The output from this tool can be imported into a spreadsheet or parsed by other scripts. This script can be used as a sample for creating custom per port reports.

Syntax

```
ethextracterror [ethreport options]
```

Options

`--help`	Produces full help text.
`ethreport` *options*	The following options are passed to `ethreport`. This subset is considered typical and useful for this command. By design, the tool ignores `-o/--output` report option.

`-X/--infile` *`snapshot_input`*	Generates a report using the data in the `snapshot_input` file. `snapshot_input` must have been generated during a previous `-o` *`snapshot`* run. `'-'` may be used as the `snapshot_input` to specify `stdin`.
`-T/--topology` *`topology_input`*	Uses `topology_input` file to augment and verify fabric information. When used, various reports can be augmented with information not available electronically. `'-'` may be used to specify `stdin`.
`-E/--eth` *`config_file`*	Specifies the Ethernet management configuration file. Default is `/etc/eth-tools/mgt_config.xml` file.
`-p` *`plane`*	Specifies the name of the enabled plane defined in Mgt config file, default is the first enabled plane.
`-F/--focus` *`point`*	Specifies the focus area for report. Used to limit scope of report. Refer to Point Syntax on page 111 for details.

Examples

```
# List all the link errors in the fabric:
ethextracterror

# List all the link errors related to a switch named "coresw1":
ethextracterror -F "node:coresw1"

# List all the link errors for end-nodes:
ethextracterror -F "nodetype:NIC"
```

5.4.3 ethextractifids

Produces a CSV file listing all or some of the ifids in the fabric. `ethextractifids` is a front end to the `ethreport` tool. The output from this tool can be imported into a spreadsheet or parsed by other scripts.

Syntax

```
ethextractifids [ethreport options]
```

Options

`--help`	Produces full help text.

`ethreport` *`options`* — The following options are passed to `ethreport`. This subset is considered typical and useful for this command. By design, the tool ignores `-o/--output` report option.

`-X/--infile` *`snapshot_input`*	Generates a report using the data in the `snapshot_input` file. `snapshot_input` must have been generated during a previous `-o` *`snapshot`* run. `'-'` may be used as the `snapshot_input` to specify `stdin`.
`-E/--eth` *`config_file`*	Specifies the Ethernet management configuration file. Default is `/etc/eth-tools/mgt_config.xml` file.
`-p` *`plane`*	Specifies the name of the enabled plane defined in Mgt config file, default is the first enabled plane.
`-F/--focus` *`point`*	Specifies the focus area for report. Used to limit scope of report. Refer to Point Syntax on page 111 for details.

Examples

```
# List all the ifids in the fabric:
ethextractifids

# List all the ifids of end-nodes:
ethextractifids -F "nodetype:NIC"
```

5.4.4 ethmergeperf2

Merges the output from two `ethextactperf2` runs from the same fabric. Delta counters for matching links will be computed (`before` subtracted from `after`) and a CSV file equivalent to `ethextractperf2`'s output format will be generated suitable for importing into a spreadsheet or parsing by other scripts.

NOTE

The `before.csv` and `after.csv` input files must be generated from the same fabric, with `before.csv` containing counters prior to `after.csv`. Both files must have been generated to contain the *running counters* without any counter clears between `before.csv` and `after.csv`.

Syntax

```
ethmergeperf2 before.csv after.csv
```

Options

`--help` Produces full help text.

`before.csv` Specifies a CSV file previously generated by `ethextractperf2`.

`after.csv` Specifies a CSV file previously generated by `ethextractperf2`.

Examples

```
ethmergeperf2 before.csv after.csv > delta.csv
```

See Also

`ethextractperf2`

5.4.5 ethextractperf

Provides a report of all the per port performance counters in a CSV format suitable for importing into a spreadsheet or parsed by other scripts for further analysis. It does this by generating a detailed `ethreport` component summary report and piping the result to `ethxmlextract`, extracting element values for NodeDesc, Chassis ID, PortNum, and all the performance counters. Extraction is performed only from the Systems portion of the report, which does not contain Neighbor information (the Neighbor portions are suppressed). This script can be used as a sample for creating custom per port reports.

Syntax

```
ethextractperf [ethreport options]
```

Options

`--help` Produces full help text.

`ethreport` *`options`* The following options are passed to `ethreport`. This subset is considered typical and useful for this command. By design, the tool ignores `-o/--output` report option.

- `-X/--infile` *`snapshot_input`* Generates a report using the data in the `snapshot_input` file. `snapshot_input` must have been generated during a previous `-o` *`snapshot`* run. `'-'` may be used as the `snapshot_input` to specify `stdin`.
- `-T/--topology` *`topology_input`* Uses `topology_input` file to augment and verify fabric information. When used, various reports can be augmented with information not available electronically. `'-'` may be used to specify `stdin`.

-E/--eth *config_file*	Specifies the Ethernet management configuration file. Default is /etc/eth-tools/mgt_config.xml file.
-p *plane*	Specifies the name of the enabled plane defined in Mgt config file, default is the first enabled plane.
-F/--focus *point*	Specifies the focus area for report. Used to limit scope of report. Refer to Point Syntax on page 111 for details.

The portion of the script that calls ethreport and ethxmlextract follows:

```
ethreport -o comps -x -d 10 "$@" | /usr/sbin/ethxmlextract -d \; \
-e NodeDesc -e ChassisID -e PortNum -e PortId -e LinkSpeedActive \
-e IfHCOutOctetsMB -e IfHCOutOctets -e IfHCOutUcastPkts \
-e IfHCOutMulticastPkts -e IfHCInOctetsMB -e IfHCInOctets \
-e IfHCInUcastPkts -e IfHCInMulticastPkts \
-e Dot3HCStatsInternalMacTransmitErrors \
-e Dot3HCStatsInternalMacReceiveErrors -e Dot3HCStatsSymbolErrors \
-e IfOutErrors -e IfInErrors -e IfInUnknownProtos \
-e Dot3HCStatsAlignmentErrors -e Dot3HCStatsFCSErrors \
-e Dot3HCStatsFrameTooLongs -e IfOutDiscards -e IfInDiscards \
-e Dot3StatsCarrierSenseErrors -e Dot3StatsSingleCollisionFrames \
-e Dot3StatsMultipleCollisionFrames -e Dot3StatsSQETestErrors \
-e Dot3StatsDeferredTransmissions -e Dot3StatsLateCollisions \
-e Dot3StatsExcessiveCollisions -s Neighbor
```

Example .

```
ethextractperf
```

5.4.6 ethextractperf2

Provides a report of all per link performance counters in a CSV format suitable for importing into a spreadsheet or parsed by other scripts for further analysis. It does this by generating a detailed ethreport component summary report and piping the result to ethxmlextract, extracting element values for NodeDesc, IfAddr, PortNum, neighbor NodeDesc, neighbor IfAddr, neighbor PortNum and all the performance counters. This script can be used as a sample for creating custom per link reports.

Syntax

```
ethextractperf2 [ethreport options]
```

Options

--help	Produces full help text.

`ethreport` *`options`*	The following options are passed to `ethreport`. This subset is considered typical and useful for this command. Do not use the `-o/--output` report option.	
	`-X/--infile` *`snapshot_input`*	Generates a report using the data in the `snapshot_input` file. `snapshot_input` must have been generated during a previous `-o` *`snapshot`* run. `'-'` may be used as the `snapshot_input` to specify `stdin`.
	`-T/--topology` *`topology_input`*	Uses `topology_input` file to augment and verify fabric information. When used, various reports can be augmented with information not available electronically. `'-'` may be used to specify `stdin`.
	`-E/--eth` *`config_file`*	Specifies the Ethernet management configuration file. Default is `/etc/eth-tools/mgt_config.xml` file.
	`-p` *`plane`*	Specifies the name of the enabled plane defined in Mgt config file, default is the first enabled plane.
	`-F/--focus` *`point`*	Specifies the focus area for report. Used to limit scope of report. Refer to Point Syntax on page 111 for details.

Example .

```
ethextractperf2
```

See Also

`ethmergeperf2`

5.4.7 ethextractstat

Performs an error analysis of a fabric and provides augmented information from a `topology_file`. The report provides cable information.

`ethextractstat` generates a detailed `ethreport` errors report that also has a topology file (see ethreport for more information about topology files). The report is piped to `ethxmlextract`, which extracts values for Link, Cable and Port. (The port element names are context-sensitive.) Note that `ethxmlextract` generates two extraction records for each link (one for each port on the link); therefore, `ethextractstat` merges the two records into a single record and removes redundant link and cable information. This script can be used as a sample for creating custom reports.

`ethextractstat` contains a `while read` loop that reads the CSV line-by-line, uses `cut` to remove redundant information, and outputs the data on a common line.

Syntax

```
ethextractstat topology_file [ethreport options]
```

Options

`--help`	Produces full help text.
`topology_file`	Specifies topology_file to use.
`ethreport` *`options`*	The following options are passed to `ethreport`. This subset is considered typical and useful for this command. By design, the tool ignores `-o/--output` report option.

`-X/--infile` *`snapshot_input`*	Generates a report using the data in the `snapshot_input` file. `snapshot_input` must have been generated during a previous `-o` *`snapshot`* run. `'-'` may be used as the `snapshot_input` to specify `stdin`.
`-c/--config` *`file`*	Specifies the error thresholds configuration file. Default is `/etc/eth-tools/ethmon.conf` file.
`-E/--eth` *`config_file`*	Specifies the Ethernet management configuration file. Default is `/etc/eth-tools/mgt_config.xml` file.
`-p` *`plane`*	Specifies the name of the enabled plane defined in Mgt config file, default is the first enabled plane.
`-L/--limit`	Limits operation to exact specified focus with `-F` for port error counters check (`-o` *`errors`*). Normally, the neighbor of each selected port is also checked. Does not affect other reports.
`-F/--focus` *`point`*	Specifies the focus area for report. Used to limit scope of report. Refer to Point Syntax on page 111 for details.

The portion of the script that calls `ethreport` and `ethxmlextract` follows:

```
ethreport -x -Q -d 10 -o errors -T "$@" | ethxmlextract -H -d \; \
-e Link:id -e Rate -e LinkDetails -e CableLength -e CableLabel \
-e CableDetails -e Port.NodeDesc -e Port.PortNum -e Port.PortId
```

Examples

```
ethextractstat topology_file
ethextractstat topology_file -c my_ethmon.conf
```

5.4.8 ethshowallports

Displays basic port state and statistics for all host nodes.

NOTE

`ethreport` is more powerful Intel® Ethernet Fabric Suite FastFabric commands. For general fabric analysis, use `ethreport`with options such as `-o errors` and `-o slowlinks` to perform an efficient analysis of link speeds and errors.

Syntax

```
ethshowallports [-p plane] [-f hostfile] [-h 'hosts']
```

Options

`--help` Produces full help text.

`-p plane` Specifies the name of the plane to use. Default is the first enabled plane defined in Mgt config file.

`-f hostfile` Specifies the file containing the list of hosts in cluster. It overrides the HostsFile for the selected plane that is defined in Mgt config file.

`-h hosts` Specifies the list of hosts for which to show ports.

Environment Variables

The following environment variables are also used by this command:

`HOSTS` List of hosts, used if `-h` option not supplied.

`HOSTS_FILE` File containing list of hosts, used in absence of `-f` and `-h`.

`FABRIC_PLANE` Name of fabric plane used in absence of `-p`, `-f`, and `-h`.

Example

```
ethshowallports
ethshowallports -p plane1
ethshowallports -h 'elrond arwen'
HOSTS='elrond arwen' ethshowallports
```

Notes

When performing `ethshowallports` against hosts, internally SSH is used. The command `ethshowallports` requires that password-less SSH be set up between the host running the Intel® Ethernet Fabric Suite FastFabric Toolset and the hosts `ethshowallports` is operating against. The `ethsetupssh` FastFabric tool can aid in setting up password-less SSH.

Related Links

Selection of Hosts on page 26

5.5 Configuration and Control for Host

The CLIs described in this section are used for general management of hosts in the fabric.

5.5.1 ethhostadmin

Performs a number of multi-step host initialization and verification operations, including upgrading software, rebooting hosts, and other operations. In general, operations performed by `ethhostadmin` involve a login to one or more host systems.

Syntax

```
ethhostadmin [-c] [-e] [-p plane] [-f hostfile] [-h 'hosts'] [-r release]
[-I install_options] [-U upgrade_options] [-d dir] [-T product]
[-P packages] [-S] operation ...
```

Options

`--help`	Produces full help text.
`-c`	Cleans the result files from any previous run before starting this run.
`-e`	Specifies to exit after the first operation that fails.
`-p plane`	Specifies the name of the plane to use. Default is the first enabled plane defined in Mgt config file.
`-f hostfile`	Specifies the file with the names of hosts in a cluster. Default is `/etc/eth-tools/hosts` file.
`-h hosts`	Specifies the list of hosts to execute the operation against.
`-r release`	Specifies the software version to load/upgrade to. Default is the version of Intel® Ethernet Fabric Suite Software presently being run on the server.
`-d dir`	Specifies the directory to retrieve *product*.*release*.tgz for load or upgrade.

`-I install_options` Specifies the software install options.

`-U upgrade_options` Specifies the software upgrade options.

`-T product` Specifies the product type to install. Default is `FF_PRODUCT`. Options include:

- `IntelEth-Basic.<distro>`
- `IntelEth-FS.<distro>`

where `<distro>` is the distribution and CPU, such as RHEL81-x86_64.

`-P packages` Specifies the packages to install. Default is `eth eth_rdma`. Refer to `INSTALL -C` for complete list of packages.

`-S` Securely prompts for user password on remote system.

operation Performs the specified *operation*, which can be one or more of the following:

`load`	Preforms an initial installation of all hosts.
`upgrade`	Upgrades installation of all hosts.
`reboot`	Reboots hosts, ensures they go down and come back.
`rping`	Verifies this host can ping each host through RDMA.
`pfctest`	Verifies PFC works on all hosts.
`mpiperf`	Verifies latency and bandwidth for each host.
`mpiperfdeviation`	Verifies latency and bandwidth for each host against a defined threshold (or relative to average host performance).

Example

```
ethhostadmin -c reboot
ethhostadmin upgrade
ethhostadmin -p plane1 rping
ethhostadmin -h 'elrond arwen' reboot
HOSTS='elrond arwen' ethhostadmin reboot
```

Details

ethhostadmin provides detailed logging of its results. During each run, the following files are produced:

- test.res: Appended with summary results of run.
- test.log: Appended with detailed results of run.
- save_tmp/: Contains a directory per failed test with detailed logs.
- test_tmp*/: Intermediate result files while test is running.

The -c option removes all log files.

Results from ethhostadmin are grouped into test suites, test cases, and test items. A given run of ethhostadmin represents a single test suite. Within a test suite, multiple test cases occur; typically one test case per host being operated on. Some of the more complex operations may have multiple test items per test case. Each test item represents a major step in the overall test case.

Each ethhostadmin run appends to test.res and test.log, and creates temporary files in test_tmp$PID in the current directory. test.res provides an overall summary of operations performed and their results. The same information is also displayed while ethhostadmin is executing. test.log contains detailed information about what was performed, including the specific commands executed and the resulting output. The test_tmp directories contain temporary files that reflect tests in progress (or killed). The logs for any failures are logged in the save_temp directory with a directory per failed test case. If the same test case fails more than once, save_temp retains the information from the first failure. Subsequent runs of ethhostadmin are appended to test.log. Intel recommends reviewing failures and using the -c option to remove old logs before subsequent runs of ethhostadmin.

ethhostadmin implicitly performs its operations in parallel. However, as for the other tools, FF_MAX_PARALLEL can be exported to change the degree of parallelism. 1000 parallel operations is the default.

Environment Variables

The following environment variables are also used by this command:

HOSTS	List of hosts, used if -h option not supplied.
HOSTS_FILE	File containing list of hosts, used in absence of -f and -h.
FABRIC_PLANE	Name of fabric plane used in absence of -p, -f, and -h.
FF_MAX_PARALLEL	Maximum concurrent operations are performed.
FF_SERIALIZE_OUTPUT	Serialize output of parallel operations (yes or no).
FF_TIMEOUT_MULT	Multiplier for all timeouts associated with this command. Used if the systems are slow for some reason.
FF_PRODUCT	Product to install during load and upgrade operations.

ethhostadmin Operation Details

(Host) Intel recommends that you set up password SSH or SCP for use during this operation. Alternatively, the `-S` option can be used to securely prompt for a password, in which case the same password is used for all hosts. Alternately, the password may be put in the environment or the `ethfastfabric.conf` file using `FF_PASSWORD` and `FF_ROOTPASS`.

`load`

Performs an initial installation of Intel® Ethernet Fabric Suite Software on a group of hosts. Any existing installation is uninstalled and existing configuration files are removed. Subsequently, the hosts are installed with a default Intel® Ethernet Fabric Suite Software configuration. The `-I` option can be used to select different install packages. Default is `eth_tools eth_rdma mpi` The `-r` option can be used to specify a release to install other than the one that this host is presently running. The *`FF_PRODUCT`*`.`*`FF_PRODUCT_VERSION`*`.tgz` file (for example, `IntelEth-Basic.`*`version`*`.tgz`) is expected to exist in the directory specified by `-d`. Default is the current working directory. The specified software is copied to all the selected hosts and installed.

`upgrade`

Upgrades all selected hosts without modifying existing configurations. This operation is comparable to the `-U` option when running `./INSTALL` manually. The `-r` option can be used to upgrade to a release different from this host. The default is to upgrade to the same release as this host. The *`FF_PRODUCT`*`.`*`FF_PRODUCT_VERSION`*`.tgz` file (for example, `IntelEth-Basic.`*`version`*`.tgz`) is expected to exist in the directory specified by `-d`. The default is the current working directory. The specified software is copied to all the end nodes and installed.

NOTE

Only components that are currently installed are upgraded. This operation fails for hosts that do not have Intel® Ethernet Fabric Suite Software installed.

`reboot`

Reboots the given hosts and ensures they go down and come back up by pinging them during the reboot process. The ping rate is slow (5 seconds), so if the servers boot faster than this, false failures may be seen.

`rping`

Verifies RDMA basic operation by ensuring that the nodes can ping each other through RDMA. To run this command, Intel® Ethernet Fabric software must be installed, RDMA must be configured and running on the host, and the given hosts, and switches must be up.

`pfctest` Specifies an empirical test that verifies PFC is working right. To run this command, Intel® Ethernet Fabric software must be installed, PFC must be configured on both hosts and switches, and the given hosts and switches must be up.

`mpiperf` Verifies that MPI is operational and checks MPI end-to-end latency and bandwidth between pairs of nodes (for example, 1-2, 3-4, 5-6). Use this to verify switch latency/hops, PCI bandwidth, and overall MPI performance. The `test.res` file contains the results of each pair of nodes tested.

NOTE

This option is available for the Intel® Ethernet Host Software OFA Delta packaging, but is not presently available for other packagings of OFED.

To obtain accurate results, this test should be run at a time when no other stressful applications (for example, MPI jobs or high stress file system operations) are running on the given hosts.

Bandwidth issues typically indicate server configuration issues (for example, incorrect slot used, incorrect BIOS settings, or incorrect NIC model), or fabric issues (for example, symbol errors, incorrect link width, or speed). Assuming `ethreport` has previously been used to check for link errors and link speed issues, the server configuration should be verified.

Note that BIOS settings and differences between server models can account for 10-20% differences in bandwidth. For more details about BIOS settings, consult the documentation from the server supplier and/or the server PCI chipset manufacturer.

`mpiperfdeviation` Specifies the enhanced version of `mpiperf` that verifies MPI performance. Can be used to verify switch latency/hops, PCI bandwidth, and overall MPI performance. It performs assorted pair-wise bandwidth and latency tests, and reports pairs outside an acceptable tolerance range. The tool identifies specific nodes that have problems and provides a concise summary of results. The `test.res` file contains the results of each pair of nodes tested.

By default, concurrent mode is used to quickly analyze the fabric and host performance. Pairs that have 20% less bandwidth or 50% more latency than the average pair are reported as failures.

The tool can be run in a sequential or a concurrent mode. Sequential mode runs each host against a reference host. By default, the reference host is selected based on the best performance from a quick test of the first 40 hosts. In concurrent mode, hosts are paired up and all pairs are run

concurrently. Since there may be fabric contention during such a run, any poor performing pairs are then rerun sequentially against the reference host.

Concurrent mode runs the tests in the shortest amount of time, however, the results could be slightly less accurate due to switch contention. In heavily oversubscribed fabric designs, if concurrent mode is producing unexpectedly low performance, try sequential mode.

NOTE

This option is available for the Intel® Ethernet Host Software OFA Delta packaging, but is not presently available for other packagings of OFED.

To obtain accurate results, this test should be run at a time when no other stressful applications (for example, MPI jobs, high stress file system operations) are running on the given hosts.

Bandwidth issues typically indicate server configuration issues (for example, incorrect slot used, incorrect BIOS settings, or incorrect NIC model), or fabric issues (for example, symbol errors, incorrect link width, or speed). Assuming `ethreport` has previously been used to check for link errors and link speed issues, the server configuration should be verified.

Note that BIOS settings and differences between server models can account for 10-20% differences in bandwidth. A result 5-10% below the average is typically not cause for serious alarm, but may reflect limitations in the server design or the chosen BIOS settings.

For more details about BIOS settings, consult the documentation from the server supplier and/or the server PCI chipset manufacturer.

The deviation application supports a number of parameters that allow for more precise control over the mode, benchmark, and pass/fail criteria. The parameters to use can be selected using the `FF_DEVIATION_ARGS` configuration parameter in `ethfastfabric.conf`

Available parameters for deviation application:

```
[-bwtol bwtol] [-bwdelta MBs] [-bwthres MBs]
[-bwloop count] [-bwsize size] [-lattol latol]
[-latdelta usec] [-latthres usec] [-latloop count]
[-latsize size][-c] [-b] [-v] [-vv]
[-h reference_host]
```

`-bwtol` Specifies the percent of bandwidth degradation allowed below average value.

-bwbidir	Performs a bidirectional bandwidth test.
-bwunidir	Performs a unidirectional bandwidth test (Default).
-bwdelta	Specifies the limit in MB/s of bandwidth degradation allowed below average value.
-bwthres	Specifies the lower limit in MB/s of bandwidth allowed.
-bwloop	Specifies the number of loops to execute each bandwidth test.
-bwsize	Specifies the size of message to use for bandwidth test.
-lattol	Specifies the percent of latency degradation allowed above average value.
-latdelta	Specifies the limit in µsec of latency degradation allowed above average value.
-latthres	Specifies the lower limit in µsec of latency allowed.
-latloop	Specifies the number of loops to execute each latency test.
-latsize	Specifies the size of message to use for latency test.
-c	Runs test pairs concurrently instead of the default of sequential.
-b	When comparing results against tolerance and delta, uses best instead of average.
-v	Specifies the verbose output.
-vv	Specifies the very verbose output.
-h	Specifies the reference host to use for sequential pairing.

Both bwtol and bwdelta must be exceeded to fail bandwidth test.

When bwthres is supplied, bwtol and bwdelta are ignored.

Both `lattol` and `latdelta` must be exceeded to fail latency test.

When `latthres` is supplied, `lattol` and `latdelta` are ignored.

For consistency with OSU benchmarks, MB/s is defined as 1000000 bytes/s.

Related Links

Selection of Hosts on page 26

5.5.2 Interpreting the ethhostadmin log files

Each run of `ethhostadmin` creates `test.log` and `test.res` files in the current directory.

The `test.res` file summarizes which tests have failed and identifies servers that have failed. If the problem is not immediately obvious, check the `test.log` file. The most recent results are at the end of the file. The `save_tmp/*/test.log` files are easier to read since they represent the logs for a single test case, typically against a single host.

The keyword `FAILURE` is used to mark any failures. Due to the roll-up of error messages, the first instance of `FAILURE` in a given sequence shows the operations in process at the time of failure. The log also shows the exact sequence of commands issued to the target host and the resulting output from that host before the `FAILURE` keyword.

If there is a `FAILURE` message indicating timeout, it means the expected output did not occur within a reasonable time limit. The time limits used are generous, so such failures often indicate a host is offline. It could also indicate unexpected prompts, such as a password prompt when password-less SSH is expected. Review the `test.log` first for such prompts. Also verify that the host can SSH to the target host with the expected password behavior.

One common source of timeout errors is incorrect host shell command prompts. Verify that both this host and the target host meet the following criteria for command prompts:

- The command line prompt must end in `#` or `$`.
- There must be a space after either character.

Another common source of timeouts is typographical errors in selected host names. Verify that the host names in the `test.log` file match the intended host names.

5.6 Basic Setup and Administration Tools

The tools described in this section are available on a node that has Intel® Ethernet Fabric Suite installed.

5.6.1 ethpingall

Pings a group of hosts or switches to verify that they are powered on and accessible through TCP/IP ping.

Syntax

```
ethpingall [-C] [-p] [-f hostfile] [-F switchesfile] [-h 'hosts'] [-H 'switches']
```

Options

`--help`	Produces full help text.
`-C`	Performs a ping against switches. Default is hosts.
`-p`	Pings all hosts/switches in parallel.
`-f hostfile`	Specifies the file with hosts in cluster. Default is `/etc/eth-tools/hosts`.
`-F switchesfile`	Specifies the file with switches in cluster. Default is `/etc/eth-tools/switches`.
`-h hosts`	Specifies the list of hosts to ping.
`-H switches`	Specifies the list of switches to ping.

Example

```
ethpingall
ethpingall -h 'arwen elrond'
HOSTS='arwen elrond' ethpingall
ethpingall -C
```

NOTE

This command pings all hosts/switches found in the specified host/switches file. The use of `-C` option selects the default file and/or environment variable to use. For this command, it is valid to use a file that lists both hosts and switches.

```
ethpingall -C -H 'switch1 switch2'
SWITCHES='switch1 switch2' ethpingall -C
```

Environment Variables

`HOSTS`	List of hosts, used if `-h` option not supplied.
`SWITCHES`	List of switches, used if `-H` option not supplied.
`HOSTS_FILE`	File containing list of hosts, used in absence of `-f` and `-h`.

SWITCHES_FILE	File containing list of switches, used in absence of -F and -H.
FF_MAX_PARALLEL	When -p option is used, maximum concurrent operations are performed.

5.6.2 ethsetupssh

Creates SSH keys and configures them on all hosts so the system can use SSH and SCP into all other hosts without a password prompt. Typically, during cluster setup this tool enables the root user on the Management Node to log into the other hosts (as root) using password-less SSH.

Syntax

```
ethsetupssh [-p|U] [-f hostfile] [-h 'hosts'] [-u user] [-S] [-R|P]
```

Options

--help	Produces full help text.
-p	Performs operation against all hosts in parallel.
-U	Performs connect only (to enter in local hosts, known hosts). When run in this mode, the -S option is ignored.
-f *hostfile*	Specifies the file with hosts in cluster. Default is /etc/eth-tools/hosts file.
-h *hosts*	Specifies the list of hosts to set up.
-u *user*	Specifies the user on remote system to allow this user to SSH to. Default is current user code for host(s).
-S	Securely prompts for password for user on remote system.
-R	Skips setup of SSH to local host.
-P	Skips ping of host (for SSH to devices on Internet with ping firewalled).

Examples

```
ethsetupssh -S
ethsetupssh -U
ethsetupssh -h 'arwen elrond' -U
HOSTS='arwen elrond' ethsetupssh -U
```

Environment Variables

The following environment variables are also used by this command:

HOSTS_FILE	File containing list of hosts, used in absence of -f and -h.
HOSTS	List of hosts, used if -h option not supplied.
FF_MAX_PARALLEL	When -p option is used, maximum concurrent operations.

Description

ethsetupssh provides an easy way to create SSH keys and distribute them to the hosts in the cluster. Many of the FastFabric tools (as well as many versions of MPI) require that SSH is set up for password-less operation. Therefore, ethsetupssh is an important setup step.

This tool also sets up SSH to the local host. This capability is required by selected FastFabric Toolset commands and may be used by some applications (such as MPI).

ethsetupssh has two modes of operation. The mode is selected by the presence or absence of the -U option. Typically, ethsetupssh is first run without the -U option, then it may later be run with the -U option.

Host Initial Key Exchange

When run without the -U option, ethsetupssh performs the initial key exchange and enables password-less SSH and SCP. The preferred way to use ethsetupssh for initial key exchange is with the -S option. This requires that all hosts are configured with the same password for the specified "user" (typically root). In this mode, the password is prompted for once and then SSH and SCP are used in conjunction with that password to complete the setup for the hosts. This mode also avoids the need to set up rsh/rcp/rlogin (which can be a security risk).

Refreshing Local Systems Known Hosts

If aspects of the host have changed, such as IP addresses, MAC addresses, software installation, or server OS reinstallation, you can refresh the local host's SSH known_hosts file by running ethsetupssh with the -U option. This option does not transfer the keys, but instead connects to each host to refresh the SSH keys. Existing entries for the specified hosts are replaced within the local known_hosts file. When run in this mode, the -S option is ignored. This mode assumes SSH has previously been set up for the hosts; therefore, no files are transferred to the specified hosts and no passwords should be required.

Related Links

Selection of Hosts on page 26

5.6.3 ethcmdall

(Linux) Executes a command on all hosts. This powerful command can be used for configuring servers, verifying that they are running, starting and stopping host processes, and other tasks.

NOTE

`ethcmdall` depends on the Linux convention that utilities return 0 for success and > 0 for failure. If `ethcmdall` is used to execute a non-standard utility like `diff` or a program that uses custom exit codes, then `ethcmdall` may erroneously report "Command execution FAILED" when it encounters a non-zero exit code. However, the command output is returned normally and the error may be safely ignored.

Syntax

```
ethcmdall [-pqP] [-f hostfile] [-h hosts] [-u user]
[-T timelimit] cmd
```

Options

`--help`	Produces full help text.
`-p`	Runs command in parallel on all hosts.
`-q`	Specifies quiet mode and does not show the command to execute.
`-P`	Outputs the hostname as a prefix to each output line. This can make script processing of the output easier.
`-f hostfile`	Specifies the file with hosts in cluster. Default is `/etc/eth-tools/hosts` file.
`-h host`	Specifies the list of hosts to execute command on.
`-u user`	Specifies the user to perform the command as: • For hosts, the default is current user.
`-T timelimit`	Specifies the time limit in seconds when running host commands. Default is -1 (infinite).

Examples

Operations on Host

```
ethcmdall date
ethcmdall 'uname -a'
ethcmdall -h 'elrond arwen' date
HOSTS='elrond arwen' ethcmdall date
```

Environment Variables

The following environment variables are also used by this command:

`HOSTS`	List of hosts, used if `-h` option not supplied.

HOSTS_FILE	File containing list of hosts, used in absence of -f and -h.
FF_MAX_PARALLEL	When -p option is used, maximum concurrent operations are performed.
FF_SERIALIZE_OUTPUT	Serialize output of parallel operations (yes or no).

Notes

All commands performed with ethcmdall must be non-interactive in nature. ethcmdall waits for the command to complete before proceeding. For example, when running host commands such as rm, the -i option (interactively prompt before removal) should not be used. (Note that this option is sometimes part of a standard bash alias list.) For further information about Linux operating system commands, consult the man pages.

When performing ethcmdall against hosts, SSH is used internally. The command ethcmdall requires that password-less SSH be set up between the host running the Intel® Ethernet Fabric Suite FastFabric Toolset and the hosts ethcmdall is operating against. The ethsetupssh FastFabric tool can aid in setting up password-less SSH.

Related Links

Selection of Hosts on page 26

5.6.4 ethcaptureall

Captures supporting information for a problem report from all hosts and uploads to this system.

When a host ethcaptureall is performed, ethcapture is run to create the specified capture file within ~root on each host (with the .tgz suffix added as needed). The files are uploaded and unpacked into a matching directory name within upload_dir/*hostname*/ on the local system. The default file name is hostcapture.

The uploaded captures are combined into a .tgz file with the file name specified and the suffix .all.tgz added.

Syntax

```
ethcaptureall [-p] [-f hostfile] [-h 'hosts'] [-d upload_dir]
[-D detail_level] [file]
```

Options

--help	Produces full help text.
-p	Performs capture upload in parallel on all hosts.
-f *hostfile*	Specifies the file with hosts in cluster. Default is /etc/eth-tools/hosts file.

`-h hosts` Specifies the list of hosts to capture.

`-d upload_dir` Specifies the directory to upload to. Default is `uploads`. If not specified, the environment variable `UPLOADS_DIR` is used.

`-D detail_level` Specifies the level of detail of the capture passed to the local host `ethcapture`.

`1` (Local)	Obtains local information from each host.
`2` (Fabric)	In addition to *Local*, also obtains basic fabric information using `ethreport`.
`3` (Analysis)	In addition to *Fabric*, also obtains `ethallanalysis` results. If `ethallanalysis` has not yet been run, it is run as part of the capture.

NOTES

- Detail levels 2-3 can be used when fabric operational problems occur. If the problem is node-specific, detail level 1 should be sufficient. Detail levels 2-3 require an operational fabric. Typically, your support representative requests a given detail level. If a given detail level takes excessively long or fails to be gathered, try a lower detail level.
- For detail levels 2-3, the additional information is only gathered on the node running the `ethcaptureall` command.

file Specifies the name for capture file. If the specified name does not end in `.tgz`, the suffix `.tgz` is appended.

Examples

```
ethcaptureall
# Creates  a hostcapture directory in upload_dir/hostname/ for each host in
/etc/eth-tools/hosts file, then creates hostcapture.all.tgz.

ethcaptureall mycapture
# Creates a mycapture directory in upload_dir/hostname/ for  each  host in
/etc/eth-tools/hosts file, then creates mycapture.all.tgz.

ethcaptureall -h 'arwen elrond' 030127capture
# Gets  the  list  of  hosts  from arwen elrond file and creates
030127capture.tgz file.
```

Environment Variables

The following environment variables are also used by this command:

`HOSTS` List of hosts, used if `-h` option not supplied.

HOSTS_FILE	File containing a list of hosts, used in the absence of -f and -h.
UPLOADS_DIR	Directory to upload to, used in the absence of -d.
FF_MAX_PARALLEL	When -p option is used, maximum concurrent operations are performed.

More Information

When performing ethcaptureall against hosts, SSH is used internally. The command ethcaptureall requires that password-less SSH be set up between the host running Intel® Ethernet Fabric Suite FastFabric Toolset and the hosts ethcaptureall is operating against. The ethsetupssh command can aid in setting up password-less SSH.

NOTE

The resulting host capture files can require significant amounts of space on the Intel® Ethernet Fabric Suite FastFabric Toolset host. Actual size varies, but sizes can be multiple megabytes per host. Intel recommends that you ensure adequate space is available on the Intel® Ethernet Fabric Suite FastFabric Toolset system. In many cases, it may not be necessary to run ethcaptureall against all hosts; instead, a representative subset may be sufficient. Consult with your support representative for further information.

Related Links

Selection of Hosts on page 26

5.6.5 ethsetupsnmp

Sets up SNMP on hosts to allow ethreport to query fabric data through SNMP.

Syntax

```
ethsetupsnmp [-p] [-L] [-f hostfile] [-h 'hosts'] [-a admin]
[-c community] [-m mibs]
```

Options

--help	Produces full help text.
-p	Performs operation against all hosts in parallel.
-f *hostfile*	Specifies the file with hosts in cluster. Default is /etc/eth-tools/hosts.
-h *hosts*	Specifies the list of hosts in cluster.
-L	Includes localhost (the current node) in setup.

`-a admin`	Specifies the list of admin hosts that can issue SNMP query. Default is the current host.
`-c community`	Specifies the community string used for SNMP query. Default is `public`.
`-m mibs`	Specifies the list of MIBs that are readable in SNMP queries. Default is all MIBs required by FastFabric.

Examples

```
ethsetupssh -h 'elrond arwen' -a 'elrond'
HOSTS='elrond arwen' ethsetupsnmp -a 'elrond'
ethsetupsnmp -a 'elrond' -c 'public' -m '1.3.6.1.2.1.1 1.3.6.1.2.1.2'
```

Environment Variables

The following environment variables are also used by this command:

`HOSTS`	List of hosts, used if `-h` option not supplied.
`HOSTS_FILE`	File containing list of hosts, used in absence of `-f` and `-h`.
`FF_MAX_PARALLEL`	When `-p` option is used, maximum concurrent operations.

Description

`ethreport` collects fabric data through issuing SNMP queries to hosts and switches. Intel recommends that you set up SNMP using `ethsetupsnmp` to ensure `ethreport` has proper permission to query each host; each host can provide desired data; and, all hosts have consistent SNMP configuration. `ethsetupsnmp` accepts the following parameters to configure SNMP. These parameters can be specified as command line arguments or collected during user interaction. `ethsetupsnmp` works under user-interaction mode when one or more of the required parameters are not specified in command line.

- *admin*: Space separated management node names. `ethsetupsnmp` will configure each host to allow SNMP query from these nodes.
- *community*: The community string for SNMP v2 query. The default is `public`. If set to a different string, ensure update `hpnmgmt_config.xml` with the string.
- *mibs*: The MIBs allow SNMP query. FastFabric requires the following MIBs. You can provide extra MIBs if they are required by other applications.

```
1.3.6.1.2.1.1 (SNMPv2-MIB:system)
1.3.6.1.2.1.2 (IF-MIB:interfaces)
1.3.6.1.2.1.4 (IP-MIB:ip)
1.3.6.1.2.1.10.7 (EtherLike-MIB:dot3)
1.3.6.1.2.1.31.1 (IP-MIB:ifMIBObjects)
```

When working under interactive mode, follow the prompts to complete the operation.

```
Configuring SNMP...
Enter space separated list of admin hosts (mindy1):
Enter SNMP community string (public):
Fast Fabric requires the following MIBs:
        1.3.6.1.2.1.1 (SNMPv2-MIB:system)
        1.3.6.1.2.1.2 (IF-MIB:interfaces)
        1.3.6.1.2.1.4 (IP-MIB:ip)
        1.3.6.1.2.1.10.7 (EtherLike-MIB:dot3)
        1.3.6.1.2.1.31.1 (IP-MIB:ifMIBObjects)
Do you accept these MIBs [y/n] (y):
Enter space separated list of extra MIBs to support (NONE):
Will config SNMP with the following settings:
admin hosts: mindy1
community: public
MIBs: 1.3.6.1.2.1.1 1.3.6.1.2.1.2 1.3.6.1.2.1.4 1.3.6.1.2.1.10.7
1.3.6.1.2.1.31.1
Do you accept these settings [y/n] (y):
scp -q /usr/sbin/ethsetupsnmp root@[mindy1]:/tmp/ethsetupsnmp
scp -q /usr/sbin/ethsetupsnmp root@[mindy2]:/tmp/ethsetupsnmp
[root@phwfstl005]# /tmp/ethsetupsnmp -l -a 'mindy1' -c 'public' -m
'1.3.6.1.2.1.1 1.3.6.1.2.1.2 1.3.6.1.2.1.4 1.3.6.1.2.1.10.7 1.3.6.1.2.1.31.1
';rm -f /tmp/ethsetupsnmp
Configuring SNMP...
SNMP configuration completed
[root@phwfstl006]# /tmp/ethsetupsnmp -l -a 'mindy1' -c 'public' -m
'1.3.6.1.2.1.1 1.3.6.1.2.1.2 1.3.6.1.2.1.4 1.3.6.1.2.1.10.7 1.3.6.1.2.1.31.1
';rm -f /tmp/ethsetupsnmp
Configuring SNMP...
SNMP configuration completed
SNMP configuration completed
```

Related Links

Management Configuration File on page 37

5.7 File Management Tools

The tools described in this section aid in copying files to and from large groups of nodes in the fabric. Internally, these tools make use of SCP.

The tools require that password-less SSH/SCP is set up between the host running the FastFabric Toolset and the hosts that are being transferred to and from. Use `ethsetupssh` to set up password-less SSH/SCP.

5.7.1 ethscpall

Copies files or directories from the current system to multiple hosts in the fabric. When copying large directory trees, use the `-t` option to improve performance. This option tars and compresses the tree, transfers the resulting compressed tarball to each node, and untars it on each node.

Use this tool for copying data files, operating system files, or applications to all the hosts (or a subset of hosts) within the fabric.

NOTES

- This tool can only copy from this system to a group of systems in the cluster. To copy from hosts in the cluster to this host, use `ethuploadall`.
- `user@` syntax cannot be used when specifying filenames.
- Be aware that for the `-r` option, when copying a single source directory, if the destination directory does not exist it will be created and the source files placed directly in it.
- In other situations, the `-r` option will copy the source directory as a directory under the destination directory.
- The `-R` option will always copy the source directory as a directory under the destination directory.
- The `-t` option will always place the files found in `source_dir` directly in the destination directory.

Syntax

```
ethscpall [-pq] [-r|-R] [-f hostfile] [-h 'hosts'] [-u user] [-B interface]
source_file ... dest_file
```

```
ethscpall [-t] [-pq] [-Z tarcomp] [-f hostfile] [-h 'hosts'] [-u user] [-B
interface] [source_dir [dest_dir]]
```

Options

`--help`	Produces full help text.
`-p`	Performs copy in parallel on all hosts.
`-q`	Does not list files being transferred.
`-r`	Performs recursive copy of directories using scp.
`-R`	Performs recursive copy of directories using rsync (only copy changed files).
`-t`	Performs optimized recursive copy of directories using tar. *dest_dir* is optional. If *dest_dir* is not specified, it defaults to the current directory name. If both *source_dir* and *dest_dir* are omitted, they both default to the current directory name.
`-h` *hosts*	Specifies the list of hosts to copy to.
`-f` *hostfile*	Specifies the file with hosts in cluster. Default is `/etc/eth-tools/hosts` file.
`-u` *user*	Specifies the user to perform copy to. Default is current user.

`-B interface`	Specifies local network interface to use for scp or rsync. **NOTE** The destination hosts specified must be accessible via the given *interface*'s IP subnet. This may imply the use of alternate hostnames or IP addresses for the destination hosts.
`-Z tarcomp`	Specifies a simple tar compression option to use, such as `--xz` or `--lzip`. When the host list is large, better compression may be preferred. When host list is small, faster compression may be preferred. `-Z ' '` will not use compression. Default is `-z`.
source_file	Specifies the file or list of source files to copy.
source_dir	Specifies the name of the source directory to copy. If omitted, current working directory is used.
dest_file or *dest_dir*	Specifies the name of the destination file or directory. If copying multiple files, use a directory name instead as the destination. If the file or directory name is omitted, the source file or current directory name is used, respectively.

Example

```
# efficiently copy an entire directory tree
ethscpall -t -p /usr/src/eth/mpi_apps /usr/src/eth/mpi_apps

# copy a group of files
ethscpall a b c /root/tools/

# copy to an explitly specified set of hosts
ethscpall -h 'arwen elrond' a b c /root/tools
HOSTS='arwen elrond' ethscpall a b c /root/tools

# copy to an explitly specified set of hosts over local eth2 nic
echscpall -h 'arwen elrond' -B eth2 a b c /root/tools
```

Environment Variables

The following environment variables are also used by this command:

`HOSTS`	List of hosts; used if `-h` option not supplied.
`HOSTS_FILE`	File containing list of hosts; used in absence of `-f` and `-h`.

FF_MAX_PARALLEL	When the -p option is used, maximum concurrent operations are performed.

Related Links

Selection of Hosts on page 26

5.7.2 ethuploadall

Copies one or more files from a group of hosts to this system. Since the file name is the same on each host, a separate directory on this system is created for each host and the file is copied to it. This is a convenient way to upload log files or configuration files for review. This tool can also be used in conjunction with ethdownloadall to upload a host-specific configuration file, edit it for each host, and download the new version to all the hosts.

NOTES

- To copy files from this host to hosts in the cluster, use ethscpall or ethdownloadall.
- user@ syntax cannot be used in filenames specified.
- A local directory within *upload_dir/* is created for each hostname.
- Each uploaded file is copied to upload_dir/*hostname*/*dest_file* within the local system.
- If more than one source file is specified or dest_file has a trailing /, a dest_file directory will be created.

Syntax

```
ethuploadall [-rp] [-f hostfile] [-d upload_dir] [-h 'hosts'] [-u user]
source_file ... dest_file
```

Options

--help	Produces full help text.
-p	Performs copy in parallel on all hosts.
-r	Performs recursive upload of directories.
-f *hostfile*	Specifies the file with hosts in cluster. Default is /etc/eth-tools/hosts file.
-h *hosts*	Specifies the list of hosts to upload from.
-u *user*	Specifies the user to perform copy to. Default is current user.

`-d upload_dir`	Specifies the directory to upload to. Default is `uploads`. If not specified, the environment variable `UPLOADS_DIR` is used. If that is not exported, the default, `uploads`, is used.
source_file	Specifies the name of files to copy to this system, relative to the current directory. Multiple files may be listed.
dest_file	Specifies the name of the file or directory on this system to copy to. It is relative to *upload_dir/hostname*.

Example

```
# upload two files from 2 hosts
ethuploadall -h 'arwen elrond' capture.tgz /etc/init.d/ipoib.cfg .

# upload two files from all hosts
ethuploadall -p capture.tgz /etc/init.d/ipoib.cfg .

# upload network config files from all hosts
ethuploadall capture.tgz /etc/init.d/ipoib.cfg pre-install
```

Environment Variables

The following environment variables are also used by this command:

`HOSTS`	List of hosts; used if `-h` option not supplied.
`HOSTS_FILE`	File containing list of hosts; used in absence of `-f` and `-h`.
`UPLOADS_DIR`	Directory to upload to, used in absence of `-d`.
`FF_MAX_PARALLEL`	When the `-p` option is used, maximum concurrent operations are performed.

Related Links

Selection of Hosts on page 26

5.7.3 ethdownloadall

Copies one or more files to a group of hosts from a system. Since the file contents to copy may be different for each host, a separate directory on this system is used for the source files for each host. This can also be used in conjunction with `ethuploadall` to upload a host-specific configuration file, edit it for each host, and download the new version to all the hosts.

NOTES

- The tool `ethdownloadall` can only copy from this system to a group of hosts in the cluster. To copy files from hosts in the cluster to this host, use `ethuploadall`.
- `user@` syntax cannot be used in filenames specified.

Syntax

```
ethdownloadall [-pr] [-f hostfile] [-h 'hosts'] [-u user]
 [-d download_dir] source_file ... dest_file
```

Options

`--help`	Produces full help text.
`-p`	Performs copy in parallel on all hosts.
`-r`	Performs recursive download of directories.
`-f hostfile`	Specifies the file with hosts in cluster. Default is `/etc/eth-tools/hosts` file.
`-h hosts`	Specifies the list of hosts to download files to.
`-u user`	Specifies the user to perform the copy. Default is the current user.
`-d download_dir`	Specifies the directory to download files from. Default is `downloads`. If not specified, the environment variable `DOWNLOADS_DIR` is used. If that is not exported, the default is used.
source_file	Specifies the list of source files to copy from the system.

NOTE

The option *source_file* is relative to *download_dir/hostname*. A local directory within *download_dir/* must exist for each host being downloaded to. Each downloaded file is copied from *download_dir/hostname/source_file*.

dest_file	Specifies the name of the file or directory on the destination hosts to copy to.

NOTE

If more than one source file is specified, *dest_file* is treated as a directory name. The given directory must already exist on the destination host. The copy fails for hosts where the directory does not exist.

Example

```
ethdownloadall -h 'arwen elrond' irqbalance vncservers /etc
# Copies two files to 2 hosts

ethdownloadall -p irqbalance vncservers /etc
# Copies two files to all hosts
```

Environment Variables

The following environment variables are also used by this command:

HOSTS	List of hosts; used if -h option not supplied.
HOSTS_FILE	File containing list of hosts; used in absence of -f and -h.
DOWNLOADS_DIR	Directory to download from, used in absence of -d.
FF_MAX_PARALLEL	When the -p option is used, the maximum concurrent operations are performed.

Related Links

Selection of Hosts on page 26

5.7.4 Simplified Editing of Node-Specific Files

(Linux) The combination of ethuploadall and ethdownloadall provide a simple and powerful mechanism for reviewing or editing node-specific files without the need to log in to each node.

For example, assume the file /etc/network-scripts/ifcfg-eth2 needs to be reviewed and edited for each host. This file typically contains the IP configuration information and may contain a unique IP address per host. Perform the following steps:

1. To upload the file from all the hosts, use the command:

```
uploadall /etc/network-scripts/ifcfg-eth2 ifcfg-eth2
```

2. Edit the uploaded files with an editor, such as vi with the command:

```
vi uploads/*/ifcfg-eth2
```

3. If the file was changed for some or all of the hosts, it can then be downloaded to all the hosts with the command:

```
ethdownloadall -d uploads ifcfg-eth2 /etc/network-scripts/ifcfg-eth2
```

Alternatively, you can download the file to a subset of hosts using the -h option or by creating an alternate host list file:

```
ethdownloadall -d uploads -h 'host1 host32' ifcfg-eth2 /etc/network-scripts/
ifcfg-eth2
```

NOTE

When downloading to a subset of hosts, make sure that only the hosts uploaded from are specified.

5.7.5 Simplified Setup of Node-Generic Files

(Linux) `ethscpall` can provide a simple and powerful mechanism for transferring generic files to all nodes.

For example, assume all nodes in the cluster use the same DNS server and TCP/IP name resolution. Perform the following steps:

1. Create an appropriate local file with the desired information. For example:

```
vi resolv.conf
```

2. Copy the file to all hosts with the command:

```
ethscpall resolv.conf /etc/resolv.conf
```

5.8 FastFabric Utilities

The CLIs described in this section are used for miscellaneous information about the fabric. They are also available for custom scripting.

5.8.1 dsa_setup

The Data Streaming Accelerator (DSA) is a high-performance data copy and transformation accelerator integrated into Intel® Xeon® Processors starting with the 4th Generation Intel® Xeon® Scalable Processors. PSM3 may be enabled to take advantage of DSA to optimize intra-node communications that use PSM3's `shm` device.

`/usr/share/eth-tools/samples/dsa_setup` is provided as a sample script to create DSA work queues in `/dev/dsa` for use by PSM3 jobs. This sample script should be copied to `/usr/local/bin/` and then edited as appropriate for the system. The resulting script must be run as `root` to configure DSA work queues each time the system reboots or immediately prior to and after each job which will use PSM3 with DSA enabled. To configure `dsa_setup` to be run at boot time, copy `/usr/share/eth-tools/samples/dsa.service` to `/etc/systemd/system/` and then edit `/etc/systemd/system/dsa.service` and follow the instructions in the file.

When configuring DSA work queues, `dsa_setup` will remove all existing DSA work queues, so if run per job, it should only be used when no other applications are using DSA. If the DSA configuration is to be selected per job, `dsa_setup` may be used in post job processing with the `-w none` or `-w restart` options to remove DSA resources after the job finishes. Then at the start of the next job, the appropriate `-w workload` option can be provided.

The use of `restart` is only required on some older distros, such as RHEL 8.6 and RHEL 9.0, to fully clear out DSA resources. Be aware that the use of restart may affect other applications that are using any of the CPU accelerators managed by the `idxd` kernel driver.

Syntax

```
dsa_setup [-u user] [-w workload] [-T timelimit]
```

or

```
dsa_setup --list
```

or

```
dsa_setup --help
```

Options

`--help`	Produces full help text.
`--list`	Shows DSA resources and configuration.
`-w workload`	Configures DSA work queues for specified workload. Default is `ai`. When run to configure DSA work queues, must be run as root. Workloads may be added by adding `setup_all_WORKLOAD` functions. Valid workload values are: `ai`, `hpc`, `shared`, `none`, and `restart`.
`-u user`	Specifies the owner for DSA work queue devices. Default is `root`. Specified as `[owner][:[group]]` similar to `chown` command.

NOTES

- If `:` is not specified, then only the user is granted read/write (`rw`) access.
- If `:` is specified, then the queues are granted group and user `rw` access for the specified group.
- If `:` is specified, but no group is specified, then the user's group is used.
- If `all` is specified, then everyone is granted `rw` access.

`-T timelimit`	Specifies the seconds to wait for DSA device discovery. Default is 0. Sometimes during boot, a non-zero timeout is needed to allow time for the idxd kernel driver to discover and enumerate the devices.

Examples

```
dsa_setup --help
dsa_setup --list
dsa_setup
dsa_setup -u myname -w ai
dsa_setup -u myname: -w ai
dsa_setup -u myname:mygroup -w hpc
dsa_setup -u :mygroup -w hpc
dsa_setup -w none
dsa_setup -w restart
```

NOTE

For more information on DSA and how to enable it within the CPU, BIOS and Linux kernel, see https://www.intel.com/content/www/us/en/developer/articles/technical/intel-sdm.html, https://www.intel.com/content/www/us/en/products/docs/ondemand/overview.html, and https://cdrdv2.intel.com/v1/dl/getContent/759709.

5.8.2 eth2rm

Permits the generation of configuration files for FastFabric or resource managers from a topology xml file.

When using a topology spreadsheet and `ethxlattopology` to design and prepare for deployment verification of a fabric, `eth2rm` may be used to generate resource manager configuration from the planned cluster design. Using this approach will allow the resulting configuration files to be complete, even if some nodes in the fabric have not yet been installed or made operational. Alternatively, `ethreport -o topology` can be used to generate a topology XML file for input to `eth2rm`. In this case, only the currently present nodes will be included.

When working with SLURM, the `eth2rm -o slurm` option should typically be used. This option will generate a SLURM configuration file that lists the hosts directly connected to each switch in a syntax that can be used by SLURM's topology/tree plugin. It also generates a single "fake" switch shown as connecting all the other switches together. This approach allows for SLURM job placement to be improved while avoiding undo overhead in SLURM. This option also allows for topologies that are not a pure fat-tree.

When the configuration is a pure fat tree or oversubscribed fat tree, the `eth2rm -o slurmfull` option may be used to generate the full description of the fabric, including all intermediate and core switches in the fat tree topology. This option may permit better job placement optimization than the output from the `eth2rm -o slurm` option. However for larger fabrics, it may also increase the overhead within SLURM.

Syntax

```
eth2rm [-v] [-q] -o output [-g|-u|-t] [-F point] [-p prefix] [-s suffix]
topology_input
```

Options

`--help`	Produces full help text.
`-v/--verbose`	Specifies verbose output.
`-q/--quiet`	Disables progress reports.
`-o/--output` *output*	Specifies the output type:
	`slurm` SLURM tree nodes. Supports a variety of topologies.

`slurmfull`	SLURM fat tree nodes and ISLs. Only supports pure trees.
`hosts`	FastFabric hosts file omitting this host

`-g/--guid`	Specifies to output switch GUIDs instead of names.
`-u/--underscore`	Changes spaces in switch names to underscores.
`-t/--trunc`	Truncates switch names at first space. This will treat large director switches as a single, big switch. If `-g`, `-u` or `-t` are not specified, the switch name's suffix, after the first space, will be placed at the start of the name. For example, 'core5 Leaf 101' becomes 'Leaf101_core5'.
`-p/--prefix` *`prefix`*	Specifies the prefix to prepend to all NIC hostnames.
`-s/--suffix` *`suffix`*	Specifies the suffix to append to all NIC hostnames.
`-F/--focus` *`point`*	Specifies the focus area for output. Limits the scope of output to links that match any of the given focus points. May be specified up to 10 times.
`-C/--check`	Performs more topology file validation. Requires all links to resolve against nodes and all nodes connected to same fabric. Treats any resolution errors as fatal.
`-S/--strict`	Performs strict topology file validation. Performs all checks in `-C`, and requires all nodes to list PortNum and all node list ports used.
`topology_input`	Specifies the topology_input file to use. '-' may be used to specify stdin.

Point Syntax

`node:`*`value`*	*value* is node description (node name).
`node:`*`value1`*`:port:`*`value2`*	*value1* is node description (node name); *value2* is port number.
`nodepat:`*`value`*	*value* is glob pattern for node description (node name).
`nodepat:`*`value1`*`:port:`*`value2`*	*value1* is glob pattern for node description (node name); *value2* is port number.
`nodetype:`*`value`*	*value* is node type (SW or NIC).

`nodetype:`*`value1`*`:port:`*`value2`*	*value1* is node type (SW or NIC); *value2* is port number.
`rate:`*`value`*	*value* is string for rate (25g, 50g, 75g, 100g).
`mtucap:`*`value`*	*value* is MTU size (2048, 4096, 8192, 10240); omits switch mgmt port 0.
`linkdetpat:`*`value`*	*value* is glob pattern for link details.
`portdetpat:`*`value`*	*value* is glob pattern for port details to value.

Examples

```
eth2rm -o slurm topology.xml
eth2rm -o slurm -F 'nodepat:compute*' -F 'nodepat:ethcore1 *' topology.xml
eth2rm -o nodes -F 'nodedetpat:compute*' topology.xml
eth2rm -o hosts topology.xml
```

5.8.3 ethexpandfile

Expands a Intel® Ethernet Fabric Suite FastFabric hosts or switches file. This tool expands and filter outs blank and commented lines. This can be useful when building other scripts that may use these files as input.

Syntax

```
ethexpandfile file
```

Options

`--help` Produces full help text.

`file` Specifies the FastFabric file to be processed.

Example

```
ethexpandfile allhosts
```

5.8.4 ethsorthosts

Sorts stdin in a typical host name order and outputs to stdout. Hosts are sorted alphabetically (case-insensitively) by any alpha-numeric prefix, and then sorted numerically by any numeric suffix. Host names may end in a numeric field, which may optionally have leading zeros. Unlike a pure alphabetic sort, this command results in intuitive sequencing of host names such as: host1, host2, host10.

This command does not remove duplicates; any duplicates are listed in adjacent lines.

Use this command to build `mpi_hosts` input files for applications that place hosts in order by name.

Syntax

```
ethsorthosts <hostlist> output_file
```

Options

`--help` Produces full help text.

`hostlist` Specifies the list of host names.

`output_file` Specifies the sorted list output.

```
ethsorthosts < host.xml > Sorted_host
```

Standard Input

```
ethsorthosts
osd04
osd1
compute20
compute3
mgmt1
mgmt2
login
```

Standard Output

```
compute3
compute20
login
mgmt1
mgmt2
osd1
osd04
```

5.8.5 ethxmlextract

Extracts element values from XML input and outputs the data in CSV format. `ethxmlextract` is intended to be used with `ethreport`, to parse and filter its XML output, and to allow the filtered output to be imported into other tools such as spreadsheets and customer-written scripts. `ethxmlextract` can also be used with any well-formed XML stream to extract element values into a delimited format.

Five sample scripts are available as prototypes for customized scripts. They combine various calls to `ethreport` with a call to `ethxmlextract` with commonly used parameters.

Syntax

```
ethxmlextract [-v][-H][-d delimiter][-e extract_element]
[-s suppress_element][-X input_file] [-P param_file]
```

Options

`--help`	Produces full help text.
`-v/--verbose`	Produces verbose output. Includes output progress reports during extraction and output prepended wildcard characters on element names in output header record.
`-H/--noheader`	Does not output element name header record.
`-d/--delimit` *delimiter*	Uses single character or string as the delimiter between element names and element values. Default is semicolon.
`-e/--extract` *extract_element*	Specifies the name of an XML element to extract. Elements can be nested in any order, but are output in the order specified. Elements can be specified multiple times, with a different attribute name or attribute value. An optional attribute (or attribute and value) can also be specified with elements: • `-e` *element* • `-e` *element:attrName* • `-e` *element:attrName:attrValue* **NOTES** • Elements can be compound values separated by a dot. For example, `Switches.Node` is a Node element contained within a Switches element. • To output the attribute value as opposed to the element value, a specification such as `-e NICs.Node:id` can be used. This will output the value of the id attribute of any Node elements within NICs element. • If desired, a specific element can be selected by its attribute value, such as `-e NICs.Node.PortInfo:LinkSpeedActive:100Gb`, which will output the value of the PortInfo element within Node element where the PortInfo element has an attribute of LinkSpeedActive with a value of `100Gb`. • A given element can be specified multiple times each with a different AttrName or attrValue.
`-s/--suppress` *suppress_element*	Specifies the name of an XML element to suppress extraction. Can be used multiple times (in any order). Supports the same syntax as `-e`.
`-X/--infile` *input_file*	Parses XML from *input_file*.

`-P/--pfile` *`param_file`*	Reads command parameters from `param_file`.

Example

Here is an example of `ethreport` output filtered by `ethxmlextract`:

```
# ethreport -o comps -x | ethxmlextract -d \; -e NodeDesc -e ChassisID -e
NumPorts -s Neighbor
Getting All Fabric Records...
Done Getting All Fabric Records
NodeDesc;ChassisID;NumPorts
hdarei001;0x0000444ca8e9ddd5;41
hds1fnc7041-eno1;0x0000a4bf01553c89;1
hds1fnc7041-eth1;0x0000a4bf01553c89;1
hds1fnc7061-eth2;0x0000a4bf015540f0;1
hds1fnc7061-eno1;0x0000a4bf015540f0;1
hds1fnc7061-eth1;0x0000a4bf015540f0;1
hds1fnc7081-eth2;0x0000a4bf01554175;1
hds1fnc7081-eno1;0x0000a4bf01554175;1
hds1fnc7081-eth1;0x0000a4bf01554175;1
```

Details

`ethxmlextract` is a flexible and powerful tool to process an XML stream. The tool:

- Requires no specific element names to be present in the XML.
- Assumes no hierarchical relationship between elements.
- Allows extracted element values to be output in any order.
- Allows an element's value to be extracted only in the context of another specified element.
- Allows extraction to be suppressed during the scope of specified elements.

`ethxmlextract` takes the XML input stream from either stdin or a specified input file. `ethxmlextract` does not use or require a connection to a fabric.

`ethxmlextract` works from two lists of elements supplied as command line or input parameters. The first is a list of elements whose values are to be extracted, called *extraction elements*. The second is a list of elements for which extraction is to be suppressed, called *suppression elements*. When an extraction element is encountered and extraction is not suppressed, the value of the element is extracted for later output in an extraction record. An extraction record contains a value for all extraction elements, including those that have a null value.

When a suppression element is encountered, then no extraction is performed during the extent of that element, from start through end. Suppression is maintained for elements specified inside the suppression element, including elements that may happen to match extraction elements. Suppression can be used to prevent extraction in sections of XML that are present, but not of current interest. For example, NodeDesc or IfAddr inside a Neighbor specification of `ethreport` can be suppressed.

`ethxmlextract` attempts to generate extraction records with data values that are valid at the same time. Specifying extraction elements that are valid in the same scope produces a single record for each group of extraction elements. However, mixing extraction elements from different scopes (including different XML levels) may cause `ethxmlextract` to produce multiple records.

`ethxmlextract` outputs an extraction record under the following conditions:

- One or more extraction elements containing a non-null value go out of scope (that is, the element containing the extraction elements is ended) and a record containing the element values has not already been output.
- A new and different value is specified for an extraction element and an extraction record containing the previous value has not already been output.

Element names (extraction or suppression) can be made context-sensitive with an enclosing element name using the syntax `element1.element2`. In this case, `element2` is extracted (or extraction is suppressed) only when `element2` is enclosed by `element1`.

The syntax also allows '`*`' to be specified as a wildcard. In this case, `*.element3` specifies `element3` enclosed by any element or sequence of elements (for example, `element1.element3` or `element1.element2.element3`). Similarly, `element1.*.element3` specifies `element3` enclosed by `element1` with any number of (but at least 1) intermediate elements.

`ethxmlextract` prepends any entered element name not containing a '`*`' (anywhere) with '`*.`', matching the element regardless of the enclosing elements.

NOTE

Any element names that include a wildcard should be quoted to the shell attempting to wildcard match against filenames.

At the beginning of operation, `ethxmlextract`, by default, outputs a delimited header record containing the names of the extraction elements. The order of the names is the same as specified on the command line and is the same order as that of the extraction record. Output of the header record can be disabled with the `-H` option. By default, element names are shown as they were entered on the command line. The `-v` option causes element names to be output as they are used during extraction, with any prepended wildcard characters.

Options (parameters) to `ethxmlextract` can be specified on the command line, with a parameter file, or using both methods. A parameter file is specified with `-P` *`param_file`*. When a parameter file specification is encountered on the command line, option processing on the command line is suspended, the parameter file is read and processed entirely, and then command line processing is resumed.

Option syntax within a parameter file is the same as on the command line. Multiple parameter file specifications can be made, on the command line or within other parameter files. At each point that a parameter file is specified, current option processing is suspended while the parameter file is processed, then resumed. Options are processed in the order they are encountered on the command line or in parameter files. A parameter file can be up to 8192 bytes in size and may contain up to 512 parameters.

5.8.6 ethxmlfilter

Processes an XML file and removes all specified XML tags. The remaining tags are output and indentation can also be reformatted. `ethxmlfilter` is the opposite of `ethxmlextract`.

Syntax

```
ethxmlfilter [-t|-k] [-l] [-i indent] [-s element] [input_file]
```

Options

`--help`	Produces full help text.
`-t`	Trims leading and trailing whitespace in tag contents.
`-k`	Keeps newlines as-is in tags with purely whitespace that contain newlines. Default is to format as an empty list.
`-l`	Adds comments with line numbers after each end tag. Makes comparison of resulting files easier since original line numbers are available.
`-i` *indent*	Sets indentation to use per level. Default is 4.
`-s` *element*	Specifies the name of the XML element to suppress. Can be used multiple times (maximum of 100) in any order.
input_file	Specifies the XML file to read. Default is `stdin`.

5.8.7 ethxmlindent

Takes well-formed XML as input, filters out comments, and generates a uniformly-indented equivalent XML file. Use `ethxmlindent` to reformat files for easier reading, reviewing. Can also be used for reformating a file for easy comparison with a diff tool.

Syntax

```
ethxmlindent [-t|-k] [-i indent] [input_file]
```

Options

`--help`	Produces full help text.
`-t`	Trims leading and trailing whitespace in tag contents.
`-k`	keeps newlines as-is in tags with purely whitespace that contain newlines. Default is to format as an empty list.
`-i` *indent*	Sets indentation to use per level. Default is 4.

input_file Specifies the XML file to read. Default is stdin.

5.8.8 ethxmlgenerate

Takes comma-separated-values (CSV) as input and generates sequences of XML containing user-specified element names and element values within start and end tag specifications. Use this tool to create an XML representation of fabric data from its CSV form.

Syntax

```
ethxmlgenerate [-v] [-d delimiter] [-i number] [-g element]
[-h element] [-e element] [-X input_file] [-P param_file]
```

Options

--help	Produces full help text.
-g/--generate *element*	Generates an XML element with given name, using value in next field from the input file. Can be used multiple times on the command line. Values are assigned to elements in order.
-h/--header *element*	Specifies the name of the XML element that is the enclosing header start tag.
-e/--end *element*	Specifies the name of the XML element that is the enclosing header end tag.
-d/--delimit *delimiter*	Specifies the delimiter character that separates values in the input file. Default is semicolon.
-i/--indent *number*	Specifies the number of spaces to indent each level of XML output. Default is 0.
-X/--infile *input_file*	Generates XML from CSV in input_file. One record per line with fields in each record separated by the specified delimiter.
-P/--pfile *param_file*	Uses input command line options (parameters) from param_file.
-v/--verbose	Produces verbose output. Includes output progress reports during extraction.

Details

ethxmlgenerate takes the CSV data from an input file. It generates fragments of XML, and in combination with a script, can be used to generate complete XML sequences. ethxmlgenerate does not use nor require a connection to an Intel® Ethernet Fabric.

`ethxmlgenerate` reads CSV element values and applies element (tag) names to those values. The element names are supplied as command line options to the tool and constitute a template that is applied to the input.

Element names on the command line are of three types, distinguished by their command line option: `Generate`, `Header`, and `Header_End`. The `Header` and `Header_End` types together constitute enclosing element types. Enclosing elements do not contain a value, but serve to separate and organize `Generate` elements.

`Generate` elements, along with a value from the CSV input file, cause XML in the form of `<element_name>`*`value`*`</element_name>` to be generated. `Generate` elements are normally the majority of the XML output since they specify elements containing the input values. Header elements cause an XML header start tag of the form `<element_name>` to be generated. `Header_End` elements cause an XML header end tag of the form `</element_name>` to be generated. Output of enclosing elements is controlled entirely by the placement of those element types on the command line. `ethxmlgenerate` does **not** check for matching start and end tags or proper nesting of tags.

Options (parameters) to `ethxmlgenerate` can be specified on the command line, with a parameter file, or both. A parameter file is specified with `-P` *`param_file`*. When a parameter file specification is encountered on the command line, option processing on the command line is suspended, the parameter file is read and processed entirely, and then command line processing is resumed. Option syntax within a parameter file is the same as on the command line. Multiple parameter file specifications can be made on the command line or within other parameter files. At each point that a parameter file is specified, current option processing is suspended while the parameter file is processed, then resumed. Options are processed in the order they are encountered on the command line or in parameter files. A parameter file can be up to 8192 bytes in size and may contain up to 512 parameters.

Using ethxmlgenerate to Create Topology Input Files

`ethxmlgenerate` can be used to create scripts to translate from user-specific format into the `ethreport topology_input` file format. `ethxmlgenerate` itself works against a CSV style file with one line per record. Given such a file, it can produce hierarchical XML output of arbitrary complexity and depth.

The typical flow for a script that translates from a user-specific format into `ethreport topology_input` would be:

1. As needed, reorganize the data into link and node data CSV files, in a sequencing similar to that used by `ethreport topology_input`. One link record per line in one temporary file and one node record per line in another temporary file.
2. The script must directly output the boilerplate for XML version, etc.
3. `ethxmlgenerate` can be used to output the Link section of the `topology_input`, using the link record temporary file.
4. `ethxmlgenerate` can be used to output the Node sections of the `topology_input` using the node record temporary file. If desired, there could be separate node record temporary files for NICs and Switches.
5. The script must directly output the closing XML tags to complete the `topology_input` file.

5.8.9 ethcheckload

Returns load information on hosts in the fabric.

Syntax

```
ethcheckload [-f hostfile] [-h 'hosts'] [-r] [-a|-n numprocs] [-d uploaddir] [-H]
```

Options

`--help`	Produces full help text.
`-f hostfile`	Specifies the file with hosts to check. Default is `/etc/eth-tools/hosts`.
`-h hosts`	Specifies the list of hosts to check.
`-r`	Reverses output to show the least busy hosts. Default is busiest hosts.
`-n numprocs`	Specifies the number of top hosts to show. Default is 10.
`-a`	Shows all hosts. Default is 10.
`-d upload_dir`	Specifies the target directory to upload `loadavg`. Default is `uploads`.
`-H`	Suppresses headers for script parsing.

Examples

```
ethcheckload
ethcheckload -h 'arwen elrond'
HOSTS='arwen elrond' ethcheckload
```

Environment Variables

The following environment variables are also used by this command:

`HOSTS`	List of hosts, used if `-h` option not supplied.
`HOSTS_FILE`	File containing list of hosts, used in absence of `-f` and `-h`.
`UPLOADS_DIR`	Directory to upload loadavg, used in absence of `-d`.
`FF_MAX_PARALLEL`	Maximum concurrent operations.

5.8.10 ethshmcleanup

If a PSM3 job terminates abnormally, such as with a segmentation fault, there could be POSIX shared memory files left over in the `/dev/shm` directory. This script is intended to remove unused files related to PSM3.

The unused files that are removed include:

- `/dev/shm/psm3_shm*`
- `/dev/shm/sem.psm3_nic_affinity*`
- `/dev/shm/psm3_nic_affinity*`

Syntax

```
ethshmcleanup
```

Options

`--help` Produces full help text.

Examples

```
ethshmcleanup
```

6.0 FastFabric Diagnostics Capabilities

6.1 Overview

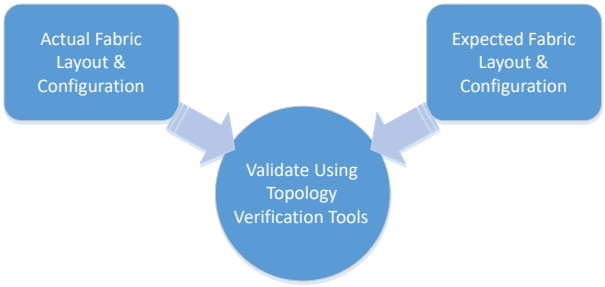
Many features are built into the Intel® Ethernet Fabric Suite Software to help diagnose fabric issues. For example, many tools found in the FastFabric focus on topology verification, which gives you a way to compare the current fabric configuration against the expected fabric configuration. Fabric anomalies can be analyzed to resolve issues. The tools can help you debug cabling, NICs, switches, or configuration issues.

6.2 Topology Verification

FastFabric provides several ways to assist you in validating a running fabric's configuration and layout against a predefined/expected topology. This verification process can help you to identify issues including detecting missing cables, hosts, or switches; or verifying that cables are in the correct places. You can run topology verification tools during the initial fabric startup or at any other time after a fabric is configured.

6.2.1 Creating the Expected Fabric Layout File

The expected fabric layout is defined by an input file in XML format. Topology verification tools use this file along with the actual fabric configuration to analyze a fabric.



You use the `ethxlattopology` tool to translate a user-friendly, CSV-formatted file into the required XML format. To create your custom `<topologyfile>.xml` file, perform the following:

1. Edit one of the sample files (`detailed_topology.xlsx` or `minimal_topology.xlsx`) found in the `/usr/share/eth-tools/samples` folder to depict the expected layout of the fabric.

   Refer to Sample Topology Spreadsheet Overview for more details.

2. Save your custom `<topologyfile>.xlsx` as CSV format.

3. Run the ethxlattopology tool against the CSV file to get the expected fabric layout in XML format.

   Refer to ethxlattopology for more details.

   You can now use the <topologyfile>.xml file to verify or validate against the actual fabric layout.

6.2.2 Validating a Topology Against an Actual Fabric Layout

ethreport is one of the main tools provided by FastFabric used in the topology verification process. It provides various options to verify specific parts of the fabric. All these options require a customized <topologyfile>.xml file as an input.

The following table summarizes the ethreport options as well as the specific area of the fabric that they are used to verify.

Command	Verifies
ethreport -o verifyfis -T <topologyfile>.xml	NIC
ethreport -o verifysws -T <topologyfile>.xml	Switches
ethreport -o verifynodes -T <topologyfile>.xml	Nodes
ethreport -o verifylinks -T <topologyfile>.xml	Links
ethreport -o verifyextlinks -T <topologyfile>.xml	Links External to a System
ethreport -o verifyfilinks -T <topologyfile>.xml	Links to NIC
ethreport -o verifyislink -T <topologyfile>.xml	Inter Switch Links
ethreport -o verifyextislink -T <topologyfile>.xml	Inter Switch Links, External to System
ethreport -o verifyall -T <topologyfile>.xml	All the Parameters

Advanced Topology Verification on page 117 provides a detailed explanation to help interpret the ethreport output for some of the output types above.

Refer to the following for additional verification details on:

- Topology verification using ethreport
- Other topology verification tools: ethextractbadlinks, ethextractlink, ethextractmissinglinks, ethextractsellinks, ethextractstat2, and ethlinkanalysis
- Fabric verification using ethfabricanalysis

6.2.3 Interpreting Output of Topology Verification Tools

You can use the command `ethreport -o verifyall -T <topologyfile>.xml` to verify NIC, switches, and links.

No Errors Detected

If no errors are detected in the topology, an output is shown similar to the example below:

```
[root@node057 topology]$ ethreport -o verifyall -T topology.xml
Getting All Fabric Records...
Done Getting All Fabric Records
Parsing topology.xml...
NICs Topology Verification

NICs Found with incorrect configuration:
4 of 4 Fabric NICs Checked

NICs Expected but Missing or Duplicate in input:
4 of 4 Input NICs Checked

Total of 0 Incorrect NICs found
0 Missing, 0 Unexpected, 0 Duplicate, 0 Different
--------------------------------------------------------------------------------
SWs Topology Verification

SWs Found with incorrect configuration:
1 of 1 Fabric SWs Checked

SWs Expected but Missing or Duplicate in input:
1 of 1 Input SWs Checked

Total of 0 Incorrect SWs found
0 Missing, 0 Unexpected, 0 Duplicate, 0 Different
--------------------------------------------------------------------------------
Links Topology Verification

Links Found with incorrect configuration:
4 of 4 Fabric Links Checked

Links Expected but Missing, Duplicate in input or Incorrect:
4 of 4 Input Links Checked

Total of 0 Incorrect Links found
0 Missing, 0 Unexpected, 0 Misconnected, 0 Duplicate, 0 Different
--------------------------------------------------------------------------------
```

For each verification type run by the tool, the following attributes are reported:

- Missing
- Unexpected
- Misconnected
- Duplicate
- Different

A description for each of these attributes can be found in Table 9 on page 94.

Possible Errors Detected

The following examples show possible error conditions, as well as an interpretation of the output that the `ethreport` tool will report.

NOTE

The `ethreport` verification tool reports all issues from multiple perspectives. Therefore, it may output incorrect links more than once. The tool does not try to match up issues or differentiate the source of duplicate errors. For example, the same error could be reported twice, once appearing as a mismatched port on an expected link and again as an error looking like a cabling mistake.

Example 1

The following example shows an output of a verification where only one side of the expected links matches a port in fabric. No link exists in the topology file for `node059-eth2`. For this error condition, the tool can confidently match the other side of the link and can indicate that one side of the link is incorrect. Two issues are reported: one unexpected and one misconnected link.

```
[root@node057 topology]$ ethreport -o verifyall -T topology.xml
Getting All Fabric Records...
Done Getting All Fabric Records
Parsing topology.xml...
NICs Topology Verification

NICs Found with incorrect configuration:
4 of 4 Fabric NICs Checked

NICs Expected but Missing or Duplicate in input:
4 of 4 Input NICs Checked

Total of 0 Incorrect NICs found
0 Missing, 0 Unexpected, 0 Duplicate, 0 Different
--------------------------------------------------------------------------------
SWs Topology Verification

SWs Found with incorrect configuration:
1 of 1 Fabric SWs Checked

SWs Expected but Missing or Duplicate in input:
1 of 1 Input SWs Checked

Total of 0 Incorrect SWs found
0 Missing, 0 Unexpected, 0 Duplicate, 0 Different
--------------------------------------------------------------------------------
Links Topology Verification

Links Found with incorrect configuration:
Rate IfAddr            Port PortId          Type Name
100g 0x001175010265baf7   3 Eth3            SW   Switchbaf7
<->  0x0011758101660683   1 758101660683    NIC  node059-eth2
Unexpected Link

4 of 4 Fabric Links Checked

Links Expected but Missing, Duplicate in input or Incorrect:
Rate MTU   IfAddr             Port/MgmtIfAddr PortId           Type Name
100g  8192                       1             758101660683    NIC  login1-eth1
<->                              3             Eth3            SW   Switchbaf7
Incorrect Link, 1st port found to be:
     0x0011758101660683   1 NIC   node059-eth2

4 of 4 Input Links Checked
```

```
Total of 2 Incorrect Links found
0 Missing, 1 Unexpected, 1 Misconnected, 0 Duplicate, 0 Different
--------------------------------------------------------------------------------
```

Example 2

The following example shows a situation where both sides of an expected link match the same port in the fabric. In this output, the tool indicates that either the input topology file has an issue (duplicate occurrence of the port) or the link is incorrectly cabled.

```
[root@node057 topology]$ ethreport -o verifyall -T topology.xml
Getting All Fabric Records...
Done Getting All Fabric Records
Parsing topology.xml...
NICs Topology Verification

NICs Found with incorrect configuration:
4 of 4 Fabric NICs Checked

NICs Expected but Missing or Duplicate in input:
4 of 4 Input NICs Checked

Total of 0 Incorrect NICs found
0 Missing, 0 Unexpected, 0 Duplicate, 0 Different
--------------------------------------------------------------------------------
SWs Topology Verification

SWs Found with incorrect configuration:
1 of 1 Fabric SWs Checked

SWs Expected but Missing or Duplicate in input:
1 of 1 Input SWs Checked

Total of 0 Incorrect SWs found
0 Missing, 0 Unexpected, 0 Duplicate, 0 Different
--------------------------------------------------------------------------------
Links Topology Verification

Links Found with incorrect configuration:
Rate IfAddr             Port PortId         Type Name
100g 0x001175010160357f   1 75010160357f   NIC  node060-eth2
<->  0x001175010265baf7   4 Eth4           SW   Switchbaf7
Unexpected Link

4 of 4 Fabric Links Checked

Links Expected but Missing, Duplicate in input or Incorrect:
Rate MTU   IfAddr             Port/MgmtIfAddr PortId         Type Name
100g  8192                       1            75010160357f   NIC  node059-eth2
    Duplicate Port in input or incorrectly cabled
<->                              5            Eth5           SW  Switchbaf7
    Duplicate Port in input or incorrectly cabled
Duplicate Port in input or incorrectly cabled

4 of 4 Input Links Checked

Total of 2 Incorrect Links found
0 Missing, 1 Unexpected, 0 Misconnected, 1 Duplicate, 0 Different
--------------------------------------------------------------------------------
```

7.0 MPI Sample Applications

As part of a Intel® Ethernet Fabric Suite FastFabric Toolset installation, sample MPI applications and benchmarks are installed in `/usr/src/eth/mpi_apps`. The sample applications can be used to perform basic tests and performance analysis of MPI, the servers, and the fabric.

The sample applications provided in the package include:

- OSU Micro-Benchmarks
- Intel® MPI Benchmarks (IMB)
- High Performance Linpack (HPL2)
- Latency/bandwidth deviation test
- Latency test
- Bandwidth test
- MPI stress test
- MPI fabric stress tests
- MPI batch run_* scripts

In addition a set of run_* scripts and supporting infrastructure are provided to make it easier to run the applications and keep track of the results from the runs

To tune the fabric for optimal performance, refer to the *Intel® Ethernet Fabric Performance Tuning Guide*.

7.1 Building and Running Sample Applications

7.1.1 Building MPI Sample Applications

The Intel® Ethernet Fabric Suite FastFabric TUI can assist with building the MPI sample applications. Follow Building Test Applications and Copying to Hosts to complete the operation.

Alternatively, you can perform the following procedure to build the applications:

1. If a GPU enabled build of the sample applications is desired, do one of the following:
   - `export MPI_APPS_ONEAPI=y` This will build the sample applications for Intel GPUs. The Intel® MPI Benchmarks (IMB) `MPI1-GPU` suite will also be built and mpi_stress Test will be enabled to use GPU buffers.
   - `export MPI_APPS_CUDA=y` This will build the sample applications for NVIDIA GPUs. Optionally `CUDA_DIR` may also be specified with the directory where CUDA is installed, the default is `/usr/local/cuda`. Various OSU Micro-Benchmarks and mpi_stress Test will be enabled to use GPU buffers.

2. Do one of the following:
   - If using Intel MPI as included in the oneAPI packaging: type `source /opt/intel/oneapi/setvars.sh`
   - If using Intel MPI outside the oneAPI packaging or a specific version of Intel MPI: type `source PREFIX/env/vars.sh`

     where:
     - `PREFIX` is the path to the desired version of Intel MPI.
   - If using another MPI: type `export MPICH_PREFIX=/usr/mpi/COMPILER/Y`

     where:
     - `COMPILER` is the compiler that MPI uses. It can be `gcc`, `icc`, etc.
     - `Y` is an MPI variation such as `openmpi-4.1.4-ofi`.
3. Type `cp -r /usr/src/eth/mpi_apps/ WORKSPACE` to copy `mpi_apps` to your workspace

   where: `WORKSPACE` is the destination location such as `/root` or your home directory.

   Alternately, if you use the `mpi-selector` package to define which MPI you use, you can use the `get_selected_mpi.sh` script to do this for you by typing: `. /usr/src/eth/mpi_apps/get_selected_mpi.sh`.

   This will show you the currently-selected MPI and set the `MPICH_PREFIX` variable to match.
4. Type `cd WORKSPACE/mpi_apps`.
5. Type `make clean`.
6. Type `make all` to build all of the sample applications.

   Build target alternatives include:
   - `eth-base` - Builds applications in core RPM: Deviation, group stress, and `mpi_check`.
   - `all` – Builds everything in `eth-base`, plus OSU Benchmarks, Intel® MPI Benchmarks (IMB), Deviation, HPL2, and Group Stress.
   - `full` and `quick` are currently aliases for `all`.

NOTE

The MPI used does not have to be in the `/usr/mpi/COMPILER` directory. The default MPIs installed with the Intel® EFS Software are located here, however, you can also export `MPICH_PREFIX` to point to any location where you have another third-party MPI installed (e.g., such as alternate versions of Intel MPI).

7.1.2 Running MPI Sample Applications

Intel provides `run_*` scripts to facilitate running the applications. These run scripts allow the `mpi_hosts` filename to be specified through the environment variable `MPI_HOSTS`. If this variable is not defined, the default `$PWD/mpi_hosts` is used.

To run the applications, an mpi_hosts file must be created in WORKSPACE/mpi_apps that provides the names of the hosts on which processes should be run. This file must list one host per line, but may contain comments with a # in the fist column.

NOTE

The run scripts select the number of processes for the job, typically via their first argument or for some scripts an implicit use of two processes. If the mpi_hosts file contains more entries than needed, the extra entries at the end of the file will be silently ignored.

NOTE

Some run scripts change directory before launching the job, so it is recommended to always specify a full pathname for MPI_HOSTS.

Since modern servers have multi-core CPUs, it's often desirable to run more than 1 process per host. This can be accomplished in a few ways:

- Simply list the given host multiple times in the MPI_HOSTS file, in the order you would like ranks assigned to the hosts.
- Specify PROCS_PER_NODE in the environment (default is 1). In which case PROCS_PER_NODE processes will be started as sequential ranks in the job for each host found in the MPI_HOSTS file.
- Specify more processes for the job via the run scripts arguments, than PROCS_PER_NODE times the number of entries in the MPI_HOSTS file. In which case, once the MPI_HOSTS is exhausted, subsequent processes will be assigned starting back at the beginning of the MPI_HOSTS file.
- Specify all to the run script as the number of processes, in which case PROCS_PER_NODE processes will be started for every entry in the MPI_HOSTS file.

NOTE

A best practice can be the creation of separate mpi_hosts files for frequently used lists of hosts and use of the MPI_HOSTS environment variable to select the proper file for a given run.

NOTE

When running the applications, all hosts listed in MPI_HOSTS must have a copy of the applications (in the same WORKSPACE directory name) and compiled for the same value of MPICH_PREFIX, for example, the same variation and version of MPI.

NOTE

Some implementations of MPI offer binary compatibility for applications across multiple versions of the MPI library. For example, Intel MPI has this capability. When using such an MPI, you may set MPICH_PREFIX at runtime to point to a different version of MPI than the one the sample applications were built against. However, such a MPI must be binary compatible with the MPI the applications were built against.

When the `run_*` scripts are used to execute the applications, the variation of MPI used to build the applications (or selected via `MPICH_PREFIX`) is detected and the proper mpirun is used to start the application.

To determine which variation of MPI the applications have been built with, use the command:

```
cat WORKSPACE/mpi_apps/.prefix
```

NOTE

Some variations of MPI may require that the MPD daemon be started prior to running applications. Consult the documentation on the specific variation of MPI for more information on how to start the MPD daemon.

When MPI applications are run with the `run_*` scripts provided, the results of the run are logged to `WORKSPACE/mpi_apps/logs`. The filename for a given run includes the date, application name, time of the run, and `LOGSUFFIX`.

7.1.2.1 MPI Sample Applications Parameter Files

To allow for easy tuning of job parameters, the run scripts include a number of parameter files. The parameter files are simply bash scripts which can set environment variables directly interpreted by the job, such as `PSM3_IDENTIFY` or add settings and variables directly to the mpirun command line by appending to `MPI_CMD_ARGS`.

The `run_*` scripts automatically use the `intelmpi.params` or `openmpi.params` files to set up parameters for mpirun. The file used is selected based on the MPI variation being used. In addition the `psm3.params` file is always used. Finally, for jobs using oneCCL, such as via the `run_oneccl` script, the `oneccl.params` file will also be used.

These files have various samples of setting commonly used parameters which can be easily un-commented and/or edited. It is recommended to use the bash `export variable=${variable:-value}` syntax to set variables in these files such that the user can override settings via the environment or so files like `intelmpi.params` can override settings in the `psm3.params` file when necessary. The parameter files can also set the `MPI_CMD_ARGS` variable to provide additional arguments directly to mpirun. Any such settings in `MPI_CMD_ARGS` must be done using the proper syntax for the selected MPI, so Intel recommends avoiding setting `MPI_CMD_ARGS`, especially in the `psm3.params` file.

The files are included in the following order:

1. `psm3.params`
2. `intelmpi.params` or `openmpi.params` as appropriate
3. When using oneCCL, such as `run_oneccl`, the `oneccl.params` file.

When the job is run, settings and variables specified via MPI_CMD_ARGS will take precedence over variables in the environment. If a setting or variable appears in `MPI_CMD_ARGS` more than once, typically the last specification of the given value will be used.

In most cases, a given setting will be specified in only one file or the environment. When using the Intel recommended `export variable=${variable:-value}` syntax, variables in the files may also be potentially overridden by the environment.

The following list reflects the detailed order of precedence if a given variable is specified in multiple places, in which case the entry in this list with the lower number will "win".

1. When using oneCCL, such as `run_oneccl`, settings added to `MPI_CMD_ARGS` in the `oneccl.params` file. These will be placed at the end of the `mpirun` command line.
2. Settings added to `MPI_CMD_ARGS` in the relevant `intelmpi.params` or `openmpi.params` file. These will be placed at the end of the `mpirun` command line before any from the `oneccl.params` file.
3. When using oneCCL, such as `run_oneccl`, variables added to `MPI_CMD_ARGS` in the `oneccl.params` file with an explicit value such as `-genv variable=value` without using the `${variable:-value}` syntax .
4. Variables added to `MPI_CMD_ARGS` in the relevant `intelmpi.params` or `openmpi.params` file with an explicit value such as `-genv variable=value` for Intel MPI or `-x variable=value` for openmpi without using the `${variable:-value}` syntax.
5. When using oneCCL, such as `run_oneccl`, explicit settings of variables in the `oneccl.params` file such as `variable=value` without using the `${variable:-value}` syntax.
6. Variables set in the relevant `intelmpi.params` or `openmpi.params` file with an explicit value such as `variable=value` without using the `${variable:-value}` syntax.
7. Variables set in the `psm3.params` file with an explicit value such as `variable=value` without using the `${variable:-value}` syntax.
8. Environment variables passed into the job such as an export immediately before starting the job or on the job command line. Such as: `variable=value ./run_x`.
9. Environment variables set by the system or user. Such as via `.bashrc`, or explicitly loaded modules files.
10. Variables set in the `psm3.params` file with the `${variable:-value}` syntax.
11. Variables set in the relevant `intelmpi.params` or `openmpi.params` file with the `${variable:-value}` syntax.
12. When using oneCCL, such as `run_oneccl`, variables set in the `oneccl.params` file with the `${variable:-value}` syntax.
13. Variables set in `/etc/psm3.conf`.

7.1.2.2 Available Sample Application run Scripts

The following tables list the run scripts.

Table 10. Benchmark Run Scripts

mpi_apps run Script	Application	Application Directory Under mpi_apps	Description
run_alltoall5	osu_alltoall	osu-micro-benchmarks-5.9/mpi/collective/	Benchmark MPI AlltoAll collective
run_bcast5	osu_bcast	osu-micro-benchmarks-5.9/mpi/collective/	Benchmark MPI Broadcast collective
run_bibw5	osu_bibw	osu-micro-benchmarks-5.9/mpi/pt2pt/	Benchmark bi-directional MPI point to point (send-recv) bandwidth
run_bw5	osu_bw	osu-micro-benchmarks-5.9/mpi/pt2pt/	Benchmark uni-directional MPI point to point (send-recv) bandwidth
run_imb	IMB-MPI1 or other IMB test suites	imb/	The Intel MPI Benchmark. This test performs a variety of point to point and collective benchmarks across a group of hosts. For more information see https://software.intel.com/content/www/us/en/develop/documentation/imb-user-guide/
run_oneccl	benchmark	not included in mpi_apps, directory specified via `ONECCL_EXAMPLES_DIR`	The Intel oneCCL example Benchmark. This test performs a variety of oneCCL collectives across a group of hosts. For more information see https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/.
run_lat5	osu_latency	osu-micro-benchmarks-5.9/mpi/pt2pt/	Benchmark one-way MPI point to point (send-recv) latency
run_mbw_mr5	osu_mbr_mr	osu-micro-benchmarks-5.9/mpi/pt2pt/	Benchmark uni-directional MPI point to point (send-recv) multi-process message rate.
run_multi_lat5	osu_multi_lat	osu-micro-benchmarks-5.9/mpi/pt2pt/	Benchmark latency with multiple pairs of processes concurrently communicating
run_osu5	specified on command line	osu-micro-benchmarks-5.9/mpi/	Run an OSU pt2pt or collective MPI benchmark. For more information on OSU benchmarks see mpi_apps/osu-micro-benchmarks-5.6.3/README

Table 11. Tests Run Scripts

mpi_apps run Script	Application	Application Directory Under mpi_apps	Description
run_allniclatency	mpi_latencystress	groupstress/	Measure latency for all pairings of NICs and identify outliers. Useful as a fabric routing and cabling verification test. Some overlap with deviation test.
run_app	runmyapp	./	Wrapper to aid debug of an application. Edit runmyapp to invoke the proper program (xhpl shown as an example In runmyapp)
run_batch_cabletest	run_cabletest	./	Launch multiple cable stress tests, each with a batch of hosts. Useful to stress host to switch cables as a fabric verificaition test. Used by ethcabletest
run_batch_script	specified on command line	N/A	Build a set of mpi_hosts files each with up to BATCH_SIZE hosts and then launch the given MPI application for each set of hosts in parallel. Useful to construct various fabric stress tests and benchmarks, such as running concurrent pairwise latency tests or running multiple independent collective tests.
run_bw	bw	bandwidth/	Test uni-directional MPI point to point (send-recv) bandwidth. This is based on an earlier version of osu_bw. For benchmarking run_bw5 (osu_bw), run_imb (IMB-MPI1) or other tests should be used. This test is useful to aid debug of fabric or software issues as the specific sizes to test can be controlled and the script can be easily modified to run a single message size.
run_cabletest	mpi_groupstress	groupstress/	Run a bi-directional high bandwidth stress test among a group of hosts. Useful to stress host to switch cables as a fabric verificaition test. Used by ethcabletest via run_batch_cabletest.
run_deviation	deviation	deviation/	Execute pairwise latency and bandwidth tests across a group of hosts to identify hosts which are outliers. Arguments control tests and thresholds. Only a subset of pairings are run with a focus on identiying outlier hosts which may have configuration, HW or cable issues.

continued...

mpi_apps run Script	Application	Application Directory Under mpi_apps	Description
run_hpl2	xhpl	hpl-2.3/bin/*/	Run an HPL2 test. This is used by hostverify.sh as part of ethverifyhosts. The focus is on use of single node HPL as a HW and software stress test and to compare hosts within a cluster to identify outliers. This should not be used for top500 submissions or comparisons as there are other more optimized versions of this benchmark built with optimized linear algebra (BLAS) libraries such as Intel's Math Kernel Library (MKL) library.
run_lat	latency	latency/	Test MPI point to point (send-recv) latency. This is based on an earlier version of osu_lat. For benchmarking run_lat5 (osu_latency), run_imb (IMB-MPI1) or other tests should be used. This test is useful to aid debug of fabric or software issues as the specific sizes and loop counts to test can be controlled and the script can be easily modified to run a single message size.
run_mpicheck	mpicheck	mpicheck/	Verify MPI communications stack. This is not a benchmark.
run_mpi_stress	mpi_stress	mpi_stress/	Stress and validate the MPI communications stack
run_multibw	mpi_multibw	mpi_multibw/	Benchmark uni-directional and bi-direction MPI point to point (send-recv) multi-process message rate. This is a variation of a previous OSU message rate benchmark which includes a fix for a bug which can cause inaccurately high reporting of message rate.

NOTE

Each of the run scripts is a relatively brief and simple bash script. If desired, these may be copied and edited to create new variations or to hardcoded additional options for the underlying test program they run. The heart of the run script infrastructure is in the prepare_run script. Due to the potential to impacting all run scripts, Intel recommends you not modify the prepare_run script.

7.1.2.3 Examples of Running MPI Sample Applications

The typical procedure for running a given sample application is:

1. Edit the relevant set of parameter files (`WORKSPACE/mpi_apps/*.params`) and review or alter settings and variables as desired.

2. Execute the desired run script(s) with their required and/or optional arguments as appropriate.

Typically a user will establish a common set of parameters planned for all runs and will set the parameters files only once.

When experimenting or tuning, it can be desirable to try multiple values for a given parameter to see what performs best. In which case the need to edit the parameter file(s) between runs can become cumbersome and inefficient. To address this issue, Intel recommends the use of the `${variable:-value}` syntax in the parameter files anytime a variable is being set or specified. This approach allows the environment to override the given parameter without needing to edit the parameter file.

For example:

```
export PSM3_RDMA=1
./run_lat5
./run_bw5
PSM3_MULTIRAIL=2 ./run_bw5
```

Will use the value `PSM3_RDMA=1` for all three jobs. This will override any setting of `PSM3_RDMA` found in the parameters file with the `${variable:-value}` syntax as well as overriding any value specified in `/etc/psm3.conf`. In this example, the last job will also use `PSM3_MULTIRAIL=2` as this is specified using the bash syntax for exporting a variable just for one command.

The run script infrastructure also allows for a number of other environment variables to be used to control how the run scripts themselves behave, this includes:

- `MPI_HOSTS` - The MPI hosts file to use. The default is `$PWD/mpi_hosts`.
- `PROCS_PER_NODE` - How many processes should be launched on a host before progressing to the next host in the `MPI_HOSTS` file. Default of `1`.
- `LOGSUFFIX` - Suffix to append to generated job specific log filename.
- `SHOW_MPI_HOSTS` - If set to `y`, the hosts being used, along with any comment lines in the `MPI_HOSTS` file prior to the last host, will be output prior to starting the job. Set to `n` to disable. Defaults to `y`.
- `SHOW_MPI_HOSTS_LINES`- Specifies the maximum lines in `MPI_HOSTS` to show. Default is `128`. Only lines applicable to the job will be shown. Comment lines prior to the last host shown are also output.
- `SHOW_ENV` - If set to `y` middleware and provider environment variables will be output prior to starting the job. Set to `n` to disable. Defaults to `y`.
- `SHOW_SLURM_ENV` - If set to `y` SLURM environment variables will be output prior to starting the job. Set to `n` to disable. Defaults to `y`.
- `MPI_TASKSET` - When specified, `/bin/taskset MPI_TASKSET` or `/usr/bin/taskset MPI_TASKSET` willl be called as part of each process in the job to control CPU selection for the process. Default is to let the middleware or application make it's own choice.

For each job run, a logfile is produced in `WORKSPACE/mpi_apps/logs/` with an generated name based on the date, application name, time of the run, and `LOGSUFFIX`. The variable `LOGFILE` is exported with the full pathname of this file. `LOGFILE` may be used as input in the various parameters files. For example, the

sample `psm3.params` file uses `LOGFILE` to set `PSM3_PRINT_STATS_PREFIX` so the statistics files will also be written to `WORKSPACE/mpi_apps/logs/` and named similarly to the job output log.

The variables can allow significant flexibility when running jobs, for example:

```
export LOGSUFFIX='-myrdma1'
export PSM3_RDMA=1
./run_lat5
./run_bw5
```

will include the string `-myrdma1` as part of the log filename of logs for each of the resulting runs. Making it easier to locate these files for future review.

```
MPI_HOSTS=spr_hosts ./run_imb all
MPI_HOSTS=spr_hosts PROCS_PER_NODE=80 ./run_imb all
MPI_HOSTS=icx_hosts ./run_imb all
MPI_HOSTS=icx_hosts PROCS_PER_NODE=30 ./run_imb all
```

will run the Intel MPI Benchmark across all the hosts listed in the `spr_hosts` file, first with 1 process per node, then with 80 processes per node. Next it will run the Intel MPI Benchmark across all the hosts listed in the `icx_hosts`file, first with 1 process per node, then with 30 processes per node.

7.1.2.4 Running Sample Applications with SLURM

SLURM is a popular job scheduler which can queue jobs, allocate nodes, and then lanch the job on the set of allocated nodes. The run scripts have been designed to make it easy to take advantage of the run scripts in a cluster using SLURM. In such a cluster the exact hosts being used for a given job may vary and be selected by SLURM when the job is launched. As such, attempting to pre-create an `mpi_hosts` file is problematic.

In support of SLURM, you can simply specify `MPI_HOSTS=slurm` in the environment. In which case the run script infrastructure will query SLURM for the list of hosts. The resulting list of hosts is placed in `$PWD/mpi_hosts.slurm` and then used to run the job just as any `mpi_hosts` file would be used. SLURM will typically list each host allocated to the job exactly once.

Alternately, if you plan to use SLURM for most of your jobs, you can simply create an `MPI_HOSTS` file whose first non-comment line says `slurm` instead of specifying a host name. This will result in the exact same behavior as if `MPI_HOSTS=slurm` was specified. If you place this value in the default `$PWD/mpi_hosts` file, you can avoid needing to specify `MPI_HOSTS` in the environment.

These capabilities coupled with the ability to use environment variables provides a powerful capability for launching multiple jobs within the SLURM job queue. For example:

```
echo "slurm" > mpi_hosts
sbatch -N 2 sbatch_args ./run_lat5
sbatch -N 2 sbatch_args ./run_bw5
sbatch -N 10 sbatch_args /run_imb all
PROCS_PER_NODE=80 sbatch -N 10 sbatch_args ./run_imb all
LOGSUFFIX='-myrdma1' PSM3_RDMA=1 sbatch -N 2 sbatch_args ./run_lat5
LOGSUFFIX='-myrdma1' PSM3_RDMA=1 sbatch -N 2 sbatch_args ./run_bw5
```

```
LOGSUFFIX='-myrdma1' PSM3_RDMA=1 sbatch -N 10 sbatch_args /run_imb all
LOGSUFFIX='-myrdma1' PSM3_RDMA=1 PROCS_PER_NODE=80 sbatch -N 10 sbatch_args ./
run_imb all
```

Submits a whole sequence of jobs, first with the parameters as specified in the various mpi_apps parameter files, and then with `PSM3_RDMA=1` and a unique `LOGSUFFIX` to help distinguish the log files later. In these samples, replace `sbatch_args` with any additional arguments which may be required by the SLURM configuration, such as selecting a SLURM partition name. Consult with your SLURM administrator for further details.

NOTE

Each of the run scripts has some SLURM comments at the start. Most of these are disabled, however the comment which sets the output file location for the job is enabled. The samples provided use the `logs` directory with a name similar to the default `LOGFILE` to make associating SLURM output files with run script logs easier.

NOTE

Since the `WORKSPACE/mpi_apps/logs` directory may have many files from recent and historic runs, one technique to view the most recent files first is a command such as `vi `ls -t logs/*``.

7.2 Sample Benchmark Applications

7.2.1 OSU Micro-Benchmarks

For more information on OSU benchmarks, see `mpi_apps/osu-micro-benchmarks-5.9/README`.

Use `run_*5` sample scripts in `WORKSPACE/mpi_apps` to run the OSU benchmarks. For example:

1. Type `cd WORKSPACE/mpi_apps`
2. Type `./run_multi_lat5` *`NP options`*

   where:

   `NP` is the number of processes to run. Specifying `all` indicates the test should be run with `PROCS_PER_NODE` processes for every entry in the `MPI_HOSTS` file. A minimum of two processes is required, except when requesting help. Specifying `--help` will provide help about the given `run_*` script itself.

   `options` is any additional options to be passed to the specific OSU benchmark. When --help is specified, the underlying OSU benchmark will provide additional help, such as `./run_multi_lat5 1 --help`.

Available Sample Application run Scripts lists the available scripts.

NOTE

`run_lat5`, `run_bw5`, and `run_bibw5` do not support *NP* the argument as they only use two processes. To obtain additional help about the script itself, specify `--help`, such as `./run_lat5 --help`. To obtain additional help about the underlying OSU benchmark, specify `-hh`, such as `./run_lat5 -hh`.

The `run_osu5` script is a generic script used to run any OSU benchmark. Its usage follows:

```
Usage: ./run_osu5 <number_of_processes> <command> [options]
           or
       ./run_osu5 --help
    number_of_processes may be 'all' in which case PROCS_PER_NODE ranks will
                      be started for every entry in the MPI_HOSTS file.
    <command> osu benchmark to run
    [options] are passed to <command>

For example: ./run_osu5 2 osu_allgatherv -f

Possible commands are:
osu_init, osu_hello, osu_bcast, osu_gather,
osu_allgather, osu_scatter, osu_iallgather, osu_ibcast,
osu_ialltoall, osu_ibarrier, osu_igather, osu_iscatter,
osu_iscatterv, osu_igatherv, osu_iallgatherv, osu_ialltoallv,
osu_ialltoallw, osu_ireduce, osu_iallreduce, osu_reduce,
osu_allreduce, osu_alltoall, osu_alltoallv, osu_allgatherv,
osu_scatterv, osu_gatherv, osu_reduce_scatter, osu_barrier,
osu_acc_latency, osu_get_bw, osu_get_latency, osu_put_bibw,
osu_put_bw, osu_put_latency, osu_get_acc_latency, osu_fop_latency,
osu_cas_latency, osu_bibw, osu_bw, osu_latency,
osu_mbw_mr, osu_multi_lat, osu_latency_mt, osu_latency_mp,


To get more details about [options] available: ./run_osu5 1 <command> --help
For example: ./run_osu5 1 osu_fop_latency --help
```

NOTE

run_multibw is a version of `run_mbw_mr5` which includes a bug fix to avoid inaccurately reporting an overly high message rate.

Related Links

run_multibw on page 209

7.2.2 Intel® MPI Benchmarks (IMB)

Use the `run_imb` sample script in `WORKSPACE/mpi_apps` to run the IMB-MPI1 benchmarks.

1. Type `cd WORKSPACE/mpi_apps`
2. Type `./run_imb NP imb_suite options`

   where:

NP is the number of processes to run. Specifying `all` indicates the test should be run with `PROCS_PER_NODE` processes for every entry in the `MPI_HOSTS` file. Specifying `--help` will provide help about the `run_imb` script itself. A minimum of two processes is required, except when requesting help.

imb_suite is an optional parameter. It may specify the suite of IMB tests. For example specifying `P2P` will run the `IMB-P2P` suite.. If not specified, the `IMB-MPI1` suite will be used. For the full list of available IMB variations use `./run_imb --help`.

options is the arguments to pass to IMB-MPI1. Specifying `--help` will provide help about the selected IMB suite. Specifying `--help option` will provide help about the given option within the selected IMB suite. For more information see https://software.intel.com/content/www/us/en/develop/documentation/imb-user-guide/.

For example:

```
./run_imb 4
./run_imb 16 allreduce barrier
./run_imb 4 P2P
./run_imb --help
./run_imb 1 -help
./run_imb 1 -help npmin
./run_imb 1 P2P --help
./run_imb 1 P2P --help msglog
```

NOTE

If desired the `run_imb` script may be edited to set `BASE_DIR` to the location of a different version of the IMB benchmark, such as the one included in the Intel® oneAPI release.

7.2.3 oneCCL Benchmarks (benchmark)

Use the `run_oneccl` sample script in `WORKSPACE/mpi_apps` to run the oneCCL `benchmark` example application.

1. Type `cd WORKSPACE/mpi_apps`
2. Type `./run_oneccl NP options`

   where:

   NP is the number of processes to run. Specifying `all` indicates the test should be run with `PROCS_PER_NODE` processes for every entry in the `MPI_HOSTS` file. Specifying `--help` will provide help about the `run_oneccl` script itself. A minimum of two processes is required, except when requesting help.

   options is the arguments pass to the oneCCL benchmark. Specifying `--help` will provide help about the oneCCL `benchmark` example application. For more information see https://www.intel.com/content/www/us/en/docs/oneccl/developer-guide-reference/.

For example:

```
./run_oneccl 4
./run_oneccl 16 -l allreduce
./run_oneccl --help
./run_oneccl 1 --help
```

NOTE

The oneCCL benchmark binary is not included in the Intel® Ethernet Fabric Suite FastFabric Toolset . Prior to using `run_oneccl` ensure the `WORKSPACE/mpi_apps/oneccl.param` file or the environment specifies the proper path to the `benchmark` program via `ONECCL_EXAMPLES_DIR`

7.3 Sample Test Applications

7.3.1 High Performance Linpack (HPL2)

This test is a standard benchmark for Floating Point Linear Algebra performance. Version 2.3 is provided. HPL requires a Basic Linear Algebra Subprograms library (BLAS). When using Intel MPI as part of Intel oneAPI, the Intel Math Kernel Library (MKL) will be used by default.. Otherwise the `openblas` library will be used.

NOTE

HPL2 is known to scale very well, and is the benchmark of choice for identifying a systems ranking in the Top 500 supercomputers (http://www.top500.org).. The focus of including it in sample applications is on the use of single node HPL as a hardware and software stress test, and to compare hosts within a cluster to identify outliers. This should not be used for top500 submissions or comparisons as there are other more optimized versions of this benchmark built with a targeted Basic Linear Algebra Subprograms (BLAS) library such as Intel MKL.

Prior to running this application, an `HPL.dat` file must be installed in `WORKSPACE/mpi_apps/hpl-2.3/bin/ICS.${ARCH}.${CC}` on all nodes. The `config_hpl2` script and some sample configurations are included.

The `config_hpl2` script can select from one of the assorted `HPL.dat` files in `WORKSPACE/mpi_apps/hpl-config`. These files are a good starting point for most clusters. To get a list of available pre-built `HPL.dat` files, run `./config_hpl2 --help`

```
 Usage: ./config_hpl2 [-l] config_name [problem_size]
    -l - only configure local file, default is to configure on all MPI_HOSTS
              or
  ./config_hpl2 --help
For example: ./config_hpl2 32t
either create hpl-config/HPL.dat-'config_name'
or select one of:
HPL.dat-128l  HPL.dat-18t   HPL.dat-2s    HPL.dat-32m   HPL.dat-64l
HPL.dat-128m  HPL.dat-1l    HPL.dat-2t    HPL.dat-32s   HPL.dat-64m
HPL.dat-128s  HPL.dat-1m    HPL.dat-300l  HPL.dat-32t   HPL.dat-64s
HPL.dat-128t  HPL.dat-1s    HPL.dat-300m  HPL.dat-4l    HPL.dat-64t
HPL.dat-16l   HPL.dat-1t    HPL.dat-300s  HPL.dat-4m    HPL.dat-8l
```

```
HPL.dat-16m   HPL.dat-256l  HPL.dat-300t  HPL.dat-4s    HPL.dat-8m
HPL.dat-16s   HPL.dat-256m  HPL.dat-320l  HPL.dat-4t    HPL.dat-8s
HPL.dat-16t   HPL.dat-256s  HPL.dat-320m  HPL.dat-512l  HPL.dat-8t
HPL.dat-18l   HPL.dat-256t  HPL.dat-320s  HPL.dat-512m
HPL.dat-18m   HPL.dat-2l    HPL.dat-320t  HPL.dat-512s
HPL.dat-18s   HPL.dat-2m    HPL.dat-32l   HPL.dat-512t
```

NOTE

When running on a cluster where `WORKSPACE/mpi_apps` is on a shared filesystem, use the `config_hpl2 -l` option. Otherwise `config_hpl2` must be run with the appropriate `MPI_HOSTS` file and `ethscpall` will be used to copy the selected `HPL.dat` file to all the hosts specified in `MPI_HOSTS`.

The problem sizes used assume a cluster with 1 GB of physical memory per processor. For each process count, four files are provided:

- `t` – A very small test run (5000 problem size)
- `s` – A small problem size on the low end of problem sizes
- `m` – A medium problem size
- `l` – A large problem size

For example, to quickly confirm that HPL2 runs with 16 processes in the `WORKSPACE/mpi_apps/mpi_hosts` file:

1. Type `./config_hpl2 16t`.

   This command edits the `HPL.dat` file on the local host for a 16 process "very small" test, and copies that file to all hosts in the `MPI_HOSTS` file.

2. Once the `HPL.dat` has been configured and copied, HPL2 can be run using the script.
   Type `cd WORKSPACE/mpi_apps`

3. Type `./run_hpl2 NP`

   where:

   NP is the number of processors for the run. Specifying `all` indicates the test should be run with `PROCS_PER_NODE` processes for every entry in the `MPI_HOSTS` file. Specifying `--help` will provide help about the `run_hpl2` script itself. The number of processes specified must be at least as large as the process count in the selected `HPL.dat` file.

   For example:

```
./run_hpl2 16
./run_hpl2 --help
```

For more information about HPL2, refer to the `README`, `TUNING`, and assorted HTML files in the `WORKSPACE/mpi_apps/hpl2-<version>` directory.

7.3.2 Performance Test

7.3.2.1 Latency/Bandwidth Deviation Test

This is an analysis/diagnostic tool to perform assorted pairwise bandwidth and latency tests and report pairs outside an acceptable tolerance range. The tool identifies specific nodes that have problems and provides a concise summary of results.

This tool is also used by the Intel® Ethernet Fabric Suite FastFabric Toolset `Check MPI performance` TUI menu item, see Checking MPI Performance. It can also be invoked using `ethhostadmin mpiperfdeviation`, see ethhostadmin.

Perform the following procedure to use the script provided to run this application:

1. Type `cd WORKSPACE/mpi_apps`
2. Type `./run_deviation` *`NP options`*

   where:

   `NP` is the number of processes to run. Specifying `all` indicates the test should be run with `PROCS_PER_NODE` processes for every entry in the `MPI_HOSTS` file. Specifying `--help` will provide help about the `run_deviation` script itself.

   `options` are the optional arguments to `run_deviation` and/or the `deviation` test program, as discussed below. Specifying `--help` will provide help about the `deviation` test program.

   For example:

```
./run_deviation 4
./run_deviation --help
./run_deviation 1 --help
```

This script runs a quick latency and bandwidth test against pairs of the hosts specified in `MPI_HOSTS` file. By default, each host is run against a single reference host and the results are analyzed. Pairs that have 20% less bandwidth or 50% more latency than the average pair are reported as failures.

NOTE

For this test, the `MPI_HOSTS` file should not list a given host more than once and `PROCS_PER_NODE` should be `1`, regardless of how many CPUs the host has. However when testing hosts with more than 1 NIC per host it may be beneficial to assign 1 process per NIC.

The tool can be run in a sequential or a concurrent mode. Sequential mode is the default and it runs each host against a reference host. By default, the reference host is selected based on the best performance from a quick test of the first 40 hosts.

In concurrent mode, hosts are paired up and all pairs are run concurrently. Since there may be fabric contention during such a run, any poor performing pairs are then rerun sequentially against the reference host.

Concurrent mode runs the tests in the shortest amount of time, however, the results could be slightly less accurate due to switch contention. In heavily oversubscribed fabric designs, if concurrent mode is producing unexpectedly low performance, try sequential mode.

run_deviation supports a number of parameters that allow for more precise control over the mode, benchmark, and pass/fail criteria.

```
'ff'      When specified, the configured FF_DEVIATION_ARGS will be used
 bwtol    Percent of bandwidth degradation allowed below Avg value
 lattol    Percent of latency degradation allowed above Avg value

Other deviation arguments:
      [bwbidir] [bwunidir] [-bwdelta MBs] [-bwthres MBs] [-bwloop count]
[-bwsize size][-latdelta usec] [-latthres usec] [-latloop count] [-latsize size]
[-c] [-b] [-v] [-vv] [-h reference_host]
-bwbidir   Perform a bidirectional bandwidth test
-bwunidir  Perform a unidirectional bandwidth test (default)
-bwdelta   Limit in MB/s of bandwidth degradation allowed below Avg value
    -bwthres   Lower Limit in MB/s of bandwidth allowed below Avg value
    -bwloop    Number of loops to execute each bandwidth test
    -bwsize    Size of message to use for bandwidth test
    -latdelta  Limit in usec of latency degradation allowed above Avg value
    -latthres  Upper Limit in usec of latency allowed
    -latloop   Number of loops to execute each latency test
    -latsize   Size of message to use for latency test
    -c         Run test pairs concurrently instead of the default of sequential
    -b         When comparing results against tolerance and delta use best
               instead of Avg
    -v         verbose output
    -vv        Very verbose output
    -h         Baseline host to use for sequential pairing
Both bwtol and bwdelta must be exceeded to fail bandwidth test
When bwthres is supplied, bwtol and bwdelta are ignored
Both lattol and latdelta must be exceeded to fail latency test
When latthres is supplied, lattol and latdelta are ignored

For consistency with OSU benchmarks MB/s is defined as 1000000 bytes/s

Examples:
   ./run_deviation 20 ff
   ./run_deviation 20 ff -v
   ./run_deviation 20 20 50 -c
   ./run_deviation 20 '' '' -c -v -bwthres 1200.5 -latthres 3.5
   ./run_deviation 20 20 50 -c -h compute0001
   ./run_deviation 20 0 0 -bwdelta 200 -latdelta 0.5
   ./run_deviation --help
   ./run_deviation 1 --help

  Example of 4 hosts with both 20% bandwidth and latency tolerances running in
concurrent mode using the verbose option with a specified baseline host.

  ./run_deviation 4 20 20 -c -v -h hostname
```

7.3.2.2 mpi_stress Test

This test can be used to place stress on the interconnect as part of verifying stability. The run_mpi_stress script can be used to run this application.

This MPI stress test program is designed to load an MPI interconnect with point-to-point messages while optionally checking for data integrity. By default, it runs with all-to-all traffic patterns, optionally including traffic with between the current process (-s), local processes on the same node (-i), and processes across the fabric. It can also be set up with multi-dimensional grid traffic patterns, and can be parameterized to run rings, open 2D grids, closed 2D grids, cubic lattices, hypercubes, and so forth. Optionally, the message data can be randomized and checked using CRC checksums (strong but slow) or XOR checksums (weak but fast). The communication kernel is

built out of non-blocking, point-to-point calls to load the interconnect. The program is not designed to exhaustively test different MPI primitives. Performance metrics are displayed, but may not be entirely accurate.

Usage

```
run_mpi_stress number_of_processes [mpi_stress_arguments]
```

or

```
./run_mpi_stress --help
```

Options

number_of_processes is the number of processes to run. Specifying `all` indicates the test should be run with `PROCS_PER_NODE` processes for every entry in the `MPI_HOSTS` file. Specifying `--help` will provide help about the `run_mpi_stress` script itself.

mpi_stress_arguments

- `-a` *INT* – Desired alignment for buffers (must be power of 2 and multiple of 8)
- `-A` – Test misaligned buffers by adjusting start addresses
- `-b` *BYTE* – Byte value to initialize non-random send buffers (otherwise, 0)
- `-c` – Enable CRC checksums
- `-D` *INT* – Set maximum data amount per message size (default is 1073741824)
- `-d` – Enable data checksums (otherwise, headers only)
- `-e` – Exercise the interconnect with random length messages
- `-E` *INT* – Set maximum number of data byte errors to show
- `-g` *INT* – Use INT-dimensional grid connectivity (non-periodic)
- `-G` *INT* – Use INT-dimensional grid connectivity (periodic) (default is to use all-to-all connectivity)
- `-h` or `--help` – Display this help page
- `-i` – Include local ranks as destinations (only for all-to-all)
- `-I` *INT* – Set message size increment (default power of 2)
- `-l` *INT* – Set minimum message size (default is 0)
- `-L` *INT* – Set minimum message count (default is 100)
- `-m` *INT* – Set maximum message size (default is 4194304)
- `-M` *INT* – Set maximum message count (default is 10000)
- `-n` *INT* – Number of times to repeat (default is 1)
- `-N` – Allocate buffers from multiple devices (only available when built for Intel GPU)
- `-O` – Show options and parameters used for the run.
- `-p` – Show progress

- `-P` – Poison receive buffers at init and after each receive
- `-q` – Quiet mode (don't show error details)
- `-r` – Fill send buffers with random data (else 0 or `-b` *`BYTE`*)
- `-R` – Round-robin destinations (default is random selection)
- `-s` – Include self as a destination (only for all-to-all)
- `-S` – Use non-blocking synchronous sends (MPI_Issend)
- `-t` *`INT`* – Run for INT minutes (implicitly adds `-n` *`BIGNUM`*)
- `-u` – Unidirectional traffic (only for grid)
- `-v` – Enable verbose mode (more `-v` for more verbose)
- `-w` *`INT`* – Number of send/receive in window (default is 20)
- `-x` – Enable XOR checksums
- `-z` – Enable typical options for data integrity (`-drx`) (for stronger integrity checking try using `-drc` instead)
- `-Z` – Zero receive buffers at init and after each receive
- `SEND_DEVICE` – Source buffer location for send, `H` will use CPU, `D` will use GPU. When built for GPU, default is `D`. When built for CPU, only `H` is allowed.
- `RECV_DEVICE` – Destination buffer location for receive, `H` will use CPU, `D` will use GPU. When built for GPU, default is `D`. When built for CPU, only `H` is allowed.

7.3.2.3 Latency Test

This test is a simple benchmark of end-to-end latency for various MPI message sizes. The values reported are one-direction latency.

Perform the following steps:

1. Type `cd WORKSPACE/mpi_apps`.
2. Type `./run_lat`.

This test runs assorted latencies from 0 to 256 bytes. To run a different set of message sizes, an optional argument specifying the maximum message size can be provided. Run `./run_lat --help` for more information.

This benchmark only uses the first two nodes listed in `MPI_HOSTS`. It is similar to `run_lat5`, which is preferred. This test can more easily be controlled and the script modified to aid detailed tuning, debug, or analysis of individual message sizes.

Related Links

OSU Micro-Benchmarks on page 196

7.3.2.4 Bandwidth Test

This test is a simple benchmark of maximum unidirectional bandwidth.

Perform the following steps:

1. Type `cd WORKSPACE/mpi_apps`

2. Type `./run_bw`

This test runs assorted bandwidths from 4 KB to 4 MB. To run a different set of message sizes, an optional argument specifying the maximum message size can be provided. Run `./run_bw --help` for more information.

This benchmark only uses the first two nodes listed in `MPI_HOSTS`. It is similar to `run_bw5`, which is preferred. This test can more easily be controlled and the script modified to aid detailed tuning, debug, or analysis of individual message sizes.

Related Links

OSU Micro-Benchmarks on page 196

7.3.3 MPI Fabric Stress Tests

These sample applications are designed to stress parts of a cluster to help ensure that the fabric is working properly. Although they report measurement data similar to other bandwidth applications, they are not intended to be benchmarking tools. Instead, they should be used to identify potential stability or performance issues in the fabric, such as bad cables.

7.3.3.1 All NIC Latency

The All NIC Latency test is a specialized stress test for large fabrics. It iterates through every possible pairing of the NICs in the fabric, and performs a latency test on each pair. At the end of each combination, the test reports the fastest and slowest pairs. This test has no real value as a performance benchmark, but is extremely useful for checking for cabling problems in the fabric. A script is provided to run this application. It requires no arguments, but can take several options if needed. To run with no arguments, follow these steps:

1. Change directory to `WORKSPACE/mpi_apps`.

   ```
   cd WORKSPACE/mpi_apps
   ```

2. Run the All NIC Latency test

   ```
   ./run_allniclatency
   ```

This test runs a 60 second test on all the hosts listed in the `MPI_HOSTS` file.

To change the default behavior, specify `-c` or `-v` and/or up to three optional arguments, for example:

```
./run_allniclatency [-c] [-v] [NP [MN [SS]]]
```

or

```
./run_allniclatency --help
```

Where:

- `-h` or `--help` - Provides help about `run_allniclatency`.
- `-c` or`--csv` - Prints all raw test results in CSV file format, into the application logfile. Useful for analyzing the raw results with a spreadsheet application.
- `-v` or `--verbose` - Runs the test in a verbose mode that shows more information.

- *NP* is the number of processes to run. Specifying `all` indicates the test should be run with `PROCS_PER_NODE` processes for every entry in the `MPI_HOSTS` file. Default `all`. It is recommended to use `PROCS_PER_NODE=1` for this test. However when testing hosts with more than 1 NIC per host it may be beneficial to assign 1 process per NIC.
- *MN* is the number of minutes the test should run.
- *SS* is the size of the messages in bytes to use when testing (between 1 and 4194304).

For example, to run a 30 minute test on 64 nodes with 4 KB messages, the following command would be used from the `WORKSPACE/mpi_apps` directory:

```
./run_allniclatency 64 30 4096
```

Once 30 minutes has elapsed, the test completes when the current round of testing is done.

If you want the tests to repeat indefinitely, set the duration to `infinite` as shown in the following CLI command:

```
./run_allniclatency 64 infinite 4096
```

To use the results of this test, look for nodes that are often listed as the slowest at the end of the round. One of those nodes may have a cabling problem, or there may be a congested switch to switch link causing those nodes to experience degraded performance.

7.3.3.2 run_cabletest

The `run_cabletest` tool is a specialized stress test for large fabrics. It groups MPI ranks into sets that are tested against other members of the set. This test has no real value as a performance benchmark, but is extremely useful for checking for cabling problems in the fabric.

`./run_cabletest` requires no arguments, but does require you to generate a group hosts file. This is done with the `gen_group_hosts` script. The name of the group hosts file is specified by the `$MPI_GROUP_HOSTS` variable, and defaults to `mpi_group_hosts`. For more information, refer to `gen_group_hosts`.

By default, `run_cabletest` runs for 60 minutes and uses 4 MB messages. These settings can be changed by using the three optional arguments: duration, smallest message size, and largest message size. The arguments are specified in order:

1. Change directory to `WORKSPACE/mpi_apps`.

   ```
   cd WORKSPACE/mpi_apps
   ```

2. Run the `run_cabletest` test including the duration in minutes, the smallest message size, and the largest message size.

   `./run_cabletest` *dd ss ll*

   where:

   - *dd* is the duration in minutes.

- *ss* is the smallest message size.
- *ll* is the largest message size.

For example, to run a one minute test with 4 MB messages, enter the following CLI command:

```
./run_cabletest 1
```

Once one minute has elapsed, the test completes when the current round of testing is done.

If you want the tests to repeat indefinitely, set the duration to `infinite`, as shown in the following CLI command:

```
./run_cabletest infinite
```

In addition to the duration, you can specify the smallest and largest messages to send. The messages must be between 16384 and 4194304 (4 MB). The following example tests message sizes between 1 and 4 MB, and runs for 24 hours:

```
./run_cabletest 1440 1048576 4194304
```

The following options are available:

- `-h` / `--help` – Provides this help text.
- `-v` / `--verbose` – Runs the test in a verbose mode that shows you how the nodes were grouped.

7.3.3.3 run_batch_cabletest

The `run_batch_cabletest` in `WORKSPACE/mpi_apps` makes it easier to run the `run_cabletest` stress test (see `run_cabletest` on page 206). The `run_batch_cabletest` script runs separate jobs for each `BATCH_SIZE` hosts, and can generate the `mpi_group_hosts` files needed using a single `mpi_hosts` file, which lists each host to be tested once, in topology order. For many clusters, ethsorthosts may help put a list of hosts in topology order, or ethfindgood may be used to identify candidate hosts. By using many small jobs, the impact of any individual host issues (host crash, hang, and so on) during the test is limited to one batch of hosts.

NOTE

When using `run_batch_cabletest`, the log files are separated. Each individual job gets its own log file, with a suffix to the log filename indicating the run number within the set of batches. For example: `cabletest.04Jan12165901.1` `cabletest.04Jan12165901.2` This avoids any intermingling of output from multiple runs in a single log file.

By default, `run_batch_cabletest` runs for 60 minutes and uses 4 MB messages. These settings can be changed by using the three optional arguments: duration, smallest message size, and largest message size. The arguments are specified in order:

1. Change directory to `WORKSPACE/mpi_apps`.

   ```
   cd WORKSPACE/mpi_apps
   ```

2. Run the `run_batch_cabletest` test including the duration in minutes, the smallest message size, and the largest message size.

   ```
   ./run_batch_cabletest [duration [minmsg [maxmsg]]]
   ```

   where:

   - *`duration`* is the duration in minutes and can be `infinite`
   - *`minmsg`* is the smallest message size. Must be between 16384 and 4194304.
   - *`maxmsg`* is the largest message size. Must be between 16384 and 4194304.

This builds a set of `mpi_hosts.#` and `mpi_group_hosts.#` files, with no more than `BATCH_SIZE` hosts each. If an odd number of hosts appears in `mpi_hosts`, the last one is skipped.

For example, to run a one minute batch test, with 4-megabyte messages, enter the following CLI command:

```
./run_batch_cabletest 1
```

Once one minute has elapsed, the batch test completes when the current round of testing completes.

If you want the tests to repeat indefinitely, set the duration to `infinite`, as shown in the following CLI command:

```
./run_batch_cabletest infinite
```

In addition to the duration, you can specify the smallest and largest messages to send. This example batches test message sizes between 1 and 4 MB, and runs for 24 hours:

```
./run_batch_cabletest 1440 1048576 4194304
```

The following options are available:

- `-h` / `--help` – Provides this help text.
- `-v` / `--verbose` – Runs the test in a verbose mode that shows you how the nodes were grouped.
- `-n` – Specifies the number of processes to run per host.
- *`duration`* – Specifies how many minutes to run. Default is 60.
- *`minmsg`* – Specifies the smallest message to use. Must be between 16384 and 4194304.
- *`maxmsg`* – Specifies the largest message to use. Must be between 16384 and 4194304.

Default *`minmsg`* and *`maxmsg`* is 4 MB.

Each run_cabletest MPI job has its output saved to a corresponding `/tmp/nohup.#.out` file.

Environment Variables

- `MPI_HOSTS` - The MPI hosts file to use. The default is `$PWD/mpi_hosts`. This file lists the hosts in topology order, one entry per host. The hosts are paired sequentially (first and second, third and fourth, and so on).
- `BATCH_SIZE` - The maximum hosts per MPI job. The default is 18, and the number must be even.

Examples

```
./run_batch_cabletest
MPI_HOSTS=good ./run_batch_cabletest 1440
BATCH_SIZE=16 MPI_HOSTS=good ./run_batch_cabletest infinite
```

7.3.3.4 gen_group_hosts

This tool generates an `mpi_group_test` file for use with `run_cabletest`. The `gen_group_hosts` tool asks three questions that need to be answered in order for it to generate the `mpi_group_hosts` file.

1. Enter the name of your hosts file.

   The hosts must be listed in this file in group order, with one host per line. The hosts cannot be listed more than once and must be listed in their physical order. The default hosts file is `WORKSPACE/mpi_apps/mpi_hosts`.

2. How big are your groups?

   For example, if you want to test each node against the node next to it, use 2 as the group size. If you want to test the nodes connected to one leaf switch against the nodes on another leaf switch, and you have 16 nodes per leaf, use 32 as the group size. The default group size is 2.

3. How many processes per node do you wish to run?

   The higher the number, the higher the link utilization. The number must be between 1 and the number of processors per node. The default number of processes per node is 3. Using more processes than needed to saturate the link does not improve testing.

After all questions are answered, the `WORKSPACE/mpi_apps/mpi_group_hosts` file is generated.

If the number of the hosts is not a multiple of the group size, a warning is shown.

7.3.3.5 run_multibw

`run_multibw` runs `mpi_multibw`, which performs a multi-core pairwise bandwidth test. `mpi_multibw` is based on osu_bw and osu_multi_lat (see OSU Micro-Benchmarks).

1. Change directory to `WORKSPACE/mpi_apps`.

   ```
   cd WORKSPACE/mpi_apps
   ```

2. Run the run_multibw test including the number of processes on which to run the test.

   ./run_multibw *processesoptions*

   where:

   - *processes* - Specifies the number of processes on which to run the test. Specifying all indicates the test should be run with PROCS_PER_NODE processes for every entry in the MPI_HOSTS file. Specifying --help will provide help about the run_multibw script itself.
   - *options* is the arguments pass to mpi_multibw. Specifying -h will provide help about mpi_multibw.

7.4 MPI Batch run_* Scripts

The run_batch_script makes it easier to run other run_* scripts as many smaller jobs. This script is located in WORKSPACE/mpi_apps and runs separate jobs for each BATCH_SIZE host. By using many, small jobs, the impact of any individual host issues (host crash, hang, etc.) during the test is limited to one batch of hosts.

NOTE

When using run_batch_script, the log files are separated. Each individual job gets its own log file with a suffix to the log filename indicating the run number within the set of batches. For example, mpi_groupstress.04Jan12165901.1 mpi_groupstress.04Jan12165901.2 This scheme avoids any intermingling of output from multiple runs in a single log file.

Usage

```
./run_batch_script [-e] run_script [args]
```

or

```
./run_batch_script --help
```

Options

- -e – Forces an even number of hosts in the final batch by skipping the last one.
- run_script – Specifies a run_* script from this directory
- *args* – Specifies arguments for run_script. If the first argument is NP, it is replaced with the process count.

  This builds a set of mpi_hosts.# files with no more than BATCH_SIZE hosts each. If -e is specified and an odd number of hosts appear in mpi_hosts, the last one is skipped. Each run_script MPI job has its output saved to a corresponding/tmp/nohup.#.out file

  This script may only be used for scripts that use MPI_HOSTS.

To run run_cabletest, use run_batch_cabletest.

Environment Variables

- `MPI_HOSTS` – The MPI hosts file to use. Default is `mpi_hosts`.
- `BATCH_SIZE` – Maximum hosts per MPI job. The default is 18. If `-e` is used, the number must be even.
- `MIN_BATCH_SIZE` – Minimum hosts per MPI job. The default is 2. If `-e` is used, the number must be even.

The following environment variables are supported in individual `run_* scripts`:

- `SHOW_MPI_HOSTS` – Set to `y` if MPI_HOSTS contents should be output prior to starting job.
- `SHOW_MPI_HOSTS_LINES` – Set to the maximum number of lines in hosts file.

Examples

```
./run_batch_script run_deviation NP ff
BATCH_SIZE=2 MPI_HOSTS=good ./run_batch_script run_lat2
BATCH_SIZE=16 MPI_HOSTS=good ./run_batch_script run_deviation ff
MIN_BATCH_SIZE=16 BATCH_SIZE=16 ./run_batch_script run_hpl2 16
```