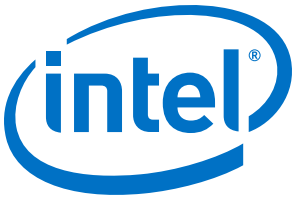


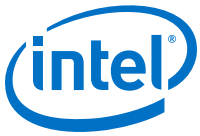
Intel® Omni-Path Fabric Suite FastFabric Command Line Interface

Reference Guide

Rev. 5.0

December 2016

Order No.: H76472-5.0



You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or visit http://www.intel.com/design/literature.htm.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at http://www.intel.com/ or from the OEM or retailer.

No computer system can be absolutely secure.

Intel, the Intel logo, Intel Xeon Phi, and Xeon are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2015–2016, Intel Corporation. All rights reserved.

Revision History

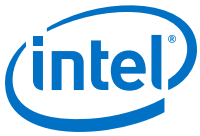
For the latest documentation, go to http://www.intel.com/omnipath/FabricSoftwarePublications.

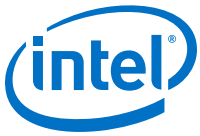
Date	Revision	Description
December 2016	5.0	Updates to this document include: • Added Cluster Configurator for Intel® Omni-Path Fabric to Preface. • Added opaextractmissinglinks • Globally, updated filepath from /opt/opa to /usr/lib/opa. • Added note to Overview that you must have root privilege to run FastFabric commands. • Added new section Utilization under Port Counters containing the following: — PortXmitData & PortVLXmitData[n] — PortRcvData & PortVLRcvData[n] — PortMulticastXmitPkts — PortMulticastRcvPkts • Moved UncorrectableErrors Counter and FMConfigErrors Counter from Other [Port Counters] section to Link Integrity. • Added PortXmitDiscards to Other [Port Counters] section. • Updated the following commands: — opa_osd_exercise — opadisableports — opafequery — opagetvf_env — opahostadmin — opalinkanalysis — opapaquery — opareport — opareports — opaswitchadmin — opaverifyhosts — opaxlattopology — opaxlattopology_cust
August 2016	4.0	Document has been updated as follows: • Updated the following commands: — opachassisadmin — opareport — oparesolvehfiport — opasetupssh — opaswitchadmin — opaconfig • Included text for various commands that previously referenced content in: — opafmcmd — opareport • Changed port counter from "SwPortCongestion" to "CongDiscards". • Updated topology.xlsx Overview.

continued...

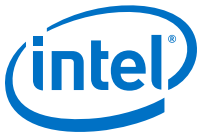
Date	Revision	Description
May 2016	3.0	Document has been updated as follows: • Modified opaxlattopology. • Added opaledports. • Updated options for the following queries: — opafequery — opapaquery — opasaquery — opasmaquery — opapmaquery • Modified opaverifyhosts to include note on copying the hostverify.sh before editing or using. • Added note pertaining to quarantined nodes to Snapshots • Modified opaswitchadmin.
February 2016	2.0	Document has been updated as follows: • Reorganized CLI commands for better usability. • Added opatmmtool • Added Sample Files. • Added Map of Intel® Omni-Path Architecture Commands.
November 2015	1.0	Document has been updated.

Contents

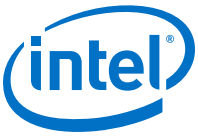




Figures



Tables



Preface

This manual is part of the documentation set for the Intel® Omni-Path Fabric (Intel® OP Fabric), which is an end-to-end solution consisting of Intel® Omni-Path Host Fabric Interfaces (HFIs), Intel® Omni-Path switches, and fabric management and development tools.

The Intel® OP Fabric delivers a platform for the next generation of High-Performance Computing (HPC) systems that is designed to cost-effectively meet the scale, density, and reliability requirements of large-scale HPC clusters.

Both the Intel® OP Fabric and standard InfiniBand* are able to send Internet Protocol (IP) traffic over the fabric, or *IPoFabric*. In this document, however, it is referred to as *IP over IB* or *IPoIB*. From a software point of view, IPoFabric and IPoIB behave the same way and, in fact, use the same `ib_ipoib` driver to send IP traffic over the ib0 and/or ib1 ports.

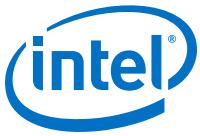
Intended Audience

The intended audience for the Intel® Omni-Path (Intel® OP) document set is network administrators and other qualified personnel.

Documentation Set

The complete end user publications set for the Intel® Omni-Path product includes the following items.

- Hardware Documents:
  - *Intel® Omni-Path Fabric Switches Hardware Installation Guide*
  - *Intel® Omni-Path Fabric Switches GUI User Guide*
  - *Intel® Omni-Path Fabric Switches Command Line Interface Reference Guide*
  - *Intel® Omni-Path Edge Switch Platform Configuration Reference Guide*
  - *Intel® Omni-Path Fabric Managed Switches Release Notes*
  - *Intel® Omni-Path Fabric Externally-Managed Switches Release Notes*
  - *Intel® Omni-Path Host Fabric Interface Installation Guide*
- Software Documents:
  - *Intel® Omni-Path Fabric Software Installation Guide*
  - *Intel® Omni-Path Fabric Suite Fabric Manager User Guide*
  - *Intel® Omni-Path Fabric Suite FastFabric User Guide*
  - *Intel® Omni-Path Fabric Host Software User Guide*
  - *Intel® Omni-Path Fabric Suite Fabric Manager GUI Online Help*
  - *Intel® Omni-Path Fabric Suite Fabric Manager GUI User Guide*



- *Intel® Omni-Path Fabric Suite FastFabric Command Line Interface Reference Guide*
- *Intel® Performance Scaled Messaging 2 (PSM2) Programmer's Guide*
- *Intel® Omni-Path Fabric Performance Tuning User Guide*
- *Intel® Omni-Path Host Fabric Interface Platform Configuration Reference Guide*
- *Intel® Omni-Path Fabric Software Release Notes*
- *Intel® Omni-Path Fabric Manager GUI Release Notes*
- *Intel® Omni-Path Storage Router Design Guide*
- *Building Lustre* Servers with Intel® Omni-Path Architecture Application Note*
- *Intel® Omni-Path Fabric Staging Guide*

Documents are available at the following URLs:

- Intel® Omni-Path Switches Installation, User, and Reference Guides

  http://www.intel.com/omnipath/SwitchPublications
- Intel® Omni-Path Host Fabric Interface Installation, User, and Reference Guides (includes software documents)

  http://www.intel.com/omnipath/FabricSoftwarePublications
- Drivers and Software (including Release Notes)

  http://www.intel.com/omnipath/Downloads

Cluster Configurator for Intel® Omni-Path Fabric

The Cluster Configurator for Intel® Omni-Path Fabric is available at: http://www.intel.com/content/www/us/en/high-performance-computing-fabrics/omni-path-configurator.html.

This tool generates sample cluster configurations based on key cluster attributes, including a side-by-side comparison of up to four cluster configurations. The tool also generates parts lists and cluster diagrams.

Documentation Conventions

The following conventions are standard for Intel® Omni-Path 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 (jump) to a figure, table, or section in this guide. Links to websites are also shown in blue. For example:

  See License Agreements on page 14 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® Omni-Path Fabric 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:

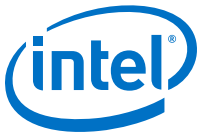
- **(Linux)** – Tasks are only applicable when Linux* is being used.
- **(Host)** – Tasks are only applicable when Intel® Omni-Path Fabric Host Software or Intel® Omni-Path Fabric Suite is being used on the hosts.
- **(Switch)** – Tasks are applicable only when Intel® Omni-Path Switches or Chassis are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

License Agreements

This software is provided under one or more license agreements. Please 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

Technical support for Intel® Omni-Path products is available 24 hours a day, 365 days a year. Please contact Intel Customer Support or visit www.intel.com for additional detail.



1.0 Introduction

This manual describes the command line interface (CLI) for the Intel® Omni-Path Fabric Suite FastFabric.

1.1 Overview

The FastFabric Toolset provides numerous powerful features; however, the rich set of capabilities can be overwhelming. This reference guide is organized for ease of use at all levels of understanding.

The FastFabric Toolset is installed in directories that are part of the standard Linux* root PATH. Most of the tools are installed in `/sbin`. For details, refer to *Intel® Omni-Path Fabric Software Installation Guide*.

Note: To run FastFabric commands, you must have root privilege.

1.2 Common Tool Options

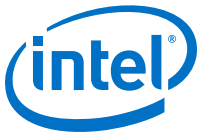
The following table lists CLI options that are applicable to most of the tools.

Table 1. 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 `opascpall`), 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 `opahostadmin`, `opachassisadmin`, and `opaswitchadmin`), parallel operation is the default mode. Parallel operation can also be disabled by setting `FF_MAX_PARALLEL` to `1`.
`-S`	Prompts for password for admin on chassis or root on host. By default, Intel® Omni-Path Fabric Suite FastFabric toolset operations against Intel® Omni-Path Chassis (such as `opacmdall`, `opacaptureall`, and `opachassisadmin`) obtain the chassis admin password from the `FF_CHASSIS_ADMIN_PASSWORD` environment variable which may be directly exported or part of `opafastfabric.conf`. Alternatively, you can use the `-S` option to be interactively prompted for the chassis admin password. The password is prompted for once, and the same password is then used to log in to each chassis during the operation. For hosts, this option is only applicable to `opasetupssh`.

continued...

Command	Description
	Note: All versions of Intel® Omni-Path Chassis firmware permit SSH keys to be configured within the chassis for secure password-less login. In this case, there is no need to configure a `FF_CHASSIS_ADMIN_PASSWORD` environment variable, and `FF_CHASSIS_LOGIN_METHOD` can be set to SSH. Intel recommends you set up a secure SSH password-less login using `opasetupssh -C`. Refer to the *Intel® Omni-Path Fabric Switches GUI User Guide* for more information.
`-C`	Specifies that the given operation should be performed against chassis. By default, many Intel® Omni-Path Fabric Suite FastFabric toolset operations are performed against hosts. However, selected FastFabric toolset commands (such as `opacmdall`, `opapingall`, and `opacaptureall`) can also operate against Intel® Omni-Path internally-managed chassis. When `-C` is specified, the operation is performed against chassis instead of hosts. Refer to Selection of Devices for details about the selection of chassis.
`-h`	Select which local HFI to use.
`-p`	Select which local HFI port to use.
`-v`	Produces verbose output.



2.0 Selection of Devices

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/sysconfig/opa/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/sysconfig/opa/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.

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/sysconfig/opa/hosts`.

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

- `/etc/sysconfig/opa/hosts-mpi` – list of MPI hosts
- `/etc/sysconfig/opa/hosts-fs` – list of file server hosts
- `/etc/sysconfig/opa/hosts` – list of all hosts except for the FastFabric toolset node
- `/etc/sysconfig/opa/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
include /etc/sysconfig/opa/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.

Hosts may be specified by IP address or a resolvable TCP/IP host name. Typically, host names are used for readability. Also, some FastFabric toolset commands translate the supplied host names to IPoIB hostnames, in which case names are generally easier to translate than numeric IP addresses. Typically management network hostnames are specified. However, if desired, IPoIB hostnames or 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/sysconfig/opa`.

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.

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'`

2.2 Selection of Chassis

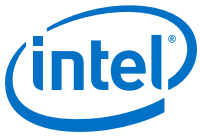
Note: In this document, the term *chassis* refers to a managed switch.

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

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

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

1. `-H` option
2. `CHASSIS` environment variable



3. `-F` option
4. `CHASSIS_FILE` environment variable
5. `/etc/sysconfig/opa/chassis` file

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

2.2.1 Chassis List Files

You can use the `-F` option to provide the name of a file containing the list of chassis on which to operate. The default is `/etc/sysconfig/opa/chassis`.

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

- `/etc/sysconfig/opa/chassis-core`: list of core switching chassis
- `/etc/sysconfig/opa/chassis-edge`: list of edge switching chassis
- `/etc/sysconfig/opa/esm_chassis`: list of chassis running an SM
- `/etc/sysconfig/opa/chassis`: list of all chassis

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

Chassis List File Format

Sample chassis file:

```
# this is a comment
192.168.0.5   # chassis IP address
edge1   # chassis resolvable TCP/IP name
include /etc/sysconfig/opa/chassis-core  # included file
```

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

A chassis may be specified by chassis 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/sysconfig/opa`.

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

The chassis file can also be generated using the `opagenchassis` command.

2.2.2 Explicit Chassis Names

When chassis are explicitly specified using the -H option or the CHASSIS environment variable, a space-separated list of names (or IP addresses) may be supplied. For example: -H *chassis1 chassis2 chassis3* .

2.2.3 Selection of Slots within a Chassis

Typically, operations are performed against the primary management module (MM) in the chassis. For operations such as opacmdall, you can specify the management module for the given chassis, if there is a redundant/secondary MM.

To perform operations against a specific subset of cards within the chassis, you can augment the chassis IP address or name within a chassis list or a chassis file with a list of slot numbers on which to operate. Use the form:

```
chassis:slot1,slot2,…
```

For example:

```
i9k229:0
i9k229:0,1,5
192.168.0.5:0,1,5
```

Note: No spaces can be used within the chassis name and slot list.

This format may be used whenever a chassis name or IP address is valid, such as the -H option, the CHASSIS environment variable, or chassis list files.

The slot number specified may be ignored on some operations.

Only slots containing MM may be specified with this format. Use the chassisQuery command to identify MM slots.

Note: For any operation, be careful that a given chassis is listed only once with all relevant slots. This prevents conflicting concurrent operations against a given chassis.

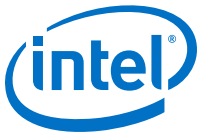
2.3 Selection of Switches

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

- On the command line, using the -N 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 -L option or the SWITCHES_FILE environment variable to specify a file containing the set of switches. Useful for groups of switches that are used often. The file is located here: /etc/sysconfig/opa/switches by default. The file must list all switches in the cluster.

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

1. -N option



2. `SWITCHES` environment variable
3. `-L` option
4. `SWITCHES_FILE` environment variable
5. `/etc/sysconfig/opa/switches` file

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

2.3.1 Switch List Files

You can use the `-L` option to provide the name of a file containing the list of switches on which to operate. The default is `/etc/sysconfig/opa/switches`.

It may be useful to create multiple files in `/etc/sysconfig/opa` representing different subsets of the fabric.

If a relative path is specified for the `-L` option or `SWITCHES_FILE` environment variable, the current directory is checked first, followed by `/etc/sysconfig/opa/`.

Switch List File Format

Sample switch list file:

```
# this is a comment
0x00117500d9000138,i9k138  # Node GUID with desired Name
0x00117500d9000139,i9k139  # Node GUID with desired Name
0x00117500d9000140:1:2,i9k140  # Node GUID with port and Name
0x00117500d9000141,i9k141,1 # Node GUID with desired Name, short distance
0x00117500d9000142,i9k142,5 # Node GUID with desired Name, longer distance
include /etc/sysconfig/opa/moreswitches  # included file
```

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

Switches can be specified by node GUID, optionally followed by a colon and the hfi:port, optionally followed by a comma and the Node Description (nodename) to be assigned to the switch, and optionally followed by the distance value indicating the relative distance from the FastFabric node for each switch.

You can use `opagenswitches` to locate externally-managed switches in the fabric and generate a `switches` file. By default, `opagenswitches` provides the proper distance value relative to the FastFabric node from which it was run. Alternatively, the `opagenswitches -R` option suppresses generation of this field.

When you use `opagenswitches` in conjunction with a topology file created during fabric design, you can associate switch names in the topology file with NodeGUIDs of the actual devices. This facilitates subsequent use of `opaswitchadmin` to configure the node descriptions for all switches according to the fabric design plan.

In a typical pure fat tree topology with externally managed switches as edge switches and internally managed switches as core switches, you can also manually specify proper distance by simply specifying 1 for the distance value of the switch next to the FastFabric node. Note that in such a topology, all other Edge switches are an equal

length from the FastFabric node and a missing distance value causes them to be treated as having a distance value which is larger than any other found in the file. Therefore, the other switches would be rebooted first and the FastFabric node's switch would be rebooted last.

The GUID is used to select the switch and, on firmware update operations, the node description is written to the switch such that other FastFabric tools (such as `opaquery` and `opareport`) can provide a more easily readable name for the switch. The node description can also be updated as part of switch basic configuration.

The hfi:port may be used to specify which local port (subnet) to use to access the switch. If this is omitted, all local ports specified are checked for the switch and the first port found to be able to access the switch is used to access it. See the *Intel® Omni-Path Fabric Suite FastFabric User Guide* for more information about how to specify the hfi:port value.

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 within the current directory first, then `/etc/sysconfig/opa`.

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

Intel recommends that a unique node description is specified for each switch. This name should follow typical naming rules and use the characters a-z, A-Z, 0-9, and underscore. No spaces are allowed in the node description. Additionally, names should not start with a digit.

For externally-managed switches, the node GUID can be found on a label on the bottom of the switch. Alternately, the node GUIDs for switches in the fabric can be found using a command such as:

```
opasaquery -t sw -o nodeguid
```

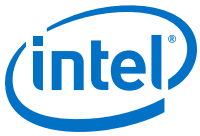
Note: The `opasaquery` command reports all switch node GUIDs, including those of internally-managed chassis such as the Intel® Omni-Path Switch 100 Series. GUIDs for internally-managed chassis cannot be used in the `switches` file.

2.3.2 Explicit Switch Names

When switches are explicitly specified using the `-N` option or the `SWITCHES` environment variable, a space-separated list of GUIDs (optionally with `hfi:port` and/or name) may be supplied. For example: `-N '0x00117500d9000138,i9k138 0x00117500d9000139,i9k139'`

2.4 Selection of Local Ports (Subnets)

Many commands permit a specific set of local Intel® Omni-Path Host Fabric Interface (HFI) ports to be used for fabric access. For example, `opareports`, `opafabricinfo`, `opaswitchadmin`, `opafabricanalysis`, and `opaallanalysis`. The default is to



use the first active port. However, for Fabric Management nodes connected to more than one subnet, you must specify the local HFI and port so that the desired subnet is analyzed.

You can specify the local ports on which to operate using one of the following methods:

- On the command line, using the `-p` option.
- Using the environment variable `PORTS` to specify a space-separated list of ports. Useful when multiple commands are performed against the same small set of ports.
- Using the `-t` option or the `PORTS_FILE` environment variable to specify a file containing the set of ports. Useful for groups of ports that are used often. The file is located here: `/etc/sysconfig/opa/ports` by default. The file must list all local ports connected to unique subnets.

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

1. `-p` option
2. `PORTS` environment variable
3. `-t` option
4. `PORTS_FILE` environment variable
5. `/etc/sysconfig/opa/ports` file
6. Default of the first active port on system. (0:0 port specification)

For example, if the `-p` option is used and the `PORTS_FILE` environment variable is also exported, the command operates only on ports specified using the `-p` option.

2.4.1 Port List Files

You can use the `-t` option or the `PORTS_FILE` environment variable to provide the name of a file containing the list of local HFI ports to use. The default is `/etc/sysconfig/opa/ports`.

It may be useful to create multiple files in `/etc/sysconfig/opa` representing different subsets of the ports. For example:

- `/etc/sysconfig/opa/ports-primary` - ports for which this node is primary
- `/etc/sysconfig/opa/ports-plane1` - port(s) for plane1 subnet
- `/etc/sysconfig/opa/ports` - list of all unique subnet ports

If a relative path is specified for the `-t` option or `PORTS_FILE` environment variable, the current directory is checked first, followed by `/etc/sysconfig/opa/`.

Port List File Format

Note: Intel® Omni-Path Host Fabric Interface has 1 port.

Sample port list file:

```
# this is a comment
1:1 # first port on 1st HFI
2:1 # first port on 2nd HFI
3:0 # first active port on 3rd HFI
include /etc/sysconfig/opa/ports-plane2 # included file
```

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

Ports are specified as `hfi:port`. No spaces are permitted. The first HFI is 1 and the first port is 1. The value `0` for HFI or port has special meaning. The allowed formats are:

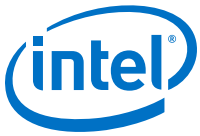
```
0:0 = 1st active port in system
0:y = port y within system
x:0 = 1st active port on HFI x
x:y = HFI x, port y
```

Files to be included may be specified using an `include` directive followed by a file name. File names specified should be absolute pathnames. If relative pathnames are used, they are searched for within the current directory first, then `/etc/sysconfig/opa`.

Comments may be placed on any line by using a `#` to precede the comment. On lines with a port or `include` directive, the `#` must be white space-separated from any preceding port or included filename.

2.4.2 Explicit Ports

When ports are explicitly specified using the `-p` option or the `PORTS` environment variable, a space-separated list of ports may be supplied. For example: `-p '1:1 2:1'`.



3.0 Descriptions of Command Line Tools

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

3.1 High-Level TUIs

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

3.1.1 opafastfabric

(Switch and Host) Starts the top-level Intel® Omni-Path Fabric Suite FastFabric Text User Interface (TUI) menu to enable setup and configuration. Refer to *Intel® Omni-Path Fabric Suite FastFabric User Guide* for additional details.

Syntax

```
opafastfabric
```

Options

None.

Example

```
#opafastfabric
 Intel FastFabric OPA Tools
 Version: X.X.X.X.X

     1) Chassis Setup/Admin
     2) Externally Managed Switch Setup/Admin
     3) Host Setup
     4) Host Verification/Admin
     5) Fabric Monitoring

     X) Exit
```

3.1.2 opatop

Starts the Fabric Performance Monitor (`opatop`) Text User Interface (TUI) menu to display performance, congestion, and error information about a fabric. Refer to *Intel® Omni-Path Fabric Suite FastFabric User Guide* for additional details.

Syntax

```
opatop [-v] [-q] [-h hfi] [-p port] [-i seconds]
```

Options

`--help`	Produces full help text.
`-v/--verbose level`	Specifies the verbose output level. Value is additive and includes: `1` Screen `4` STDERR opatop `16` STDERR PaClient
`-q/--quiet`	Disables progress reports.
`-h/--hfi` *`hfi`*	Specifies the HFI, numbered 1..n. Using 0 specifies that the `-p` *`port`* port is a system-wide port number. (Default is 0.)
`-p/--port` *`port`*	Specifies the port, numbered 1..n. Using 0 specifies the first active port. (Default is 0.)
`-i/--interval` *`seconds`*	Interval in *`seconds`* at which PA queries are performed to refresh to the latest PA image. Default = 10 seconds.

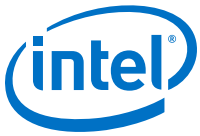
-h and -p options permit a variety of selections:

`-h 0`	First active port in system (default).
`-h 0 -p 0`	First active port in system.
`-h` *`x`*	First active port on HFI *x*.
`-h` *`x`* `-p 0`	First active port on HFI *x*.
`-h 0 -p` *`y`*	Port *y* within system (no matter which ports are active).
`-h` *`x`* `-p` *`y`*	HFI *x*, port *y*.

3.2 Health Check and Baselining Tools

(All) 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, fabric topology, and SM configuration. The tools are designed to permit easy manual execution or automated execution using `cron` or other mechanisms. The health check tools include:

- `opafabricanalysis` – Performs fabric topology and PMA error counters analysis.



- `opachassisanalysis` – Performs chassis configuration and health analysis for selected chassis.
- `opaesmanalysis` – Performs embedded SM configuration and health analysis for selected chassis.
- `opahostsmanalysis` – Performs host SM configuration and health analysis for the local host.
- `opaallanalysis` – Performs analysis on all components or a subset of components. Intel recommends this as the primary tool for general analysis.

3.2.1 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

3.2.2 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/opa/analysis/baseline is set through /etc/sysconfig/opa/opafastfabric.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

```
opafabricanalysis -e
```

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

```
opafabricanalysis -b
```

The baseline configuration is saved to *FF_ANALYSIS_DIR*/baseline. This includes files starting with links and comps, which are the results of opareport -o links and opareport -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 opareport can be used to verify the fabric deployment against the intended design.

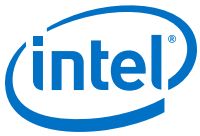
Once a good baseline has been established, use the tools to compare the present fabric against the baseline and check its health.

```
opafabricanalysis
```

Checks the present fabric links and components against the previous baseline. If there have been changes, it reports a failure and indicate which files hold the resulting snapshot and differences. It also checks the PMA error counters and link speeds for the fabric, similar to opafabricanalysis -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 opafastfabric.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 two options:

- `-s`

  Saves 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.

  Beware, 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 `opafastfabric.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`.

3.2.3 opafabricanalysis

(All) Performs analysis of the fabric.

Syntax

```
opafabricanalysis [-b|-e] [-s] [-d dir] [-c file] [-t portsfile]
[-p ports] [-T topology_input]
```

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 = `/var/usr/lib/opa/analysis`
`-c file`	Specifies the error thresholds config file. Default = `/etc/sysconfig/opa/opamon.conf`
`-t portsfile`	Specifies the file with list of local HFI ports used to access fabric(s) for analysis. Default = `/etc/sysconfig/opa/ports`
`-p ports`	Specifies the list of local HFI ports used to access fabrics for analysis. Default is first active port. The first HFI in the system is 1. The first port on an HFI is 1. Uses the format hfi:port, for example: `0:0` First active port in system. `0:y` Port *y* within system. `x:0` First active port on HFI *x*. `x:y` HFI *x*, port *y*.
`-T` *topology_input*	Specifies the name of topology input file to use. Any `%P` markers in this filename are replaced with the `HFI:port` being operated on (such as `0:0` or `1:2`). Default = `/etc/sysconfig/opa/topology.%P.xml`. If `-T NONE` is specified, no topology input file is used. See Details and opareport on page 63 for more information.

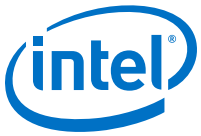
Example

```
opafabricanalysis
opafabricanalysis -p '1:1 1:2 2:1 2:2'
```

The fabric analysis tool checks the following:

- Fabric links (both internal to switch chassis and external cables)
- Fabric components (nodes, links, SMs, systems, and their SMA configuration)
- Fabric PMA 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 `opafabricanalysis`.



Environment Variables

The following environment variables are also used by this command:

`PORTS`	List of ports, used in absence of `-t` and `-p`.
`PORTS_FILE`	File containing list of ports, used in absence of `-t` and `-p`.
`FF_TOPOLOGY_FILE`	File containing `topology_input` (may have `%P` marker in filename), used in absence of `-T`.
`FF_ANALYSIS_DIR`	Top-level directory for baselines and failed health checks.

Details

For simple fabrics, the Intel® Omni-Path Fabric Suite FastFabric Toolset host is connected to a single fabric. By default, the first active port on the FastFabric Toolset host is used to analyze the fabric. However, in more complex fabrics, the FastFabric Toolset host may be connected to more than one fabric or subnet. In this case, you can specify the ports or HFIs to use with one of the following methods:

- On the command line using the `-p` option.
- In a file specified using the `-t` option.
- Through the environment variables `PORTS` or `PORTS_FILE`.
- Using the `PORTS_FILE` configuration option in `opafastfabric.conf`.

If the specified port does not exist or is empty, the first active port on the local system is used. In more complex configurations, you must specify the exact ports to use for all fabrics to be analyzed. For more information, refer to Selection of Devices on page 17.

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

- On the command line using the `-T` option.
- In a file specified through the environment variable `FF_TOPOLOGY_FILE`.
- Using the `ff_topology_file` configuration option in `opafastfabric.conf`.

If the specified file does not exist, no `topology_input` file is used. Alternately the filename can be specified as `NONE` to prevent use of an input file.

For more information, refer to opareport on page 63.

By default, the error analysis includes PMA counters and slow links (that is, links running below enabled speeds). You can change this using the `FF_FABRIC_HEALTH` configuration parameter in `opafastfabric.conf`. This parameter specifies the `opareport` options and reports to be used for the health analysis. It also can specify the PMA counter clearing behavior (`-I` *`seconds`*, `-C`, or none at all). See the *Intel® Omni-Path Fabric Suite FastFabric User Guide* for more information.

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 PMA counter analysis default to `/etc/sysconfig/opa/opamon.conf`. However, you can specify an alternate configuration file for thresholds using the `-c` option. The `opamon.si.conf` file can also be used to check for any non-zero values for signal integrity (SI) counters.

All files generated by `opafabricanalysis` start with `fabric` in their file name. This is followed by the port selection option identifying the port used for the analysis. Default is `0:0`.

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

Health Check

- `latest/fabric.0:0.errors`

  `stdout` of `opareport` for errors encountered during fabric error analysis.
- `latest/fabric.0.0.errors.stderr`

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

Baseline

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

- `baseline/fabric.0:0.snapshot.xml`

  `opareport` snapshot of complete fabric components and SMA configuration.
- `baseline/fabric.0:0.comps`

  `opareport` summary of fabric components and basic SMA configuration.
- `baseline/fabric.0.0.links`

  `opareport` summary of internal and external links.

Full Analysis

- `latest/fabric.0:0.snapshot.xml`

  `opareport` snapshot of complete fabric components and SMA configuration.
- `latest/fabric.0:0.snapshot.stderr`

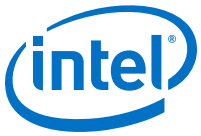
  `stderr` of `opareport` during snapshot.
- `latest/fabric.0:0.errors`

  `stdout` of `opareport` for errors encountered during fabric error analysis.
- `latest/fabric.0.0.errors.stderr`

  `stderr` of `opareport` during fabric error analysis.
- `latest/fabric.0:0.comps`

  `stdout` of `opareport` for fabric components and SMA configuration.
- `latest/fabric.0:0.comps.stderr`

  `stderr` of `opareport` for fabric components.
- `latest/fabric.0:0.comps.diff`



`diff` of baseline and latest fabric components.

- `latest/fabric.0:0.links`

  `stdout` of `opareport` summary of internal and external links.

- `latest/fabric.0:0.links.stderr`

  `stderr` of `opareport` summary of internal and external links.

- `latest/fabric.0:0.links.diff`

  `diff` of baseline and latest fabric internal and external links.

- `latest/fabric.0:0.links.changes.stderr`

  `stderr` of `opareport` comparison of links.

- `latest/fabric.0:0.links.changes`

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

- `latest/fabric.0:0.comps.changes.stderr`

  `stderr` of `opareport` comparison of components.

- `latest/fabric.0:0.comps.changes`

  `opareport` 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 `opareport -o links`:

- Unconnected/down/missing cables
- Added/moved cables
- Changes in link width and speed
- Changes to Node GUIDs in fabric (replacement of HFI or Switch hardware)
- Adding/Removing Nodes [FI, Virtual FIs, Virtual Switches, Physical Switches, Physical Switch internal switching cards (leaf/spine)]
- Changes to server or switch names

Based on `opareport -o comps`:

- Overlap with items from links report
- Changes in port MTU, LMC, number of VLs
- Changes in port speed/width enabled or supported
- Changes in HFI or switch device IDs/revisions/VendorID (for example, ASIC HW changes)
- Changes in port Capability mask (which features/agents run on port/server)
- Changes to ErrorLimits and PKey enforcement per port

- Changes to IOUs/IOCs/IOC Services provided

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

Location (port, node) and number of SMs in fabric. Includes:

- Primary and backups
- Configured priority for SM

Fabric Items Also Checked During Health Check

Based on `opareport -s -C -o errors -o slowlinks`:

- PMA error counters on all Intel® Omni-Path Fabric ports (HFI, switch external and switch internal) checked against configurable thresholds.
  - Counters are cleared each time a health check is run. Each health check reflects a counter delta since last health check.
  - 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 SA health.

3.2.4 opachassisanalysis

(Switch) Performs analysis of the chassis.

The `opachassisanalysis` tool checks the following for the Intel® Omni-Path Fabric Chassis:

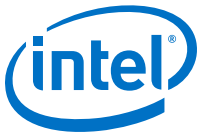
- Chassis configuration (as reported by the chassis commands specified in `FF_CHASSIS_CMDS` in `opafastfabric.conf`).
- Chassis health (as reported by the chassis command specified in `FF_CHASSIS_HEALTH` in `opafastfabric.conf`).

Syntax

```
opachassisanalysis [-b|-e] [-s] [-d dir] [-F chassisfile]
[-H 'chassis']
```

Options

`--help`	Produces full help text.
`-b`	Specifies the baseline mode. Default is the compare/check mode.
`-e`	Evaluates health only. Default is the 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 = `/var/usr/lib/opa/analysis`
`-F chassisfile`	Specifies the file with the chassis in the cluster. Default = `/etc/sysconfig/opa/chassis`
`-H 'chassis'`	Specifies the list of chassis on which to execute the command.

Example

```
opachassisanalysis
```

Environment Variables

The following environment variables are also used by this command:

`CHASSIS`	List of chassis, used if `-F` and `-H` options are not supplied.
`CHASSIS_FILE`	File containing list of chassis, used if `-F` and `-H` options are not supplied.
`FF_ANALYSIS_DIR`	Top-level directory for baselines and failed health checks.
`FF_CHASSIS_CMDS`	List of commands to issue during analysis, unused if `-e` option supplied.
`FF_CHASSIS_HEALTH`	Single command to issue to check overall health during analysis, unused if `-b` option supplied.

Details

Intel recommends that you set up SSH keys for chassis (see opasetupssh on page 136). If SSH keys are not set up, all chassis must be configured with the same admin password and the password must be kept in the `/etc/sysconfig/opa/opafastfabric.conf` configuration file.

The default set of `FF_CHASSIS_CMDS` is:

```
showInventory fwVersion showNodeDesc timeZoneConf timeDSTConf
snmpCommunityConf snmpTargetAddr showChassisIpAddr showDefaultRoute
```

The commands specified in `FF_CHASSIS_CMDS` must be simple commands with no arguments. The output of these commands are compared to the baseline using `FF_DIFF_CMD`. Therefore, commands that include dynamically changing values, such as port packet counters, should not be included in this list.

`FF_CHASSIS_HEALTH` can specify one command (with arguments) to be used to check the chassis health. For chassis with newer firmware, the `hwCheck` command is recommended. For chassis with older firmware, a benign command, such as `fruInfo`,

should be used. The default is hwCheck. Note that only the exit status of the FF_CHASSIS_HEALTH command is checked. The output is not captured and compared in a snapshot. However, on failure its output is saved to aid diagnosis.

The opachassisanalysis tool performs its analysis against one or more chassis in the fabric. As such, it permits the chassis to be specified using the -H, -F, CHASSIS, chassis_file or opafastfabric.conf. The handling of these options and settings is comparable to opacmdall -C and similar FastFabric Toolset commands against a chassis.

All files generated by opafabricanalysis start with chassis. in the file name.

The opachassisanalysis tool generates files such as the following within FF_ANALYSIS_DIR. The actual file names reflect the individual chassis commands that have been configured through the FF_CHASSIS_HEALTH and FF_CHASSIS_CMDS parameters:

Health Check

- latest/chassis.hwCheck

  Output of hwCheck command for all selected chassis

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

- baseline/chassis.fwVersion

  Output of fwVersion command for all selected chassis.
- baseline/chassis.showChassisIpAddr

  Output of the showChassisIpAddr. command for all selected chassis.
- baseline/chassis.showDefaultRoute

  Output of the showDefaultRoute command for all selected chassis.
- baseline/chassis.showNodeDesc

  Output of the showNodeDesc command for all selected chassis.
- baseline/chassis.showInventory

  Output of the showInventory command for all selected chassis.
- baseline/chassis.snmpCommunityConf

  Output of the snmpCommunityConf command for all selected chassis.
- baseline/chassis.snmpTargetAddr

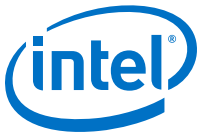
  Output of the snmpTargetAddr command for all selected chassis.
- baseline/chassis.timeDSTConf

  Output of the timeDSTConf command for all selected chassis.
- baseline/chassis.timeZoneConf

  Output of the timeZoneConf command for all selected chassis.

Full Analysis: The following .diff files are only created if differences are detected.

- latest/chassis.hwCheck



Output of the `hwCheck` command for all selected chassis.

- `latest/chassis.fwVersion`

  Output of the `fwVersion` command for all selected chassis.
- `latest/chassis.fwVersion.diff`

  `diff` of the baseline and latest `fwVersion`.
- `latest/chassis.showChassisIpAddr`

  Output of the `showChassisIpAddr` command for all selected chassis.
- `latest/chassis.showChassisIpAddr.diff`

  `diff` of baseline and latest `showChassisIpAddr`.
- `latest/chassis.showDefaultRoute`

  Output of the `showDefaultRoute` command for all selected chassis.
- `latest/chassis.showDefaultRoute.diff`

  `diff` of the baseline and the latest `showDefaultRoute`.
- `latest/chassis.showNodeDesc`

  Output of the `showNodeDesc` command for all selected chassis.
- `latest/chassis.showNodeDesc.diff`

  `diff` of the baseline and latest `showNodeDesc`.
- `latest/chassis.showInventory`

  Output of the `showInventory` command for all selected chassis.
- `latest/chassis.showInventory.diff`

  `diff` of the baseline and latest `showInventory`.
- `latest/chassis.snmpCommunityConf`

  Output of the `snmpCommunityConf` command for all selected chassis.
- `latest/chassis.snmpCommunityConf.diff`

  `diff` of the baseline and latest `snmpCommunityConf`.
- `latest/chassis.snmpTargetAddr`

  Output of the `snmpTargetAddr` command for all selected chassis.
- `latest/chassis.snmpTargetAddr.diff`

  `diff` of the baseline and latest `snmpTargetAddr`.
- `latest/chassis.timeDSTConf`

  Output of the `timeDSTConf` command for all selected chassis.
- `latest/chassis.timeDSTConf.diff`

  `diff` of the baseline and latest `timeDSTConf`.
- `latest/chassis.timeZoneConf`

  Output of the `timeZoneConf` command for all selected chassis.
- `latest/chassis.timeZoneConf.diff`

diff of the baseline and latest timeZonfConf.

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

Chassis Items Checked Against the Baseline

Based upon showInventory:

- Addition/removal of Chassis FRUs

  Replacement is only checked for FRUs that showInventory displays the serial number.
- Removal of redundant FRUs (spines, power supply, fan)

Based upon fwVersion:

- Changes to primary or alternate FW versions installed in cards in chassis.

Based upon showNodeDesc:

- Changes to configured node description for chassis. Note changes detected here would also be detected in fabric level analysis.

Based upon timeZoneConf and timeDSTConf:

- Changes to the chassis time zone and daylight savings time configuration.

Based upon snmpCommunityConf and snmpTargetAddr:

- Changes to SNMP persistent configuration within the chassis.

The following Chassis items are not checked against baseline:

- Changes to the chassis configuration on the management LAN (for example, showChassisIpAddr, showDefaultRoute). Such changes typically result in the chassis not responding on the LAN at the expected address that is detected by failures that perform other chassis checks.

Chassis Items Also Checked During Health Check

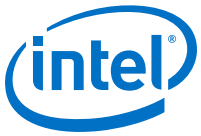
Based upon hwCheck:

- Overall health of FRUs in chassis:
  - Status of Fans in chassis
  - Status of Power Supplies in chassis
  - Temp/Voltage for each card
- Presence of adequate power/cooling of FRUs
- Presence of N+1 power/cooling of FRUs
- Presence of Redundant AC input

3.2.5 opahostsmanalysis

(All) Performs analysis against the local server only. It is assumed that both the host SM and the FastFabric are installed on the same system.

The host SM analysis tool checks the following:



- Host SM software version
- Host SM configuration file (simple text compare using `FF_DIFF_CMD`)
- Host SM health (for example, is it running?)

Syntax

```
opahostsmanalysis [-b|-e] [-s] [-d dir]
```

Options

`--help` Produces full help text.

`-b` Specifies the baseline mode. Default is the compare/check mode.

`-e` Evaluates health only. Default is the 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 = `/var/usr/lib/opa/analysis`

Example

```
opahostsmanalysis
```

Environment Variables

The following environment variables are also used by this command:

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

`FF_CURTIME` Timestamp to use on the directory created in `FF_DIFF_CMD`.

`FF_DIFF_CMD` Linux* command to use to compare baseline to latest snapshot.

Details

All files generated by `opahostsmanalysis` start with `hostsm` in the file name.

The `opahostsmanalysis` tool generates files such as the following within `FF_ANALYSIS_DIR`. The actual file names reflect the individual chassis commands that have been configured using the `FF_CHASSIS_HEALTH` and `FF_CHASSIS_CMDS` parameters:

Health Check

- `latest/hostsm.smstatus` – Output of the `sm_query smShowStatus` command.

Baseline

- `baseline/hostsm.smver` – Host SM version.

- baseline/hostsm.smconfig – Copy of opafm.xml.

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

Full Analysis

- latest/hostsm.smstatus – Output of the sm_query smShowStatus command.
- latest/hostsm.smver – Host SM version.

  latest/hostsm.smver.diff – diff of the baseline and latest host SM version.
- latest/hostsm.smconfig – Copy of opafm.xml.
- latest/hostsm.smconfig.diff – diff of the baseline and the latest opafm.xml.

The .diff 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 a time-stamped directory name under FF_ANALYSIS_DIR.

Host SM Items Checked Against the Baseline

- SM configuration file.
- Version of the SM rpm installed on the system.

Host SM Items Also Checked During Health Check

- The SM is in the running state.

3.2.6 opaesmanalysis

(Switch) Performs analysis of the embedded Subnet Manager (SM) for configuration and health. The opaesmanalysis tool checks the opafm.xml file for the chassis.

All files generated by opaesmanalysis start with esm in the file name.

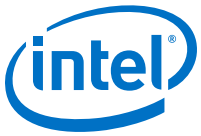
Intel recommends that you set up SSH keys for chassis (see opasetupssh on page 136). If SSH keys are not set up, all chassis must be configured with the same admin password and the password must be kept in the opafastfabric.conf configuration file.

Syntax

```
opaesmanalysis [-b|-e] [-s] [-d dir] [-G esmchassisfile]
[-E 'esmchassis']
```

Options

--help	Produces full help text.
-b	Specifies the baseline mode. Default is the compare/check mode.
-e	Evaluates health only. Default is the 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 = `/var/usr/lib/opa/analysis`
`-G` *esmchassisfile*	Specifies the file with SM chassis in the cluster. Default = `/etc/sysconfig/opa/esm_chassis`
`-E '`*esmchassis*`'`	Specifies the list of SM chassis on which to execute the command.

Example

```
opaesmanalysis
```

Environment Variables

The following environment variables are also used by this command:

`ESM_CHASSIS`	List of SM chassis, used if `-G` and `-E` options are not supplied.
`ESM_CHASSIS_FILE`	File containing list of SM chassis, used if `-G` and `-E` options are not supplied.
`FF_ANALYSIS_DIR`	Top-level directory for baselines and failed health checks.

3.2.7 opaallanalysis

(All) `opaallanalysis` command performs the set of analysis specified in `FF_ALL_ANALYSIS` and can be specified for fabric, chassis, esm, or hostsm.

Syntax

```
opaallanalysis [-b|-e] [-s] [-d dir] [-c file]
[-t portsfile] [-p ports]
[-F chassisfile] [-H 'chassis']
[-G esmchassisfile] [-E esmchassis]
[-T topology_input]
```

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 history of failures (errors/differences).

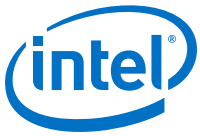
-d *dir*	Identifies the top-level directory for saving baseline and history of failed checks. Default = /var/usr/lib/opa/analysis
-c *file*	Specifies the error thresholds configuration file. Default = /etc/sysconfig/opa/opamon.conf
-t *portsfile*	Specifies the file with list of local HFI ports used to access fabric(s) for analysis. Default = /etc/sysconfig/opa/ports
-p *ports*	Specifies the list of local HFI ports used to access fabric(s) for analysis. Default is the first active port. Specified as HFI:port as follows: 0:0 First active port in system. 0:*y* Port *y* within system. *x*:0 First active port on HFI *x*. *x*:*y* HFI *x*, port *y*.
-F *chassisfile*	Specifies the file with a chassis in a cluster. Default = /etc/sysconfig/opa/chassis
-H '*chassis*'	Specifies the list of chassis on which to execute the command.
-G *esmchassisfile*	Specifies the file with embedded SM chassis in the cluster. Default = /etc/sysconfig/opa/esm_chassis
-E *esmchassis*	Specifies the list of embedded SM chassis to analyze.
-T *topology_input*	Specifies the name of topology input file to use. Any %P markers in this filename are replaced with the HFI:port being operated on, such as 0:0 or 1:2. Default = /etc/sysconfig/opa/topology.%P.xml. If -T NONE is specified, no topology input file is used. See opareport on page 63 for more information.

Example

```
opaallanalysis
opaallanalysis -p '1:1 2:1'
```

Environment Variables

The following environment variables are also used by this command:



PORTS	List of ports, used in absence of -t and -p.
PORTS_FILE	File containing list of ports, used in absence of -t and -p.
FF_TOPOLOGY_FILE	File containing topology_input (may have %P marker in filename), used in absence of -T.
CHASSIS	List of chassis, used if -F and -H options are not supplied.
CHASSIS_FILE	File containing list of chassis, used if -F and -H options are not supplied.
ESM_CHASSIS	List of SM chassis, used if -G and -E options are not supplied.
ESM_CHASSIS_FILE	File containing list of SM chassis, used if -G and -E options are not supplied.
FF_ANALYSIS_DIR	Top level directory for baselines and failed health checks.

Details

The opaallanalysis 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/sysconfig/opa/opafastfabric.conf. The analysis set includes the options: fabric, chassis, esm, or hostsm.

Note that the opaallanalysis command has options which are a super-set 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 opafabricanalysis 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.

3.2.8 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 opaallanalysis 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.

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

- Each run adds a potential burden to the SM, fabric, and switches. For infrequent runs (hourly or daily), this impact is negligible. However, if this were to be run very frequently, the impacts to fabric and SM 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 down time 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.

3.2.9 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:

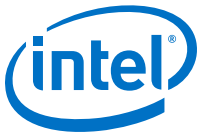
- Repair a faulty board, which leads to a new serial number for that component.
- Update switch firmware or Fabric Manager.
- Change time zones in a switch.
- Add or delete a new device or link to a fabric.
- Remove a failed link and its devices from the Fabric Manager database.

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 `opafabricinfo` which returns information for each subnet to which the fabric management server is connected. The following is an example output for a single subnet.

```
# opafabricinfo
Fabric 0:0 Information:
SM: hds1fnb6241 hfi1_0 Guid: 0x0011750101575ffe State: Master
Number of HFIs: 8
Number of Switches: 1
Number of Links: 8
Number of HFI Links: 8              (Internal: 0   External: 8)
Number of ISLs: 0                   (Internal: 0   External: 0)
Number of Degraded Links: 0         (HFI Links: 0   ISLs: 0)
Number of Omitted Links: 0          (HFI 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/opa/analysis/baseline`.
4. Run `opaallanalysis -b`



5. Check the new output files in `/var/usr/lib/analysis/baseline` to verify that the configuration is as you expect it. Refer to the *Intel® Omni-Path Fabric Suite FastFabric User Guide* for details.

3.2.10 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.

Intel recommends that you review the results for any ESM or SM health check failures. If the SM is misconfigured or not running, it can cause other health checks to fail. In this case, correct the SM problems first, then rerun the health check.

For a host SM analysis, review the files in the following order:

1. `latest/hostsm.smstatus`

   Make sure this file indicates the SM is running. If no SMs are running on the fabric, correct that problem before proceeding further. Once corrected, rerun the health checks to look for further errors.

2. `latest/hostsm.smver.diff`

   Indicates the SM version has changed. If this was not an expected change, correct the SM before proceeding further. 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.

3. `latest/hostsm.smconfig.diff`

   Indicates that the SM configuration has changed. Review this file and compare the `latest/hostsm.smconfig` file to `baseline/hostsm.smconfig`. Correct the SM configuration, 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.

For an ESM analysis, the `FF_ESM_CMDS` configuration setting selects which ESM commands are used for the analysis. When using the default setting for this parameter, review the files in the following order:

1. `latest/esm.smstatus`

   Make sure this indicates the SM is running. If no SMs are running on the fabric, correct the problem before proceeding further. Once corrected, rerun the health checks to look for further errors.

2. `latest/esm.CHASSIS.opafm.xml`

   The `opafm.xml` file for the given chassis.

3. `latest/esm.CHASSIS.opafm.xml.diff`

Indicates that the SM configuration has changed. Review this file and compare the `latest/esm.CHASSIS.opafm.xml` file to `baseline/esm.CHASSIS.opafm.xml`. Correct the SM configuration, 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.

4. `latest/esm.smShowSMParms.diff`

   Indicates that the SM configuration has changed. Review this file and compare the `latest/esm.smShowSMParms` file to `baseline/esm.smShowSMParms`. Correct the SM configuration, 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.

5. `latest/esm.smShowDefBcGroup.diff`

   Indicates that the SM broadcast group for IPoIB configuration has changed. Review this file and compare the `latest/esm.smShowDefBcGroup` file to `baseline/esm.smShowDefBcGroup`. Correct the SM configuration, 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.

6. `latest/esm.*.diff`

   If `FF_ESM_CMDS` has been modified, review the changes in results for those additional commands. Correct the SM configuration, 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.

Next, 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.0.0`. 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.0.0.errors.stderr`

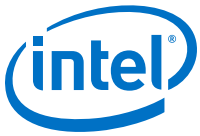
   If this file is not empty, it can indicate problems with `opareport`, such as inability to access an SM. This may result in unexpected problems or inaccuracies in the related errors file. Correct problems reported in this file first. Once corrected, rerun the health checks to look for further errors.

2. `latest/fabric.0:0.errors`

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

3. `latest/fabric.0.0.snapshot.stderr`

   If this file is not empty, it can indicate problems with `opareport`, such as inability to access an SM. 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.0:0.links.stderr` and `latest/fabric.0:0.links.changes.stderr`

   If these files are not empty, it can indicate problems with `opareport` 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. For more information on `.changes` files, refer to Interpreting Health Check .changes Files on page 48.

5. `latest/fabric.0:0.links.diff` and `latest/fabric.0:0.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.0:0.links.changes` file since it has a more concise and precise description of the fabric link changes. Compare the `latest/fabric.0:0.links` file to `baseline/fabric.0:0.links`. If components have disappeared, review the `latest/fabric.0:0.comps.diff` and `latest/fabric.0:0.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. For more information on `.changes` files, refer to Interpreting Health Check .changes Files on page 48.

6. `latest/fabric.0:0.comps.stderr` and `latest/fabric.0:0.comps.changes.stderr`

   If these files are not empty, it can indicate problems with `opareport` 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. For more information on `.changes` files, refer to Interpreting Health Check .changes Files on page 48.

7. `latest/fabric.0:0.comps.diff` and `latest/fabric.0:0.comps.changes`

   These indicate that the components in the fabric or their SMA configuration have changed. If both files are available, use the `fabric.0:0.comps.changes` file since it has a more concise and precise description of the fabric component changes. Compare the `latest/fabric.0:0.comps` file to `baseline/fabric.0:0.comps`. Correct missing nodes, missing SMs, 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. For more information on `.changes` files, refer to Interpreting Health Check .changes Files on page 48.

Review the results of the `opachassisanalysis`. If chassis configuration has changed, the `opachassisanalysis` report detects it. Previous checks should have already detected missing chassis, missing or added links and many aspects of chassis configuration. For `opachassisanalysis`, the `FF_CHASSIS_CMDS` and `FF_CHASSIS_HEALTH` configuration settings select which chassis commands are used for the analysis. When using the default setting for this parameter, review the files in the following order:

1. `latest/chassis.hwCheck`

Make sure this indicates all chassis are operating properly with the desired power and cooling redundancy. If there are problems, correct them, but other analysis files can be analyzed first. Once any problems are corrected, rerun the health checks to verify the correction.

2. `latest/chassis.fwVersion.diff`

   Indicates the chassis firmware version has changed. If this was not an expected change, correct the chassis firmware before proceeding further. 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.

3. `latest/chassis.*.diff`

   These files reflect other changes to chassis configuration based on checks selected by `FF_CHASSIS_CMDS`. Review the changes in results for these remaining commands. Correct the chassis, 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.

3.2.11 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.0:0.links.changes
Links Topology Verification

Links Found with incorrect configuration:
Rate NodeGUID          Port Type Name
100g 0x0011750101603593   1 FI   phs1fnivd13u07n3 hfi1_0
<->  0x00117501026a5619  11 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.

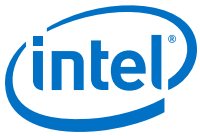


Table 2. Possible issues found in health check .changes files

Issue	Description and possible actions
Missing	This indicates an item that is in the baseline, 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've removed an HFI connection, the HFI 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 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've added an HFI 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 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.
Duplicate	This 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 addition of parallel links. It can also be reported when there enough changes to the fabric that it is difficult to properly resolve and report all the changes. It can also occur when `opareport` is run with manually generated topology input files that may have duplicate items or incomplete specifications.
Different	This 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 indicates that the attributes of a port on one side of a link have changed, such as PortGuid, 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.

continued...

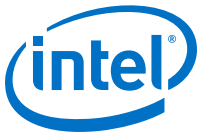
Issue	Description and possible actions
	If the configuration has changed purposely since the most recent baseline, and this difference is reflected here, save the original baseline and rerun the baseline. If a faulty device was replaced, it is important to re-establish the baseline. If this difference 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 indicates that the attributes of a node in the fabric have changed, such as NodeGuid, 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".
SM Attributes Inconsistent	This indicates that the attributes of the node or port running an SM in the fabric have changed, such as NodeGuid, Node Description, Port Number, Device Type, or others. The inconsistency is caused by moving a cable, changing from host-based subnet management to embedded subnet management (or vice-versa), or by replacing the HFI in the fabric management server. 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 the HFI in the fabric management server 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 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 GUIDs 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. This is a specific case of "Different".
Incorrect Link	This 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".

3.3 Verification, Analysis, and Control CLIs

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

3.3.1 opacabletest

(Switch) Initiates or stops Cable Bit Error Rate stress tests for Intel® Omni-Path Host Fabric Interface (HFI)-to-switch links and/or ISLs.



Syntax

```
opacabletest [-C|-A] [-c file] [-f hostfile]
[-h 'hosts'] [-n numprocs] [-t portsfile]
[-p ports] [start|start_fi|start_isl|stop|stop_fi| stop_isl] ...
```

Options

`--help`	Produces full help text.
`-C`	Clears error counters.
`-A`	Forces the system to clear hardware error counters. Implies `-C`.
`-c` *file*	Specifies the error thresholds configuration file. Default is `/etc/sysconfig/opa/opamon.si.conf` file. Only used if `-C` or `-A` specified.
`-f` *hostfile*	Specifies the file with hosts to include in HFI-to-SW test. Default is `/etc/sysconfig/opa/hosts` file.
`-h` *hosts*	Specifies the list of hosts to include in HFI-SW test.
`-n` *numprocs*	Specifies the number of processes per host for HFI-SW test.
`-t` *portsfile*	Specifies the file with list of local HFI ports used to access fabrics when clearing counters. Default is `/etc/sysconfig/opa/ports` file.
`-p` *ports*	Specifies the list of local HFI ports used to access fabrics for counter clear. Default is first active port. The first HFI in the system is 1. The first port on an HFI is 1. Uses the format hfi:port, for example: `0:0` First active port in system. `0:y` Port *y* within system. `x:0` First active port on HFI *x*. `x:y` HFI *x*, port *y*.
`start`	Starts the HFI-SW and ISL tests.
`start_fi`	Starts the HFI-SW test.
`start_isl`	Starts the ISL test.
`stop`	Stops the HFI-SW and ISL tests.

stop_fi	Stops the HFI-SW test.
stop_isl	Stops the ISL test.

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

The ISL cable test started by this tool assumes that the master Host Subnet Manager (HSM) is running on this host. If using the Embedded Subnet Manager (ESM), or if a different host is the master HSM, the ISL cable test must be controlled by the switch CLI, or by Intel® Omni-Path Fabric Suite FastFabric on the master HSM respectively.

Examples

```
opacabletest -A start
opacabletest -f good -A start
opacabletest -h 'arwen elrond' start_fi
HOSTS='arwen elrond' opacabletest stop
opacabletest -A
```

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.
PORTS	List of ports, used in absence of -t and -p.
PORTS_FILE	File containing list of ports, used in absence of -t and -p.
FF_MAX_PARALLEL	Maximum concurrent operations.

3.3.2 opaextractbadlinks

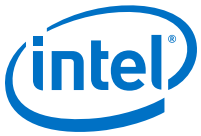
Produces a CSV file listing all or some of the links that exceed opareport -o error thresholds. opaextractbadlinks is a front end to the opareport tool. The output from this tool can be imported into a spreadsheet or parsed by other scripts.

Syntax

```
opaextractbadlinks [opareport options]
```

Options

opareport *options*	Options are passed to opareport. See opareport on page 63 for the full set of options.



Examples

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

# List all the bad links to a switch named "OmniPth00117501ffffffff":
opaextractbadlinks -F "node:OmniPth00117501ffffffff"

# List all the bad links to end-nodes:
opaextractbadlinks -F "nodetype:FI"

# List all the bad links on the 2nd HFI's fabric of a multi-plane fabric:
opaextractbadlinks -h 2
```

3.3.3 opaextractlink

Produces a CSV file listing all or some of the links in the fabric. `opaextractlink` is a front end to the `opareport` tool. The output from this tool can be imported into a spreadsheet or parsed by other scripts.

Syntax

```
opaextractlink [opareport options]
```

Options

`opareport` *`options`*	Options are passed to `opareport`. See opareport on page 63 for the full set of options.

Examples

```
# List all the links in the fabric:
opaextractlink

# List all the links to a switch named "OmniPth00117501ffffffff":
opaextractlink -F "node:OmniPth00117501ffffffff"

# List all the links to end-nodes:
opaextractlink -F "nodetype:FI"

# List all the links on the 2nd HFI's fabric of a multi-plane fabric:
opaextractlink -h 2
```

3.3.4 opaextractmissinglinks

Produces a CSV file listing all or some of the links in the fabric. `opaextractmissinglinks` is a front end to the `opareport` 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

```
opaextractmissinglinks [-T topology_input] [-o report]
[opareport options]
```

Options

`-T topology_file`	Specifies the file used for fabric verification: `/etc/sysconfig/opa/topology.0:0.xml`.
`-o report`	The output types specific for `opaextractmissinglinks` include: • `verifylinks` – Verifies links against topology input. • `verifyextlinks` – Verifies links against topology input. Limits analysis to links external to systems. • `verifyfilinks` – Verifies links against topology input. Limits analysis to FI 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.
`opareport options`	Options are passed to `opareport`. See opareport on page 63 for the full set of options.

Examples

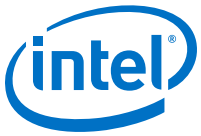
```
List all the missing links in the fabric:
      opaextractmissinglinks

List all the missing links to a switch named "OmniPth00117501ffffffff":
      opaextractmissinglinks -T topology.0:0.xml -F "node:OmniPth00117501ffffffff"

List all the missing connections to end-nodes:
      opaextractmissinglinks -o verifyfilinks

List all the missing links on the 2nd HFI's fabric of a multi-plane fabric:
      opaextractmissinglinks -h 2 -T /etc/sysconfig/opa/topology.2:1.xml

List all the missing links between two switches:
      opaextractmissinglinks -o verifyislinks -T topology.0:0.xml
```

3.3.5 opaextractsellinks

Produces a CSV file listing all or some of the links in the fabric. `opaextractsellinks` is a front end to the `opareport` tool. The output from this tool can be imported into a spreadsheet or parsed by other scripts.

Syntax

```
opaextractsellinks [opareport options]
```



Options

`opareport` *`options`*	Options are passed to `opareport`. See opareport on page 63 for the full set of options.

Examples

```
# List all the links in the fabric:
opaextractsellinks

# List all the links to a switch named "OmniPth00117501ffffffff":
opaextractsellinks -F "node:OmniPth00117501ffffffff"

# List all the connections to end-nodes:
opaextractsellinks -F "nodetype:FI"

# List all the links on the 2nd HFI's fabric of a multi-plane fabric:
opaextractsellinks -h 2
```

3.3.6 opaextractstat2

Performs an 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. `opaextractstat2` is a front end to the `opareport` and `opaxmlextract` tools.

Syntax

```
opaextractstat2 topology_file [opareport options]
```

Options

`topology_file`	Specifies topology_file to use.
`opareport` *`options`*	Options are passed to `opareport`. See opareport on page 63 for the full set of options.

The portion of the script that calls `opareport` and `opaxmlextract` follows:

```
opareport -x -d 10 -s -o errors -T $@ | opaxmlextract -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.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

```
opaextractstat2 topology_file

opaextractstat2 topology_file -c my_opamon.conf
```

3.3.7 opafabricinfo

Provides a brief summary of the components in the fabric, using the first active port on the given local host to perform its analysis. `opafabricinfo` is supplied in both:

- Intel® Omni-Path Fabric Suite FastFabric Toolset

  In this situation, the command can manage more than one fabric (subnet).

- FastFabric Tools

  In this situation, the command performs analysis against the first active port on the system only. It takes no options and uses no environment variables.

`opafabricinfo` can be very 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 opareport on page 63 and opareports on page 73. Also see `opatop` in the *Intel® Omni-Path Fabric Suite FastFabric User Guide*.

Syntax

```
opafabricinfo [-t portsfile] [-p ports]
```

Options

`--help` Produces full help text.

`-t` *portsfile* Specifies the file with list of local HFI ports used to access fabric(s) for analysis. Default is `/etc/sysconfig/opa/ports` file.

`-p` *ports* Specifies the list of local HFI ports used to access fabrics for analysis.

Default is first active port. The first HFI in the system is 1. The first port on an HFI is 1. Uses the format hfi:port, for example:

`0:0` First active port in system.

`0:y` Port *y* within system.

`x:0` First active port on HFI *x*.

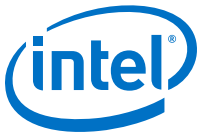
`x:y` HFI *x*, port *y*.

Environment Variables

The following environment variables are also used by this command:

`PORTS` List of ports, used in absence of `-t` and `-p`.

`PORTS_FILE` File containing list of ports, used in absence of `-t` and `-p`.



For simple fabrics, the Intel® Omni-Path Fabric Suite FastFabric Toolset host is connected to a single fabric. By default, the first active port on the FastFabric Toolset host is used to analyze the fabric. However, in more complex fabrics, the FastFabric Toolset host may be connected to more than one fabric or subnet. In this case, you can specify the ports or HFIs to use with one of the following methods:

- On the command line using the `-p` option.
- In a file specified using the `-t` option.
- Through the environment variables `PORTS` or `PORTS_FILE`.
- Using the `ports_file` configuration option in `opafastfabric.conf`.

If the specified port does not exist or is empty, the first active port on the local system is used. In more complex configurations, you must specify the exact ports to use for all fabrics to be analyzed.

For more information, refer to Selection of Devices on page 17.

Example

```
opafabricinfo
opafabricinfo -p '1:1 1:2 2:1 2:2'
```

Output example

```
# opafabricinfo
Fabric 0:0 Information:
SM: hds1fnb6241 hfi1_0 Guid: 0x0011750101575ffe State: Master
Number of HFIs: 8
Number of Switches: 1
Number of Links: 8
Number of HFI Links: 8              (Internal: 0   External: 8)
Number of ISLs: 0                   (Internal: 0   External: 0)
Number of Degraded Links: 0         (HFI Links: 0   ISLs: 0)
Number of Omitted Links: 0          (HFI Links: 0   ISLs: 0)
```

Output Definitions

`SM`	Each subnet manger (SM) running in the fabric is listed along with its node name, port GUID, and present SM state (Master, Standby, etc.).
`Number of HFIs`	Number of unique host fabric interfaces (HFIs) in the fabric. An FI with two connected ports is counted as a single FI. *Note:* Fabric Interfaces include HFIs in servers as well as HFIs within I/O Modules, Native Storage, etc.
`Number of Switches`	Number of connected switches in the fabric.

`Number of Links`	Number of links in the fabric. Note that a large switch may have internal links.
`Number of HFI Links`	Number of HFI links (Internal and External) in the fabric.
`Number of ISLs`	Number of Interswitch Links (Internal and External) in the fabric.
`Number of Degraded Links`	Number of degraded links (HSI and ISL) in the fabric.
`Number of Omitted Links`	Number of omitted links (HSI and ISL) in the fabric.

3.3.8 opafindgood

Checks for hosts that are able to be pinged, accessed via SSH, and active on the Intel® Omni-Path 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.

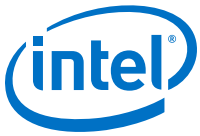
The resulting `good` file lists each good host exactly once and can be used as input to create *`mpi_hosts`* files for running `mpi_apps` and the HFI-SW cable test. The files `alive`, `running`, `active`, `good`, and `bad` are created in the selected directory listing hosts passing each criteria.

This command assumes the Node Description for each host is based on the *`hostname`* `-s` output in conjunction with an optional `hfi1_#` suffix. When using a `/etc/sysconfig/opa/hosts` file that lists the hostnames, this assumption may not be correct.

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:

```
# opafindgood
3 hosts will be checked
2 hosts are pingable (alive)
2 hosts are ssh'able (running)
2 total hosts have FIs active on one or more fabrics (active)
No Quarantine Node Records Returned
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
2015/10/04 11:33:22;phs1fnivd13u07n1 hfi1_0 p1 phs1swivd13u06 p16;Link errors
2015/10/07 10:21:05;phs1swivd13u06;Switch not found in SA DB
2015/10/09 14:36:48;phs1fnivd13u07n4;Doesn't ping
2015/10/09 14:36:48;phs1fnivd13u07n3;No active 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 port`. There may be cases where ports could be active for hosts that do not ping, especially if Ethernet host names are used for the ping test. 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.

Note that `opafindgood` queries the SA for NodeDescriptions to determine hosts with active ports. As such, ports may be active for hosts that cannot be accessed via SSH or pinged.

By default, `opafindgood` checks for and reports nodes that are quarantined for security reasons. To skip this, use the `-Q` option.

Syntax

```
opafindgood [-R|-A|-Q] [-d dir] [-f hostfile] [-h 'hosts']
[-t portsfile] [-p ports] [-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® Omni-Path Fabric software or fabric is not up.

`-Q` Skips the quarantine test. Recommended if Intel® Omni-Path Fabric software or fabric is not up.

`-d dir` Specifies the directory in which to create alive, active, running, good, and bad files. Default is `/etc/sysconfig/opa` directory.

`-f hostfile` Specifies the file with hosts in cluster. Default is `/etc/sysconfig/opa/hosts` directory.

`-h hosts` Specifies the list of hosts to ping.

`-t portsfile` Specifies the file with list of local HFI ports used to access fabric(s) for analysis. Default is `/etc/sysconfig/opa/ports` file.

`-p ports` Specifies the list of local HFI ports used to access fabric(s) for analysis.

Default is first active port. The first HFI in the system is 1. The first port on an HFI is 1. Uses the format hfi:port, for example:

`0:0` First active port in system.

`0:y` Port *y* within system.

`x:0` First active port on HFI *x*.

`x:y` HFI *x*, port *y*.

`-T` *timelimit* Specifies the time limit in seconds for host to respond to SSH. Default = 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`.
`PORTS`	List of ports, used in absence of `-t` and `-p`.
`PORTS_FILE`	File containing list of ports, used in absence of `-t` and `-p`.
`FF_MAX_PARALLEL`	Maximum concurrent operations.

Examples

```
opafindgood
opafindgood -f allhosts
opafindgood -h 'arwen elrond'
HOSTS='arwen elrond' opafindgood
HOSTS_FILE=allhosts opafindgood
opafindgood -p '1:1 1:2 2:1 2:2'
```

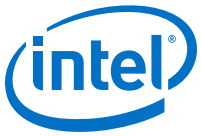
3.3.9 opalinkanalysis

(Switch) Encapsulates the capabilities for link analysis. Additionally, this tool includes cable and fabric topology verification capabilities. This tool is built on top of `opareport` (and its analysis capabilities), and accepts the same syntax for input topology and snapshot files.

In addition to being able to run assorted `opareport` 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

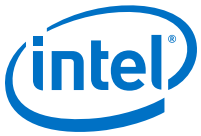
```
opalinkanalysis [-U] [-t portsfile] [-p ports]  [-T topology_input]
[-X snapshot_input] [-x snapshot_suffix] [-c file] reports ...
```



Options

`--help`	Produces full help text.
`-U`	Omits unexpected devices and links in `punchlist` file from verify reports.
`-t` *`portsfile`*	Specifies the file with list of local HFI ports used to access fabric(s) for analysis, default is `/etc/sysconfig/opa/ports`.
`-p` *`ports`*	Specifies the list of local HFI ports used to access fabrics for analysis. Default is first active port. The first HFI in the system is 1. The first port on an HFI is 1. Uses the format hfi:port, for example: `0:0` First active port in system. `0:y` Port *`y`* within system. `x:0` First active port on HFI *`x`*. `x:y` HFI *`x`*, port *`y`*.
`-T` *`topology_input`*	Specifies the name of a topology input file to use. Any `%P` markers in this filename are replaced with the `hfi:port` being operated on (such as 0:0 or 1:2). Default is `/etc/sysconfig/opa/topology.%P.xml`. If `NONE` is specified, does not use any `topology_input` files. See opareport on page 63 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 `opareport -o snapshot` run. If an errors report is specified, snapshot must have been generated with the `opareport -s` option. When this option is used, only one port may be specified to select a `topology_input` file (unless `-T` specified). When this option is used, `clearerrors` and `clearhwerrors` reports are not permitted.
`-x` *`snapshot_suffix`*	Creates a snapshot file per selected port. The files are created in `FF_RESULT_DIR` with names of the form: `snapshotSUFFIX.`*`HFI:PORT`*`.xml`.
`-c` *`file`*	Specifies the error thresholds configuration file. The default is `/etc/sysconfig/opa/opamon.si.conf`.
`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 all reports above. (`errors`, `slowlinks`, `misconfiglinks`, and `misconnlinks`)
`verifylinks`	Verifies links against topology input.
`verifyextlinks`	Verifies links against topology input. Limits analysis to links external to systems.
`verifyfilinks`	Verifies links against topology input. Limits analysis to FI 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.
`verifyfis`	Verifies FIs against topology input.
`verifysws`	Verifies switches against topology input.
`verifyrtrs`	Verifies routers against topology input.
`verifynodes`	Verifies FIs, switches, and routers against topology input.
`verifysms`	Verifies SMs against topology input.
`verifyall`	Verifies links, FIs, switches, routers, and SMs against topology input.
`clearerrors`	Clears error counters, uses PM if available.
`clearhwerrors`	Clears hardware error counters, bypasses PM.



`clear`	Includes `clearerrors` and `clearhwerrors`.

A punchlist of bad links is also appended to the file: `FF_RESULT_DIR/punchlist.csv`

Examples

```
opalinkanalysis errors
opalinkanalysis errors clearerrors
opalinkanalysis -p '1:1 1:2 2:1 2:2'
```

Environment Variables

The following environment variables are also used by this command:

`PORTS`	List of ports, used in absence of `-t` and `-p`.
`PORTS_FILE`	File containing list of ports, used in absence of `-t` and `-p`.
`FF_TOPOLOGY_FILE`	File containing *`topology_input`*, used in absence of `-T`.

3.3.10 opareport

(All) Provides powerful fabric analysis and reporting capabilities. Must be run on a host connected to the Intel® Omni-Path Fabric with the Intel® Omni-Path Fabric Suite FastFabric Toolset installed.

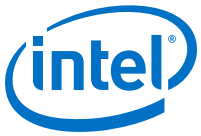
Syntax

```
opareport [-v][-q] [-h hfi] [-p port]
[-o report] [-d detail] [-P|-H] [-N] [-x]
[-X snapshot_input] [-T topology_input] [-s] [-r] [-V]
[-i seconds] [-b date_time] [-e date_time] [-C]
[-a] [-m] [-K mkey] [-M] [-A] [-c file] [-L]
[-F point] [-S point] [-D point] [-Q]
```

Options

`--help`	Produces full help text.
`-v/--verbose`	Returns verbose output.
`-q/--quiet`	Disables progress reports.
`-h/--hfi` *`hfi`*	Specifies the HFI, numbered 1..n. Using 0 specifies that the `-p` *`port`* port is a system-wide port number. (Default is 0.)
`-p/--port` *`port`*	Specifies the port, numbered 1..n. Using 0 specifies the first active port. (Default is 0.)

`-o/--output` *report*	Specifies the report type for output. Refer to Report Types for details.
`-d/--detail` *detail*	Specifies the level of detail `0-n` for output. Default is 2.
`-P/--persist`	Only includes data persistent across reboots.
`-H/--hard`	Only includes permanent hardware data.
`-N/--noname`	Omits node and IOC names.
`-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. When used, the `-s`, `-i`, `-C`, and `-a` options are ignored. Not permitted with `-o` *route* and `-F` *route*. `'-'` 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 (such as cable labels and lengths). `'-'` may be used to specify `stdin`.
`-s/--stats`	Gets performance statistics for all ports.
`-i/--interval` *seconds*	Obtains performance statistics over interval *seconds*, clears all statistics, waits interval *seconds*, then generates report. Implies `-s` option.
`-b/--begin` *date_time*	Obtains portCounters over an interval beginning at *date_time*. *date_time* may be a time entered as HH:MM[:SS] or date as mm/dd/YYYY, dd.mm.YYYY, YYYY-mm-dd or date followed by time; for example, "2016-07-04 14:40". Relative times are taken as "x [second|minute|hour|day](s) ago.
`-e/--end` *date_time*	Obtains portCounters over an interval ending at *date_time*. *date_time* may be a time entered as HH:MM[:SS] or date as mm/dd/YYYY, dd.mm.YYYY, YYYY-mm-dd or date followed by time; for example, "2016-07-04 14:40". Relative times are taken as "x [second|minute|hour|day](s) ago.



`-C/--clear`	Clears performance statistics for all ports. Only statistics with error thresholds are cleared. A clear occurs after generating the report.
`-a/--clearall`	Clears all performance statistics for all ports.
`-m/--smadirect`	Accesses fabric information directly from SMA.
`-K/--mkey` *mkey*	Specifies the SMA M_Key for direct SMA query. Default is 0.
`-M/--pmadirect`	Accesses performance statistics using direct PMA.
`-A/--allports`	Gets PortInfo for down switch ports. Uses direct SMA to get this data. If used with `-M`, also gets PMA stats for down-switch ports.
`-c/--config` *file*	Specifies the error thresholds configuration file. Default is `/etc/sysconfig/opa/opamon.conf` file.
`-L/--limit`	Limits operation to exact specified focus with `-F` for port error counters check (`-o` *errors*) and port counters clear (`-C` or `-i`). Normally, the neighbor of each selected port is also checked/cleared. Does not affect other reports.
`-F/--focus` *point*	Specifies the focus area for report. Used for all reports except `route` to limit scope of report. Refer to Point Syntax for details.
`-S/--src` *point*	Specifies the source for trace route, default is local port. Refer to Point Syntax for details.
`-D/--dest` *point*	Specifies the destination for trace route. Refer to Point Syntax for details.
`-Q/--quietfocus`	Excludes focus description from report.

-h and -p options permit a variety of selections:

`-h 0`	First active port in system (default).
`-h 0 -p 0`	First active port in system.
`-h` *x*	First active port on HFI *x*.
`-h` *x* `-p 0`	First active port on HFI *x*.
`-h 0 -p` *y*	Port *y* within system (no matter which ports are active).

`-h` *x* `-p` *y* HFI *x*, port *y*.

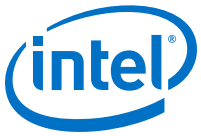
Snapshot-Specific Options

`-r/--routes` Gets routing tables for all switches.

`-V/--vltables` Gets the P-Key tables for all nodes and the QoS VL-related tables for all ports.

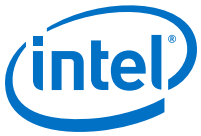
Report Types

`comps`	Summary of all systems and SMs in fabric.
`brcomps`	Brief summary of all systems and SMs in fabric.
`nodes`	Summary of all node types and SMs in fabric.
`brnodes`	Brief summary of all node types and SMs in fabric.
`ious`	Summary of all IO units in the fabric.
`lids`	Summary of all LIDs in the fabric.
`links`	Summary of all links.
`extlinks`	Summary of links external to systems.
`filinks`	Summary of links to FIs.
`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 the fabric.
`linear`	Summary of linear forwarding data base (FDB) for each switch.
`mcast`	Summary of multicast FDB for each switch in the fabric.
`mcgroups`	Summary of multicast groups. When used in conjunction with `-d`, the following report details are possible: • `-d0`: Shows the number of multicast groups • `-d1`: Shows a list of multicast groups • `-d2`: Shows a list of members per multicast group This report can be used with option `-X`.
`portusage`	Summary of ports referenced in linear FDB for each switch, broken down by NodeType of DLID.
`pathusage`	Summary of number of FI to FI paths routed through each switch port.
`treepathusage`	Analysis of number of FI to FI paths routed through each switch port for a FAT tree.
`portgroups`	Summary of adaptive routing port groups for each switch.
`quarantinednodes`	Summary of quarantined nodes.
`validateroutes`	Validates all routes in the fabric.
`validatevlroutes`	Validates all routes in the fabric using SLSC, SCSC, and SCVL tables.
`validatepgs`	Validates all port groups in the fabric.
`validatecreditloops`	Validates topology configuration of the fabric to identify any existing credit loops.
`validatevlcreditloops`	Validates topology configuration of the fabric including SLSC, SCSC, and SCVL tables to identify any existing credit loops.
`vfinfo`	Summary of virtual fabric (vFabric) information.
`vfmember`	Summary of vFabric membership information.

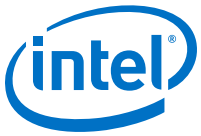
`verifyfis`	Compares fabric (or snapshot) FIs to supplied topology and identifies differences and omissions.
`verifysws`	Compares fabric (or snapshot) switches to supplied topology and identifies differences and omissions.
`verifynodes`	Returns `verifyfis` and `verifysws` reports.
`verifysms`	Compares fabric (or snapshot) SMs to supplied topology and identifies differences and omissions.
`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.
`verifyfilinks`	Compares fabric (or snapshot) links to supplied topology and identify differences and omissions. Limits analysis to links to FIs.
`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 `verifyfis`, `verifysws`, `verifysms`, and `verifylinks` reports.
`all`	Returns `comps`, `nodes`, `ious`, `links`, `extlinks`, `slowconnlinks`, and `errors` reports.
`route`	Traces route between `-S` and `-D` points.
`bfrctrl`	Reports Buffer Control Tables for all ports.
`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`, `-P`, `-H`, 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.
`none`	No report, useful to clear statistics.



Point Syntax

`gid:value`	*`value`* is numeric port GID of form: `subnet:guid`.
`lid:value`	*`value`* is numeric LID.
`lid:value:node`	*`value`* is numeric LID, selects entire `node` with given LID.
`lid:value:port:value2`	*`value`* is numeric LID of node, *`value2`* is port number.
`portguid:value`	*`value`* is numeric port GUID.
`nodeguid:value`	*`value`* is numeric node GUID.
`nodeguid:value1:port:value2`	*`value1`* is numeric node GUID, *`value2`* is port number.
`iocguid:value`	*`value`* is numeric IOC GUID.
`iocguid:value1:port:value2`	*`value1`* is numeric IOC GUID, *`value2`* is port number.
`systemguid:value`	*`value`* is numeric system image GUID.
`systemguid:value1:port:value2`	*`value1`* is the numeric system image GUID, *`value2`* is port number.
`ioc:value`	*`value`* is IOC Profile ID String (IOC Name).
`ioc:value1:port:value2`	*`value1`* is IOC Profile ID String (IOC Name), *`value2`* is port number.
`iocpat:value`	*`value`* is `glob` pattern for IOC Profile ID String (IOC Name).
`iocpat:value1:port:value2`	*`value1`* is `glob` pattern for IOC Profile ID String (IOC Name), *`value2`* is port number.
`ioctype:value`	*`value`* is IOC type (VNIC or SRP).
`ioctype:value1:port:value2`	*`value1`* is IOC type (VNIC or SRP), *`value2`* is port number.
`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 the `glob` pattern for the node description (node name), *`value2`* is port number.
`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.
`nodetype:value`	*`value`* is node type (SW, FI, or RT).
`nodetype:value1:port:value2`	*`value1`* is node type (SW, FI, or RT), *`value2`* is port number.
`rate:value`	*`value`* is string for rate (25g, 50g, 75g, 100g), omits switch mgmt port 0.
`portstate:value`	*`value`* is a string for state (init, armed, active).
`portphysstate:value`	*`value`* is a string for PHYs state (polling, disabled, training, linkup, recovery, offline, test)
`mtucap:value`	*`value`* is MTU size (2048, 4096, 8192, 10240), omits switch mgmt port 0.
`labelpat:value`	*`value`* is `glob` pattern for cable label.
`lengthpat:value`	*`value`* is `glob` pattern for cable length.
`cabledetpat:value`	*`value`* is `glob` pattern for cable details.
`cabinflenpat:value`	*`value`* is `glob` pattern for cable info length.
`cabinfvendnamepat:value`	*`value`* is `glob` pattern for cable info vendor name.
`cabinfvendpnpat:value`	*`value`* is `glob` pattern for cable info vendor part number.
`cabinfvendrevpat:value`	*`value`* is `glob` pattern for cable info vendor revision.
`cabinftype:value`	*`value`* is either `optical`, `passive_copper`, `active_copper`, or `unknown`.



`cabinfvendsnpat:value`	*value* is `glob` pattern for cable info vendor serial number.
`linkdetpat:value`	*value* is `glob` pattern for link details.
`portdetpat:value`	*value* is `glob` pattern for port details.
`sm`	Specifies the master subnet manager (SM).
`smdetpat:value`	*value* is `glob` pattern for SM details.
`route:point1:point2`	Specifies all ports along the routes between the two given points.
`led:value`	*value* is either `on` or `off` for LED port beacon.
`linkqual:value`	Specifies the ports with a link quality equal to *value*.
`linkqualLE:value`	Specifies the ports with a link quality less than or equal to *value*.
`linkqualGE:value`	Specifies the ports with a link quality greater than or equal to *value*.

Examples

`opareport` can generate hundreds of different reports. Commonly generated reports include the following:

```
opareport -o comps -d 3
opareport -o errors -o slowlinks
opareport -o nodes -F portguid:0x00117500a000447b
opareport -o nodes -F nodeguid:0x001175009800447b:port:1
opareport -o nodes -F nodeguid:0x001175009800447b
opareport -o nodes -F 'node:duster hfi1_0'
opareport -o nodes -F 'node:duster hfi1_0:port:1'
opareport -o nodes -F 'nodepat:d*'
opareport -o nodes -F 'nodepat:d*:port:1'
opareport -o nodes -F 'nodedetpat:compute*'
opareport -o nodes -F 'nodedetpat:compute*:port:1'
opareport -o nodes -F nodetype:FI
opareport -o nodes -F nodetype:FI:port:1
opareport -o nodes -F lid:1
opareport -o nodes -F lid:1:node
opareport -o nodes -F lid:1:port:2
opareport -o nodes -F gid:0xfe80000000000000:0x00117500a000447b
opareport -o nodes -F systemguid:0x001175009800447b
opareport -o nodes -F systemguid:0x001175009800447b:port:1
opareport -o nodes -F iocguid:0x00117501300001e0
opareport -o nodes -F iocguid:0x00117501300001e0:port:2
opareport -o nodes -F 'ioc:Chassis 0x001175005000010C, Slot 2, IOC 1'
opareport -o nodes -F 'ioc:Chassis 0x001175005000010C, Slot 2, IOC 1:port:2'
opareport -o nodes -F 'iocpat:*Slot 2*'
opareport -o nodes -F 'iocpat:*Slot 2*:port:2'
opareport -o nodes -F ioctype:VNIC
opareport -o nodes -F ioctype:VNIC:port:2
```

```
opareport -o extlinks -F rate:5g
opareport -o extlinks -F portstate:armed
opareport -o extlinks -F portphysstate:linkup
opareport -o extlinks -F 'labelpat:S1345*'
opareport -o extlinks -F 'lengthpat:11m'
opareport -o extlinks -F 'cableddetpat:*gore*'
opareport -o extlinks -F 'linkdetpat:*core ISL*'
opareport -o extlinks -F 'portdetpat:*mgmt*'
opareport -o links -F mtucap:2048
opareport -o nodes -F sm
opareport -o nodes -F 'smdetpat:primary*'
opareport -o nodes -F 'route:node:duster hfi1_0:node:cuda hfi1_0'
opareport -o nodes -F 'route:node:duster hfi1_0:port:1:node:cuda hfi1_0:port:2'
opareport -s -o snapshot > file
opareport -o topology > topology.xml
opareport -o errors -X file
opareport -s --begin "2 days ago"
opareport -s --begin "12:30" --end "14:00"
```

Other Information

opareport also supports operation with the Fabric Manager Performance Manager (PM)/Performance Manager Agent (PMA). When opareport detects the presence of a PM, it automatically issues any required PortCounter queries and clears to the PM to access the PMs running totals. If a PM is not detected, then opareport directly accesses the PMAs on all the nodes. The -M option can force access to the PMA even if a PM is present.

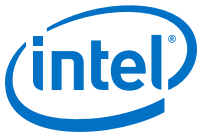
opareport takes advantage of these interfaces to obtain extensive information about the fabric from the subnet manager and the end nodes. Using this information, opareport is able to cross-reference it and produce analysis greatly beyond what any single subnet manager request could provide. As such, it exceeds the capabilities previously available in tools such as opasaquery and opafabricinfo.

opareport obtains and displays counters from the Fabric Manager PM/PA or directly from the fabric PMAs using the -M option.

opareport internally cross-references all this information so its output can be in user-friendly form. Reports include GUIDs, LIDs, and names for components. Obviously, these reports are easiest to read if the end user has taken the time to provide unique names for all the components in the fabric (node names and IOC names). All Intel components support this capability. For hosts, the node names are automatically assigned based on the network host name of the server. For switches and line cards, the names can be assigned using the element managers for each component.

Each run of opareport obtains up-to-date information from the fabric. At the start of the run opareport takes a few seconds to obtain all the fabric data, then it is output to stdout. The reports are sorted by GUIDs and other permanent information so they can be rerun in the future and produce output in the same order even if components have been rebooted. This is useful for comparison using simple tools like diff. opareport permits multiple reports to be requested for a single run (for example, one of each report type).

By default, opareport uses the first active port on the local system. However, if the Management Node is connected to more than one fabric (for example, a subnet), the Intel® Omni-Path Host Fabric Interface (HFI) and port may be specified to select the fabric to analyze.



For additional information, refer to opareport Detailed Information on page 75.

3.3.11 opareports

(All) `opareports` is a front end to `opareport` that provides many of the same options and capabilities. It can also run a report against multiple fabrics or subnets (for example, local host HFI ports). `opareports` can use an input file to augment the reports using additional details from the `topology_input` file.

Syntax

```
opareports [-t portsfile] [-p ports] [-T topology_input]
[opareport arguments]
```

Options

`--help`	Produces full help text.
`-t` *portsfile*	Specifies the file with list of local HFI ports used to access fabric for analysis. Default is `/etc/sysconfig/opa/ports` file.
`-p` *ports*	Specifies the list of local HFI ports used to access fabric for counter clear. Default is first active port. The first HFI in the system is 1. The first port on an HFI is 1. Uses the format hfi:port, for example: `0:0` First active port in system. `0:y` Port *y* within system. `x:0` First active port on HFI *x*. `x:y` HFI *x*, port *y*.
`-T` *topology_input*	Specifies the name of a topology input file to use. The filename may have `%P` as a marker which is replaced with the `hfi:port` being operated on, such as 0:0 or 1:2. The default filename is specified by `FF_TOPOLOGY_FILE` as `/etc/sysconfig/opa/topology.%P.xml`. If `-T NONE` is specified, no topology input file is used.
`opareport` *arguments*	Options are passed to `opareport`. See opareport on page 63 for the full set of options.

Notes: When using opareport arguments, regard the following:

- The -h and -X options are not available.
- The meaning of -p is different for opareports than opareport.
- When run against multiple fabrics, the -x and -o snapshot options are not available.
- When run against multiple fabrics, the -F option is applied to all fabrics.

Examples

```
opareports
opareports -p '1:1 2:1'
```

Environment Variables

The following environment variables are also used by this command:

PORTS	List of ports, used in absence of -t and -p.
PORTS_FILE	File containing list of ports, used in absence of -t and -p.
FF_TOPOLOGY_FILE	File containing topology_input (may have %P marker in filename), used in absence of -T.

Details

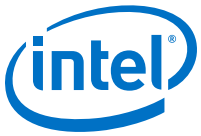
For simple fabrics, the Intel® Omni-Path Fabric Suite FastFabric Toolset host is connected to a single fabric. By default, the first active port on the FastFabric Toolset host is used to analyze the fabric.

However, in more complex fabrics, the FastFabric Toolset host may be connected to more than one fabric or subnet. In this case, you can specify the ports or HFIs to use with one of the following methods:

- On the command line using the -p option.
- In a file specified using the -t option.
- Through the environment variables PORTS or PORTS_FILE.
- Using the ports_file configuration option in /etc/sysconfig/opa/opafastfabric.conf.

If the specified port does not exist or is empty, the first active port on the local system is used. In more complex configurations, you must specify the exact ports to use for all fabrics to be analyzed. For more information, refer to Selection of Devices on page 17.

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



- On the command line using the `-T` option.
- In a file specified through the environment variable `FF_TOPOLOGY_FILE`.
- Using the `ff_topology_file` configuration option in `opafastfabric.conf`.

If the specified file does not exist, no `topology_input` file is used. Alternately the filename can be specified as `NONE` to prevent use of an input file.

For additional information, refer to opareport Detailed Information on page 75.

3.3.12 opareport Detailed Information

This section provides additional information about using `opareport`.

3.3.12.1 opareport Basics

`opareport` 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 `opareport` for a small fabric follows:

```
# opareport
Getting All Node Records...
Done Getting All Node Records
Done Getting All Link Records
Done Getting All Cable Info Records
Done Getting All SM Info Records
Node Type Brief Summary

4 Connected FIs in Fabric:
NodeGUID          Type Name
    Port LID   PortGUID           Width Speed
0x00117501016a35f0 FI coyote hfi1_0
      1 0x0004 0x00117501016a35f0    4    25Gb
0x00117501016a361d FI goblin hfi1_0
      1 0x0003 0x00117501016a361d    4    25Gb
0x00117501016a365f FI ogre hfi1_0
      1 0x0005 0x00117501016a365f    4    25Gb
0x00117501016a366d FI duster hfi1_0
      1 0x0001 0x00117501016a366d    4    25Gb

1 Connected Switches in Fabric:
NodeGUID          Type Name
    Port LID   PortGUID           Width Speed
0x00117500ff6a5619 SW edge1
      0 0x0002 0x00117500ff6a5619    1    25Gb
     12                              4    25Gb
     31                              4    25Gb
     35                              4    25Gb
     39                              4    25Gb

1 Connected SMs in Fabric:
State       GUID               Name
Master      0x00117501016a366d duster hfi1_0
```

Each `opareport` allows for various levels of detail. Increasing detail is shown as further indentation of the additional information. The `-d` option to `opareport` 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.

For example, when the previous report is run at detail level 0, the output is as follows:

```
# opareport -d 0
Getting All Node Records...
Done Getting All Node Records
Done Getting All Link Records
Done Getting All Cable Info Records
Done Getting All SM Info Records
Node Type Brief Summary

4 Connected FIs in Fabric
1 Connected Switches in Fabric
1 Connected SMs in Fabric
```

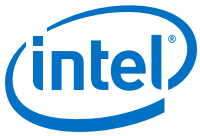
A summary of fabric components is shown in the following example. This report is very similar to `opafabricinfo`. At the next level of detail, the report has more detail:

```
# opareport -d 1
Getting All Node Records...
Done Getting All Node Records
Done Getting All Link Records
Done Getting All Cable Info Records
Done Getting All SM Info Records
Node Type Brief Summary

4 Connected FIs in Fabric:
NodeGUID          Type Name
0x00117501016a35f0 FI ogre hfi1_0
0x00117501016a361d FI goblin hfi1_0
0x00117501016a365f FI coyote hfi1_0
0x00117501016a366d FI duster hfi1_0

1 Connected Switches in Fabric:
NodeGUID          Type Name
0x00117500ff6a5619 SW edge1

1 Connected SMs in Fabric:
State       GUID               Name
Master      0x00117501016a366d duster hfi1_0
```

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 `opareport` 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`	A more verbose form of brnode that provides much greater levels of detail to drill down into all the details of every node, even down to all the port state, IOUs/IOCs/Services, and Port counters.
`comps` and `brcomps`	Very similar to brnodes and nodes, except the reports are organized around systems. The grouping into systems is based on system image GUIDs for each node. This report presents more complex systems (such as servers with multiple HFIs or large switches composed of multiple switch chips).



Note: All Intel switches implement a system image GUID and are therefore properly grouped. However, some third-party devices do not implement the system image GUID and may report a value of 0. In such a case, `opareport` 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 `SystemImageGuid`.

`lids` Similar to `brnodes`, however it is organized and sorted by LID. The output is very concise and provides a simple cross reference of LIDs assigned to each HFI and Switch in the fabric. This information can be useful in interpreting the output from the `linear`, `mcast`, and `portusage` reports.

`ious` Similar to the nodes reports, however the focus is around IOUs/IOCs and IO Services in the fabric. This report identifies various IO devices in the fabric and their capabilities, such as direct-attach storage.

`otherports` Lists all ports that are not connected to this fabric. This report identifies additional ports on FIs or Switches that are not connected to this fabric. For switches, these represent unused ports. For FIs, these may be ports connected to other fabrics or unused ports.

Additionally, `opareport` has reports that analyze the operational characteristics of the fabric and identify bottlenecks and faulty components in the fabric. These reports include:

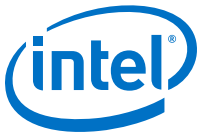
`slowlinks` Identifies links that are running slower than expected, that pinpoints bad cables or components in the fabric. The analysis includes both link speed and width.

`slowconfiglinks` Extends the `slowlinks` report to also report links that have been configured (typically by software) to run at a width or 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` Similar to `slowconfiglinks` in that it reports links that have been configured to run below their potential. However, report does **not** include links that are running slower than expected.

`misconnlinks`	Similar to `slowconnlinks` in that it reports links that have been connected between ports of different speed potential. However, 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 PMA port counters for every node and port in the fabric. All the counters are compared against configured thresholds. Defaults are listed in the `opamon.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 `opamon.si.conf` file can also be used to check for any non-zero values for signal integrity (SI) counters.
`route`	Identifies two end points in the fabric (by node name, node GUID, port name, port GUID, system image GUID, LID, port GID, IOC GIUD, or IOC name), and obtains a list of all the links and components used when these two end points communicate. If there are multiple paths between the end points, such as an FI with 2 connected ports or a system with 2 FIs, the route for every available path is reported based on presently configured routing tables.
`linear`	Shows the linear forwarding table for each switch in the fabric. Used to manually review the routing of unicast traffic in the fabric. For each switch, every unicast LID is shown along with the port it is routed out (egress port), and the neighboring Node and Port. For large fabrics, this report can be quite large.
`mcast`	Shows the multicast forwarding table for each switch in the fabric. Used to manually review the routing of multicast traffic in the fabric. For each switch, every multicast LID is shown along with the list of ports it is routed out. For large fabrics, this report can be quite large.
`portusage`	Provides a summary analysis of the unicast routing in the fabric, in terms of how many LIDs of each node type are routed out a given port. Used for analysis of how balanced the routes in the fabric are, especially for ISLs and core switches. For each switch, all the ports are shown along with the counts of how many unicast LIDs are routed out each port. The total is shown along with HFI-All, HFI-Base, Switch, and Router. • HFI-All includes all LIDs that correspond to an HFI, including LIDs that are the base LID of the HFI and LIDs that map to the HFI though LMC masking. • HFI-Base includes only LIDs that correspond to the base LID of an HFI. HFI-Base is always a subset of HFI-All. • Switch includes all LIDs that correspond to a Switch.



- Router includes all LIDs that correspond to a Router. Only Ports with a non-zero total are shown.

`pathusage` — Computes all the FI to FI dLID paths through the fabric and reports on the usage of each ISL Port (SW to SW link). The `-F` option indicates the switches and the ports on those switches to analyze. Switch Port 0 is always omitted from the analysis. These reports can also be run against snapshots that were performed with the `-r` option.

`treepathusage` — Similar to `pathusage` with the exception that `treepathusage` is applicable only to Fat Tree topologies and provides specific analysis of uplink and downlink paths, indicating what tier each switch is in within the fabric.

3.3.12.2 Simple Topology Verification

`opareport` provides a flexible way to identify changes to the fabric or the appropriate reassembly of the fabric after a move. For example, run `opareport` 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 `opareport` 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 `-P` option to omit information that does not persist across a fabric reboot, for example, LIDs and error counters. In the report, the information is marked out with `xxx`.

If software configuration changes are anticipated, use the `opareport -H` option to only include hardware information. Use this option when adjusting the timeouts the SM configures in the fabric.

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.

3.3.12.3 Advanced Topology Verification

You can use the `-T` option for `opareport` 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 `opareport -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:

```
opareport -o topology -x > topology.xml
```

Later, the fabric can be compared against that topology using a command such as:

```
opareport -T topology.xml -o verifyall
```

Unlike simple `diff` comparisons discussed in Simple Topology Verification, this method of topology verification performs a more context-sensitive comparison and presents information in terms of links, nodes, or SMs that are missing, unexpected, or incorrectly configured.

All the other capabilities of `opareport` 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, or SMs outside the scope of the desired focus.

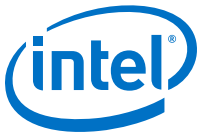
There are multiple variations of advanced topology verification: `verifycas`, `verifysws`, `verifyrtrs`, `verifysms`, `verifylinks`, and `verifyextlinks`. In addition, `verifynodes` and `verifyall` can be used to generate combined reports.

`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 SystemImageGuid, such as within a large Intel® Omni-Path Fabric 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 purposely brief and omits many links, nodes, and SMs.

```
<?xml version="1.0" encoding="utf-8" ?>
<Report>
<LinkSummary>
<Link>
<Rate>25g</Rate>
<MTU>8192</MTU>
<Internal>0</Internal>
<LinkDetails>SampleHost1 to Switch</LinkDetails>
<Cable>
<CableLength>11m</CableLength>
<CableLabel>S4567</CableLabel>
<CableDetails>sample cable model xxx</CableDetails>
</Cable>
<Port>
<NodeGUID>0x0x0011750101660572</NodeGUID>
<PortGUID>0x0011750101660572</PortGUID>
<PortNum>1</PortNum>
<NodeType>FI</NodeType>
```



```
<NodeDesc>SampleHost1 HFI-1</NodeDesc>
<PortDetails>SampleHost1 primary port</PortDetails>
</Port>
<Port>
<NodeGUID>0x0011750007000df6</NodeGUID>
<PortNum>1</PortNum>
<NodeType>SW</NodeType>
<NodeDesc>SampleSwitch1 Leaf 4, Chip A</NodeDesc>
</Port>
</Link>
<Link>
<Rate>25g</Rate>
<MTU>8192</MTU>
<Internal>0</Internal>
<Port>
<NodeGUID>0x0011750101660574</NodeGUID>
<PortGUID>0x0011750101660574</PortGUID>
<PortNum>1</PortNum>
<NodeType>FI</NodeType>
<NodeDesc>SampleHost2 HFI-1</NodeDesc>
</Port>
<Port>
<NodeGUID>0x0011750007000e6d</NodeGUID>
<PortNum>4</PortNum>
<NodeType>SW</NodeType>
<NodeDesc>SampleSwitch1 Leaf 5, Chip A</NodeDesc>
</Port>
</Link>
</LinkSummary>
<Nodes>
<FIs>
<Node id="0x0011750101660576">
<NodeGUID>0x0011750101660576</NodeGUID>
<NodeDesc>SampleHost2 HFI-1</NodeDesc>
<NodeDetails>SampleHost2 only HFI</NodeDetails>
</Node>
</FIs>
<Switches>
<Node id="0x001175000600025a">
<NodeGUID>0x001175000600025a</NodeGUID>
<NodeDesc>SampleSwitch1 Spine 1, Chip A</NodeDesc>
<NodeDetails>core switch</NodeDetails>
</Node>
</Switches>
<SMs>
<SM id="0x0011750101660578:1">
<NodeGUID>0x0011750101660578</NodeGUID>
<NodeDesc>SampleHost2 HFI-1</NodeDesc>
<PortNum>1</PortNum>
<PortGUID>0x0011750101660579</PortGUID>
<NodeType>FI</NodeType>
<NodeType_Int>1</NodeType_Int>
<SMDetails>SampleHost2 SM</SMDetails>
</SM>
</SMs>
</Nodes>
</Report>
```

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. Valid values are 2.5g, 5g, 10g, 20g, 30g, 40g, 60g, 80g, or 120g. 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 from the SMA packets. 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. Valid values are 256, 512, 1024, 2048, and 4096. If not specified, the MTU of the link is not verified.

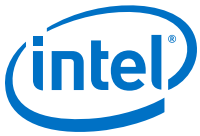
<Internal> A flag indicating if the link is internal or external. A value of 0 indicates external links that are processed by both verifylinks and verifyextlinks. A value of 1 indicates an internal link that is only processed by verifylinks. If omitted, the actual fabric link attributes or the attributes 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, route, 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, route, verifylinks, and verifyextlinks reports. Intel recommends you use this field to describe the identifying label attached to the cable using text such as S4576. This field can also be used by the labelpat focus option to select the link. 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, route, verifylinks, and verifyextlinks reports. Intel recommends you use this field to describe the type, model, and/or manufacturer of the cable. This field can also be used by the `cabledetpat` focus option to select the link.

`<Port>` A container tag providing additional information about the two ports that make up the link.

`<Port>` allows the following tags:

`<NodeGUID>` Node GUID reported by the SMA for the given FI, switch, or router.

`<PortGUID>` Port GUID reported by the SMA for the given FI, switch, or router.

Note: Switches only have PortGuids for port 0 (the internal management port), while FIs and routers have a unique GUID for every port.

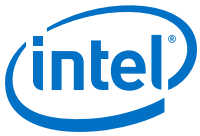
`<PortNum>` Port Number within the FI, switch, or router.

<NodeDesc>	Node Description reported by the FI, switch, or router. Intel recommends that you configure a unique value for this field in each node in your fabric. For example, Intel® Omni-Path Fabric Host Software Linux* hosts use the combination of Linux hostname and HFI number to create a unique NodeDesc.
<NodeType>	Node type reported by the node. Values include: FI, SW, or RT. Alternatively, an integer value <NodeType_Int> may be provided based on the values for NodeType from the SMA packets. 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, route, 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:

- NodeGUID, PortNum
- NodeGUID, PortGUID
- NodeGUID – if given FI has exactly 1 port.
- NodeDesc, PortNum
- NodeDesc, PortGUID
- NodeDesc – if given FI has exactly 1 port.
- PortGUID, PortNum – useful to select ports other than 0 on a switch.
- PortGUID

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:

- `NodeGuid`, `NodeDesc` – `NodeDesc` is not used to resolve port.
- `NodeGuid`, `PortNum`, `PortGuid` – `PortGuid` is not used to resolve port.
- `NodeDesc`, `PortNum`, `PortGuid` – `PortGuid` 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.

`<FIs>` Container tag describing all the FIs 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>`.

`<Routers>` Container tag describing all the Routers expected in the fabric. Many instances of this tag can occur per `<Nodes>`.

`<SMs>` Container tag describing all the SMs expected in the fabric. Many instances of this tag can occur per `<Nodes>`.

`<Node>` Container tag describing a single node (FI, SW, or RT). Many instances of this tag can occur per `<FIs>`, `<Switches>`, or `<Routers>`.

`<Node>` allows the following tags:

`<NodeGUID>` Node GUID reported by the SMA for the given FI, Switch, or Router.

`<NodeDesc>` Node Description reported by the FI, switch, or router. Intel recommends that you configure a unique value for this field in each node in your fabric. For example, Intel® Omni-Path Fabric Host Software Linux* hosts use the combination of Linux hostname and HFI number to create a unique NodeDesc.

`<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, route, comps, verifycas, verifysws, verifyrts, 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 (FI, Switch, or Router) 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:

- `NodeGUID`
- `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 `verifycas`, `verifysws`, or `verifyrtrs`, all the input provided is verified against the actual fabric and any discrepancies are reported.

An example of redundant information:

- `NodeGuid`, `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.

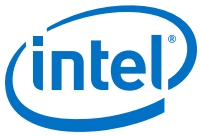
`<SM>` Container tag describing a single SM. Many instances of this tag can occur per `<SMs>`.

`<SM>` allows the following tags:

`<NodeGUID>` Node GUID reported by the SMA for the given FI, switch, or router that is running the SM.

`<NodeDesc>` Node Description reported by the FI, switch, or router that is running the SM. Intel recommends that you configure a unique value for this field in each node in your fabric. For example, Intel® Omni-Path Fabric Host Software Linux* hosts use the combination of Linux hostname and HFI number to create a unique NodeDesc.

`<PortGUID>` Port GUID reported by the SMA for the given FI, switch, or router that is running the SM.



Note: Switches only have PortGuids for port 0 (the internal management port), while FIs and routers have a unique GUID for every port.

`<PortNum>` Port Number within the FI, switch, or router that is running the SM.

`<NodeType>` Node type reported by the node that is running the SM. Values include: FI, SW, or RT. Alternatively, an integer value `<NodeType_Int>` may be provided based on the values for NodeType from the SMA packets. 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 SM is not verified.

`<SMDetails>` Free form text field of up to 64 characters. This field is optional. When provided, this is output as a SM attribute in all reports that show SM details, such as comps and verifysms reports. Intel recommends you use this field to describe the purpose of the SM. This field can also be used by the smdetpat focus option to select the SM.

The previous fields are used to associate a port running an SM in the `topology_input` file with an actual port in the fabric, also called resolving the SM. 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:

- `NodeGUID`, `PortNum`
- `NodeGUID`, `PortGUID`
- `NodeGUID` – if given FI has exactly 1 active port or is a switch.
- `NodeDesc`, `PortNum`
- `NodeDesc`, `PortGUID`
- `NodeDesc` – if given FI has exactly 1 active port or is a switch.
- `PortGUID`, `PortNum` – limited usefulness.
- `PortGUID`

If `NodeDesc` is used to specify SM ports, the fabric must be configured such that each `NodeDesc` is unique. Otherwise, the `<SM>` 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 for an SM. However, during `verifysms` all the input provided is verified against the actual fabric and any discrepancies are reported.

Some examples of redundant information:

- `NodeGuid`, `NodeDesc` – `NodeDesc` is not used to resolve port.
- `NodeGuid`, `PortNum`, `PortGuid` – `PortGuid` is not used to resolve port.
- `NodeDesc`, `PortNum`, `PortGuid` – `PortGuid` is not used to resolve port.

The `NodeType` field is never used during resolution, it is only used during verification.

3.3.12.4 Augmented Report Information

As discussed in Advanced Topology Verification, a `topology_input` file includes additional information including cable (length, label, details), links (details), ports (details), nodes (details) and SMs (details).

A `topology_input` 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.

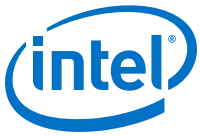
3.3.12.5 Focused Reports

One of the more powerful features of `opareport` 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, node GUID, node type, port GUID, IOC name, IOC name pattern, IOC GUID, IOC type, system image GUID, port GID, port rate, port state, port physical state, MTU capability, LID, link quality indicator, cable info for cable length, cable info for vendor name, cable info for vendor part number, cable info for vendor rev, cable info for vendor serial number, or SM.

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 on. 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`, `-C` or `-i`, 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 FI 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 LID performs reverse lookup on that LID 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.

3.3.12.6 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, node guid, node type, IOC name, IOC name pattern, IOC GUID, IOC type, and system image GUID also allow for a port number specifier. This limits the focus to the given port number. If the selection resolves to multiple switches or FIs, 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.

- Route between points

  This method focuses on all the ports involved in a particular route and can be an excellent way to determine a performance or error situation reported between two specific points in the fabric. For example, MPI may report `StatusTimeoutRetry` between two processes in its run.

  * syadmin fields supplied in a topology file (typically generated by opaxlattopology or opaxlattopology_cust) including cable labels, cable details, planned cable length, link details, port details, and SM details.

- glob-style patterns

  You can use a wildcard focus for the node name, IOC name, node details, cable label, cable length, cable details, cable vendor name, cable vendor part number, cable vendor rev, cable vendor serial number, link details, port details, or SM 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.

3.3.12.7 Focus Examples

Examples of using the focus options are shown in the following list:

```
opareport -o nodes -F portguid:0x00117500a000447b
opareport -o nodes -F nodeguid:0x001175009800447b:port:1
opareport -o nodes -F nodeguid:0x001175009800447b
opareport -o nodes -F node:duster
opareport -o nodes -F node:duster:port:1
opareport -o nodes -F 'nodepat:d*'
opareport -o nodes -F 'nodepat:d*:port:1'
opareport -o nodes -F nodetype:FI
opareport -o nodes -F nodetype:FI:port:1
opareport -o nodes -F lid:1
opareport -o nodes -F lid:1:node
opareport -o nodes -F gid:0xfe80000000000000:0x00117500a000447b
opareport -o nodes -F systemguid:0x001175009800447b
opareport -o nodes -F systemguid:0x001175009800447b:port:1
opareport -o nodes -F iocguid:0x00117501300001e0
opareport -o nodes -F iocguid:0x00117501300001e0:port:2
opareport -o nodes -F 'ioc:Chassis 0x001175005000010C, Slot 2, IOC 1'
opareport -o nodes -F 'ioc:Chassis 0x001175005000010C, Slot 2, IOC 1:port:2'
opareport -o nodes -F 'iocpat:*Slot 2*'
opareport -o nodes -F 'iocpat:*Slot 2*:port:2'
opareport -o nodes -F ioctype:XXXX
opareport -o nodes -F ioctype:XXXX:port:2
opareport -o nodes -F sm
opareport -o nodes -F route:node:duster:node:cuda
opareport -o nodes -F route:node:duster:port:1:node:cuda:port:2
```

3.3.12.8 Scriptable Output

`opareport` permits custom scripting. As previously mentioned, options like `-H`, `-P`, and `-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 `opareport` output, enabling you to create scripts to further search and refine report output formats.

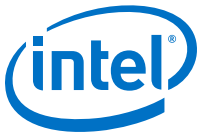
The `opaxmlextract` tool easily converts between XML files and delimited text files. For more information, see opaxmlextract on page 204.

You can integrate `opareport` into custom scripts. You can also generate customer-specific new report formats and cross-reference `opareport` with other site-specific information.

3.3.12.9 Monitor for Fabric Changes Using opareport

`opareport` 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

opareport -o all -d 5 -P > $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
```

3.3.12.10 Sample Outputs

Analyze all ports in fabric for errors, inconsistent connections, bad cables

```
[root@duster root]# opareport -o errors -o slowlinks
Links running slower than expected Summary

Links running slower than expected:
Rate NodeGUID          Port Type Name
     Active                          Enabled
     Lanes, Used(Tx), Used(Rx), Rate,  Lanes,   DownTo,  Rates
100g 0x00117501025019ab  44 SW   edge1
     4      3          3         25Gb   1,2,3,4  3,4      25Gb
<->  0x0011750102513139  44 SW   edge2
     4      3          3         25Gb   1,2,3,4  3,4      25Gb
100g 0x00117501025019ab  48 SW   edge1
     4      3          4         25Gb   1,2,3,4  3,4      25Gb
<->  0x0011750102513139  48 SW   edge2
     4      4          3         25Gb   1,2,3,4  3,4      25Gb
96 of 96 Links Checked, 2 Errors found
-------------------------------------------------------------------------------
Links with errors > threshold Summary

Configured Error Thresholds:
    LinkQualityIndicator           5
    LinkDowned                     3
    RcvErrors                      100
    ExcessiveBufferOverruns        3
    LinkErrorRecovery              3
    LocalLinkIntegrityErrors       3
    XmitConstraintErrors           10
    RcvConstraintErrors            10
    CongDiscards                   100
96 of 96 Links Checked, 0 Errors found
-------------------------------------------------------------------------------
```

Identify the route between two nodes

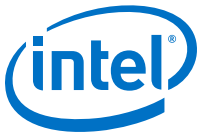
```
[root@goblin root]#   opareport -o route -S node:"goblin hfi1_0" -D node:"orc
hfi1_0"
Getting All Node Records...
Done Getting All Node Records
Done Getting All Link Records
Done Getting All Cable Info Records
Done Getting All SM Info Records
Routes Summary Between:
    Node: 0x001175010157409d FI goblin hfi1_0
and Node: 0x001175010157403d FI orc hfi1_0

Routes between ports:
     0x001175010157409d   1 FI   goblin hfi1_0
and  0x001175010157403d   1 FI   orc hfi1_0
2 Paths
    SGID: 0xfe80000000000000:001175010157409d
    DGID: 0xfe80000000000000:001175010157403d
    SLID: 0x0001 DLID: 0x0018 Reversible: Y PKey: 0x8001
    Raw: N FlowLabel: 0x00000 HopLimit: 0x00 TClass: 0x00
    SL:  0 Mtu:  8192 Rate: 100g PktLifeTime:  134 ms Pref: 0
        Rate NodeGUID          Port Type Name
```

```
        100g 0x001175010157409d   1 FI   goblin hfi1_0
        ->   0x00117501025131cb  44 SW   edge1
        100g 0x00117501025131cb  40 SW   edge2
        ->   0x001175010157403d   1 FI   orc hfi1_0
    2 Links Traversed
    SGID: 0xfe80000000000000:001175010157409d
    DGID: 0xfe80000000000000:001175010157403d
    SLID: 0x0001 DLID: 0x0018 Reversible: Y PKey: 0xffff
    Raw: N FlowLabel: 0x00000 HopLimit: 0x00 TClass: 0x00
    SL:  0 Mtu:  8192 Rate: 100g PktLifeTime:  134 ms Pref: 0
        Rate NodeGUID          Port Type Name
        100g 0x001175010157409d   1 FI   goblin hfi1_0
        ->   0x00117501025131cb  44 SW   edge1
        100g 0x00117501025131cb  40 SW   edge1
        ->   0x001175010157403d   1 FI   orc hfi1_0
    2 Links Traversed
```

Obtain very detailed information about nodes

Note: To shorten the length of the output, the following example focuses on only one node.

```
[root@duster root]# opareport -o nodes -F node:"duster hfi1_0" -d 5 -s
Getting All Node Records...
Done Getting All Node Records
Done Getting All Link Records
Done Getting All Cable Info Records
Done Getting All SM Info Records
Getting All Port Counters...
Done Getting All Port Counters
Node Type Summary
Focused on:
    Node: 0x001175010157409d FI duster hfi1_0

48 Connected FIs in Fabric:
    Name: duster hfi1_0
        NodeGUID: 0x001175010157409d Type: FI
        Ports: 1 PartitionCap: 16 SystemImageGuid: 0x001175010157409d
        BaseVer: 128 SmaVer: 128 VendorID: 0x1175 DeviceID: 0x24f0 Rev: 0x0
    1 Connected Ports:
        PortNum:   1 LID: 0x0001 GUID: 0x001175010157409d
            Neighbor:  Name: edge1
                NodeGUID: 0x00117501025131cb Type: SW PortNum:  44
            LocalPort: 1   PortState: Active                PhysState: LinkUp
            IsSMConfigurationStarted: True  NeighborNormal: True
            PortType: Standard
            LID:    0x0001              LMC: 0              Subnet:
0xfe80000000000000
            SMLID:  0x0001   SMSL:  0   RespTimeout:   32 us  SubnetTimeout:  536
ms
            M_KEY:  0x0000000000000000  Lease:     0 s          Protect: Read-only
            MTU  Supported: (0x6) 8192 bytes
            VLStallCount (per VL):  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
            MTU  Active by VL:
            00: 8192 01:    0 02:    0 03:    0 04:    0 05:    0 06:    0 07:
0
            08:    0 09:    0 10:    0 11:    0 12:    0 13:    0 14:    0 15:
2048
            16:    0 17:    0 18:    0 19:    0 20:    0 21:    0 22:    0 23:
0
            24:    0 25:    0 26:    0 27:    0 28:    0 29:    0 30:    0 31:
0
            LinkWidth: Active:       4  Supported:    1,2,3,4  Enabled:          4
            LinkWidthDnGrade: ActiveTx:  4 Rx:  4 Supported:    1,2,3,4
Enabled:        3,4
            PortLinkMode: Active: STL      Supported: STL             Enabled:
STL
            LinkSpeed: Active:    25Gb  Supported:       25Gb  Enabled:       25Gb
```

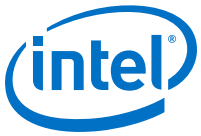


```
            SM_TrapQP: 0x0  SA_QP: 0x1 IPAddr Prim/Sec: :: / 0.0.0.0
            VLs:       Active:     8+1  Supported:        8+1
                HOQLife (Per VL):
                VL 0: 0x0 VL 1: 0x0 VL 2: 0x0 VL 3: 0x0 VL 4: 0x0
                VL 5: 0x0 VL 6: 0x0 VL 7: 0x0 VL 8: 0x0 VL 9: 0x0
                VL10: 0x0 VL11: 0x0 VL12: 0x0 VL13: 0x0 VL14: 0x0
                VL15: 0x0 VL16: 0x0 VL17: 0x0 VL18: 0x0 VL19: 0x0
                VL20: 0x0 VL21: 0x0 VL22: 0x0 VL23: 0x0 VL24: 0x0
                VL25: 0x0 VL26: 0x0 VL27: 0x0 VL28: 0x0 VL29: 0x0
                VL30: 0x0 VL31: 0x0

            VL Arb Cap:  High:      16        Low:         16  HiLimit:    0
PreemptLimit    0
            VLFlowControlDisabledMask: 0x00000000
            NeighborMode  MgmtAllowed: Yes  FwAuthenBypass:  On
NeighborNodeType: Switch
            Capability 0x00410022: CN CM APM SM
            Capability3 0x0008: SS
            Violations: M_Key:     0 P_Key:     0 Q_Key:     0
            PortMode ActiveOptimize: Off PassThrough: Off VLMarker: Off
            FlitCtrlInterleave Distance Max:  1 Enabled:  1
            MaxNestLevelTxEnabled: 0 MaxNestLevelRxSupported: 0
            SmallPktLimit: 0x00 MaxSmallPktLimit: 0x00 PreemptionLimit: 0x00
            FlitCtrlPreemption MinInital: 0x0000 MinTail: 0x0000 LargePktLim: 0x00
            BufferUnits: VL15Init 0x0110; VL15CreditRate 0x00; CreditAck 0x0;
BufferAlloc 0x3
            PortErrorActions: 0x172000: CE-UVLMCE-BCDCE-BTDCE-BHDR-BVLM
            ReplayDepth Buffer 0x80; Wire 0x0a
            DiagCode: 0x0000
            OverallBufferSpace: 0x093f
            P_Key Enforcement: In: Off Out: Off
            Performance: Transmit
                Xmit Data                    42 MB (5278567 Flits)
                Xmit Pkts                 303029
                MC Xmt Pkts                    0
            Performance: Receive
                Rcv Data                   220 MB (27592828 Flits)
                Rcv Pkts                  303026
                MC Rcv Pkts                    0
            Performance: Congestion
                Congestion Discards                    0
                Rcv FECN                               0
                Rcv BECN                               0
                Mark FECN                              0
                Xmit Time Congestion                   0
                Xmit Wait                              0
            Performance: Bubbles
                Rcv Bubble                        240092
                Xmit Wasted BW                         0
                Xmit Wait Data                         0
            Link Qual Indicator                        5 (Excellent)
            Errors: Signal Integrity
                Uncorrectable Errors                   0
               Link Downed                            0
                Rcv Errors                             0
                Exc. Buffer Overrun                    0
                FM Config Errors                       0
                Link Error Recovery                    0
                Local Link Integ Err                   0
                Rcv Rmt Phys Err                       0
            Errors: Security
                Xmit Constraint                        0
                Rcv Constraint                         0
            Errors: Other
                Rcv Sw Relay Err                       0
                Xmit Discards                          0
            QSFP Interpreted CableInfo:
                Identifier: 0xd
                ExtIdentifier: Power Class 1, 1.5W max
                Connector: 0x23
                NominalBR: 25 Gb
```

```
                OM2Length: 0m
                OM3Length: 0m
                OM4Length: 1m
                DeviceTech: Passive copper cable
                VendorName: FCI Electronics
                VendorOUI: 0xfc7ce7
                VendorPN: 10131941-2010LF
                VendorRev: 2
                MaxCaseTemp: 0 C
                CC_BASE: 0xe0
                TxCDR: N/A
                TxInpEqFixProg: False
                TxInpEqAutoAdp: False
                TxSquelchImp: False
                RxCDR: N/A
                RxOutpEmphFixProg: False
                RxOutpAmplFixProg: False
                MemPage02Provided: False
                MemPage01Provided: False
                VendorSN: CN1449FA102L0009
                DateCode: 2014/12/04-
                CC_EXT: 0x65
                CertCableFlag: N
                ReachClass: 4
                CertDataRates: 4x25G
1 Matching FIs Found

4 Connected Switches in Fabric:
0 Matching Switches Found

1 Connected SMs in Fabric:
    State: Master      Name: duster hfi1_0
        NodeGUID: 0x001175010157409d Type: FI
        PortNum:   1 LID: 0x0001 PortGUID: 0x001175010157409d
        SM_Key: 0x0000000000000000 Priority:   0 ActCount: 0x00019091
1 Matching SMs Found
--------------------------------------------------------------------------------
```

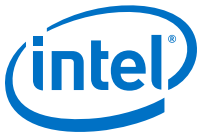
Identify connections and links composing the fabric

```
[goblin1 root@goblin1]# opareport -o links
Getting All Node Records...
Done Getting All Node Records
Done Getting All Link Records
Done Getting All Cable Info Records
Done Getting All SM Info Records
Link Summary

96 Links in Fabric:
Rate NodeGUID          Port Type Name
100g 0x001175010157401b   1 FI   goblin8 hfi1_0
<->  0x00117501025131cb   7 SW   edge2
100g 0x001175010157403d   1 FI   goblin2 hfi1_0
<->  0x00117501025131cb  40 SW   edge2
100g 0x0011750101574053   1 FI   goblin12 hfi1_0
<->  0x00117501025131cb  23 SW   edge2
100g 0x001175010157405c   1 FI   goblin16 hfi1_0
<->  0x00117501025019ab  33 SW   edge1
100g 0x001175010157406c   1 FI   goblin13 hfi1_0
<->  0x00117501025019ab  42 SW   edge1
100g 0x0011750101574071   1 FI   goblin18 hfi1_0
<->  0x00117501025131cb   4 SW   edge2
100g 0x0011750101574074   1 FI   goblin15 hfi1_0
<->  0x00117501025019ab  41 SW   edge1
100g 0x0011750101574077   1 FI   goblin20 hfi1_0
<->  0x00117501025131cb   3 SW   edge2
100g 0x001175010157409d   1 FI   goblin1 hfi1_0
<->  0x00117501025131cb  44 SW   edge2
100g 0x00117501015740bb   1 FI   goblin22 hfi1_0
```



```
<->  0x00117501025019ab   2 SW   edge1
100g 0x00117501015740bd   1 FI   goblin3 hfi1_0
<->  0x00117501025131cb  43 SW   edge2
100g 0x00117501015740db   1 FI   goblin21 hfi1_0
<->  0x00117501025019ab  10 SW   edge1
100g 0x00117501015740e0   1 FI   goblin41 hfi1_0
<->  0x00117501025019ab  22 SW   edge1
100g 0x00117501015740e3   1 FI   goblin33 hfi1_0
<->  0x00117501025131cb  48 SW   edge2
100g 0x0011750101574e8b   1 FI   goblin25 hfi1_0
<->  0x00117501025019ab  46 SW   edge1
100g 0x0011750101574f08   1 FI   goblin45 hfi1_0
<->  0x00117501025019ab  18 SW   edge1
100g 0x0011750101574f6c   1 FI   goblin42 hfi1_0
<->  0x00117501025019ab  30 SW   edge1
100g 0x0011750101574fea   1 FI   goblin29 hfi1_0
<->  0x00117501025131cb  32 SW   edge2
100g 0x0011750101575021   1 FI   goblin46 hfi1_0
<->  0x00117501025019ab  25 SW   edge1
100g 0x001175010157504e   1 FI   goblin47 hfi1_0
<->  0x00117501025019ab  17 SW   edge1
100g 0x0011750101575068   1 FI   goblin10 hfi1_0
<->  0x00117501025131cb  24 SW   edge2
100g 0x0011750101575082   1 FI   goblin23 hfi1_0
<->  0x00117501025019ab   9 SW   edge1
100g 0x00117501015750a1   1 FI   goblin48 hfi1_0
<->  0x00117501025019ab  26 SW   edge1
100g 0x001175010157513c   1 FI   goblin34 hfi1_0
<->  0x00117501025131cb  47 SW   edge2
100g 0x0011750101575153   1 FI   goblin35 hfi1_0
<->  0x00117501025131cb  36 SW   edge2
100g 0x001175010157515e   1 FI   goblin4 hfi1_0
<->  0x00117501025131cb  39 SW   edge2
100g 0x0011750101575188   1 FI   goblin17 hfi1_0
<->  0x00117501025131cb  16 SW   edge2
100g 0x00117501015751b8   1 FI   goblin11 hfi1_0
<->  0x00117501025131cb  27 SW   edge2
100g 0x00117501015751c9   1 FI   goblin30 hfi1_0
<->  0x00117501025131cb  19 SW   edge2
100g 0x00117501015751d6   1 FI   goblin6 hfi1_0
<->  0x00117501025131cb   8 SW   edge2
100g 0x00117501015751dd   1 FI   goblin37 hfi1_0
<->  0x00117501025019ab  14 SW   edge1
100g 0x00117501015751df   1 FI   goblin43 hfi1_0
<->  0x00117501025019ab  29 SW   edge1
100g 0x00117501015751e5   1 FI   goblin31 hfi1_0
<->  0x00117501025131cb  20 SW   edge2
100g 0x00117501015751ef   1 FI   goblin38 hfi1_0
<->  0x00117501025019ab   5 SW   edge1
100g 0x00117501015751ff   1 FI   goblin19 hfi1_0
<->  0x00117501025131cb  15 SW   edge2
100g 0x0011750101575f28   1 FI   goblin39 hfi1_0
<->  0x00117501025019ab   6 SW   edge1
100g 0x0011750101575f63   1 FI   goblin26 hfi1_0
<->  0x00117501025019ab  45 SW   edge1
100g 0x0011750101575f6a   1 FI   goblin44 hfi1_0
<->  0x00117501025019ab  21 SW   edge1
100g 0x0011750101575fa1   1 FI   goblin40 hfi1_0
<->  0x00117501025019ab  13 SW   edge1
100g 0x0011750101575fba   1 FI   goblin7 hfi1_0
<->  0x00117501025131cb  11 SW   edge2
100g 0x0011750101575feb   1 FI   goblin14 hfi1_0
<->  0x00117501025019ab  34 SW   edge1
100g 0x001175010157e3d1   1 FI   goblin36 hfi1_0
<->  0x00117501025131cb  35 SW   edge2
100g 0x001175010157e3f0   1 FI   goblin24 hfi1_0
<->  0x00117501025019ab   1 SW   edge1
100g 0x001175010157e3f3   1 FI   goblin32 hfi1_0
<->  0x00117501025131cb  31 SW   edge2
100g 0x001175010157e406   1 FI   goblin27 hfi1_0
<->  0x00117501025019ab  38 SW   edge1
```

```
100g 0x001175010157e40e   1 FI   goblin9 hfi1_0
<->  0x00117501025131cb  28 SW   edge2
100g 0x001175010157e418   1 FI   goblin28 hfi1_0
<->  0x00117501025019ab  37 SW   edge1
100g 0x001175010157e427   1 FI   goblin5 hfi1_0
<->  0x00117501025131cb  12 SW   edge2
100g 0x00117501025019ab   3 SW   edge1
<->  0x0011750102513145   3 SW   edge4
100g 0x00117501025019ab   4 SW   edge1
<->  0x0011750102513145   4 SW   edge4
100g 0x00117501025019ab   7 SW   edge1
<->  0x0011750102513145   7 SW   edge4
100g 0x00117501025019ab   8 SW   edge1
<->  0x0011750102513145   8 SW   edge4
100g 0x00117501025019ab  11 SW   edge1
<->  0x0011750102513139  11 SW   edge3
100g 0x00117501025019ab  12 SW   edge1
<->  0x0011750102513139  12 SW   edge3
100g 0x00117501025019ab  15 SW   edge1
<->  0x0011750102513139  15 SW   edge3
100g 0x00117501025019ab  16 SW   edge1
<->  0x0011750102513139  16 SW   edge3
100g 0x00117501025019ab  19 SW   edge1
<->  0x0011750102513145  19 SW   edge4
100g 0x00117501025019ab  20 SW   edge1
<->  0x0011750102513145  20 SW   edge4
100g 0x00117501025019ab  23 SW   edge1
<->  0x0011750102513145  23 SW   edge4
100g 0x00117501025019ab  24 SW   edge1
<->  0x0011750102513145  24 SW   edge4
100g 0x00117501025019ab  27 SW   edge1
<->  0x0011750102513139  27 SW   edge3
100g 0x00117501025019ab  28 SW   edge1
<->  0x0011750102513139  28 SW   edge3
100g 0x00117501025019ab  31 SW   edge1
<->  0x0011750102513139  31 SW   edge3
100g 0x00117501025019ab  32 SW   edge1
<->  0x0011750102513139  32 SW   edge3
100g 0x00117501025019ab  35 SW   edge1
<->  0x0011750102513145  35 SW   edge4
100g 0x00117501025019ab  36 SW   edge1
<->  0x0011750102513145  36 SW   edge4
100g 0x00117501025019ab  39 SW   edge1
<->  0x0011750102513145  39 SW   edge4
100g 0x00117501025019ab  40 SW   edge1
<->  0x0011750102513145  40 SW   edge4
100g 0x00117501025019ab  43 SW   edge1
<->  0x0011750102513139  43 SW   edge3
100g 0x00117501025019ab  44 SW   edge1
<->  0x0011750102513139  44 SW   edge3
100g 0x00117501025019ab  47 SW   edge1
<->  0x0011750102513139  47 SW   edge3
100g 0x00117501025019ab  48 SW   edge1
<->  0x0011750102513139  48 SW   edge3
100g 0x0011750102513139   1 SW   edge3
<->  0x00117501025131cb   1 SW   edge2
100g 0x0011750102513139   2 SW   edge3
<->  0x00117501025131cb   2 SW   edge2
100g 0x0011750102513139   5 SW   edge3
<->  0x00117501025131cb   5 SW   edge2
100g 0x0011750102513139   6 SW   edge3
<->  0x00117501025131cb   6 SW   edge2
100g 0x0011750102513139  17 SW   edge3
<->  0x00117501025131cb  17 SW   edge2
100g 0x0011750102513139  18 SW   edge3
<->  0x00117501025131cb  18 SW   edge2
100g 0x0011750102513139  21 SW   edge3
<->  0x00117501025131cb  21 SW   edge2
100g 0x0011750102513139  22 SW   edge3
<->  0x00117501025131cb  22 SW   edge2
100g 0x0011750102513139  33 SW   edge3
```



```
<->  0x00117501025131cb  33 SW   edge2
100g 0x0011750102513139  34 SW   edge3
<->  0x00117501025131cb  34 SW   edge2
100g 0x0011750102513139  37 SW   edge3
<->  0x00117501025131cb  37 SW   edge2
100g 0x0011750102513139  38 SW   edge3
<->  0x00117501025131cb  38 SW   edge2
100g 0x0011750102513145   9 SW   edge4
<->  0x00117501025131cb   9 SW   edge2
100g 0x0011750102513145  10 SW   edge4
<->  0x00117501025131cb  10 SW   edge2
100g 0x0011750102513145  13 SW   edge4
<->  0x00117501025131cb  13 SW   edge2
100g 0x0011750102513145  14 SW   edge4
<->  0x00117501025131cb  14 SW   edge2
100g 0x0011750102513145  25 SW   edge4
<->  0x00117501025131cb  25 SW   edge2
100g 0x0011750102513145  26 SW   edge4
<->  0x00117501025131cb  26 SW   edge2
100g 0x0011750102513145  29 SW   edge4
<->  0x00117501025131cb  29 SW   edge2
100g 0x0011750102513145  30 SW   edge4
<->  0x00117501025131cb  30 SW   edge2
100g 0x0011750102513145  41 SW   edge4
<->  0x00117501025131cb  41 SW   edge2
100g 0x0011750102513145  42 SW   edge4
<->  0x00117501025131cb  42 SW   edge2
100g 0x0011750102513145  45 SW   edge4
<->  0x00117501025131cb  45 SW   edge2
100g 0x0011750102513145  46 SW   edge4
<->  0x00117501025131cb  46 SW   edge2
```

Reverse lookup

The following example translates a LID or GUID into the information about the node or port represented.

```
[root@duster duster]# opareport -o nodes -F lid:5
Getting All Node Records...
Done Getting All Node Records
Done Getting All Link Records
Done Getting All Cable Info Records
Done Getting All SM Info Records
Node Type Summary
Focused on:
    Port:   1 0x0011750101574071
        in Node: 0x0011750101574071 FI goblin2 hfi1_0

48 Connected FIs in Fabric:
    Name: goblin2 hfi1_0
        NodeGUID: 0x0011750101574071 Type: FI
        Ports: 1 PartitionCap: 16 SystemImageGuid: 0x0011750101574071
        BaseVer: 128 SmaVer: 128 VendorID: 0x1175 DeviceID: 0x24f0 Rev: 0x0
    1 Connected Ports:
        PortNum:   1 LID: 0x0005 GUID: 0x0011750101574071
            Neighbor:  Name: edge1
                NodeGUID: 0x00117501025131cb Type: SW PortNum:   4
            Width:    4 Speed:    25Gb Downgraded?  No
1 Matching FIs Found

4 Connected Switches in Fabric:
0 Matching Switches Found

1 Connected SMs in Fabric:
0 Matching SMs Found
---------------------------------------
```

Forward lookup

The following example returns information about nodes or IOCs listed by name.

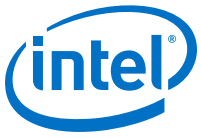
```
[root@duster root]# opareport -o nodes -F "node:goblin2 hfi1_0"
Getting All Node Records...
Done Getting All Node Records
Done Getting All Link Records
Done Getting All Cable Info Records
Done Getting All SM Info Records
Node Type Summary
Focused on:
    Node: 0x0011750101574071 FI goblin2 hfi1_0

48 Connected FIs in Fabric:
    Name: goblin2 hfi1_0
        NodeGUID: 0x0011750101574071 Type: FI
        Ports: 1 PartitionCap: 16 SystemImageGuid: 0x0011750101574071
        BaseVer: 128 SmaVer: 128 VendorID: 0x1175 DeviceID: 0x24f0 Rev: 0x0
    1 Connected Ports:
        PortNum:   1 LID: 0x0005 GUID: 0x0011750101574071
            Neighbor:  Name: edge1
                NodeGUID: 0x00117501025131cb Type: SW PortNum:   4
            Width:    4 Speed:    25Gb Downgraded?  No
1 Matching FIs Found

4 Connected Switches in Fabric:
0 Matching Switches Found

1 Connected SMs in Fabric:
0 Matching SMs 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@duster root]# opareport -o nodes -F "node:goblin2 hfi1_0" -d 5 -P
Getting All Node Records...
Done Getting All Node Records
Done Getting All Link Records
Done Getting All Cable Info Records
Done Getting All SM Info Records
Node Type Summary
Focused on:
    Node: 0x0011750101574071 FI goblin2 hfi1_0

48 Connected FIs in Fabric:
    Name: goblin2 hfi1_0
        NodeGUID: 0x0011750101574071 Type: FI
        Ports: 1 PartitionCap: 16 SystemImageGuid: 0x0011750101574071
        BaseVer: 128 SmaVer: 128 VendorID: 0x1175 DeviceID: 0x24f0 Rev: 0x0
    1 Connected Ports:
        PortNum:   1 LID: xxxxxx GUID: 0x0011750101574071
            Neighbor:  Name: edge1
                NodeGUID: 0x00117501025131cb Type: SW PortNum:   4
            LocalPort: 1   PortState: Active                PhysState: LinkUp
            IsSMConfigurationStarted: True  NeighborNormal: True
            PortType: Standard
            LID:    xxxxxx              LMC: 0              Subnet:
0xfe80000000000000
            SMLID:  xxxxxx   SMSL:  0   RespTimeout:   32 us  SubnetTimeout:  536
ms
```



```
            M_KEY:  0x0000000000000000  Lease:     0 s          Protect: Read-only
            MTU  Supported: (0x6) 8192 bytes
            VLStallCount (per VL):  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
            MTU  Active by VL:
            00: 8192 01:    0 02:    0 03:    0 04:    0 05:    0 06:    0 07:
0
            08:    0 09:    0 10:    0 11:    0 12:    0 13:    0 14:    0 15:
2048
            16:    0 17:    0 18:    0 19:    0 20:    0 21:    0 22:    0 23:
0
            24:    0 25:    0 26:    0 27:    0 28:    0 29:    0 30:    0 31:
0
            LinkWidth: Active:       4  Supported:    1,2,3,4  Enabled:          4
            LinkWidthDnGrade: ActiveTx:  4 Rx:  4 Supported:    1,2,3,4
Enabled:        3,4
            PortLinkMode: Active: STL      Supported: STL              Enabled:
STL
            LinkSpeed: Active:    25Gb  Supported:       25Gb  Enabled:       25Gb
            SM_TrapQP: 0x0  SA_QP: 0x1 IPAddr Prim/Sec: :: / 0.0.0.0
            VLs:       Active:     8+1  Supported:        8+1
                HOQLife (Per VL):
                VL 0: 0x0 VL 1: 0x0 VL 2: 0x0 VL 3: 0x0 VL 4: 0x0
                VL 5: 0x0 VL 6: 0x0 VL 7: 0x0 VL 8: 0x0 VL 9: 0x0
                VL10: 0x0 VL11: 0x0 VL12: 0x0 VL13: 0x0 VL14: 0x0
                VL15: 0x0 VL16: 0x0 VL17: 0x0 VL18: 0x0 VL19: 0x0
                VL20: 0x0 VL21: 0x0 VL22: 0x0 VL23: 0x0 VL24: 0x0
                VL25: 0x0 VL26: 0x0 VL27: 0x0 VL28: 0x0 VL29: 0x0
                VL30: 0x0 VL31: 0x0

            VL Arb Cap:  High:      16        Low:         16  HiLimit:    0
PreemptLimit    0
            VLFlowControlDisabledMask: 0x00000000
            NeighborMode  MgmtAllowed: Yes  FwAuthenBypass:  On
NeighborNodeType: Switch
            Capability 0x00410020: CN CM APM
            Capability3 0x0008: SS
            Violations: M_Key: xxxxx P_Key: xxxxx Q_Key: xxxxx
            PortMode ActiveOptimize: Off PassThrough: Off VLMarker: Off
            FlitCtrlInterleave Distance Max:  1 Enabled:  1
            MaxNestLevelTxEnabled: 0 MaxNestLevelRxSupported: 0
            SmallPktLimit: 0x00 MaxSmallPktLimit: 0x00 PreemptionLimit: 0x00
            FlitCtrlPreemption MinInital: 0x0000 MinTail: 0x0000 LargePktLim: 0x00
            BufferUnits: VL15Init 0x0110; VL15CreditRate 0x00; CreditAck 0x0;
BufferAlloc 0x3
            PortErrorActions: 0x172000: CE-UVLMCE-BCDCE-BTDCE-BHDR-BVLM
            ReplayDepth Buffer 0x80; Wire 0x0c
            DiagCode: 0x0000
            OverallBufferSpace: 0x093f
            P_Key Enforcement: In: Off Out: Off
            QSFP Interpreted CableInfo:
                Identifier: 0xd
                ExtIdentifier: Power Class 1, 1.5W max
                Connector: 0x23
                NominalBR: 25 Gb
                OM2Length: 0m
                OM3Length: 0m
                OM4Length: 1m
                DeviceTech: Passive copper cable
                VendorName: FCI Electronics
                VendorOUI: 0xfc7ce7
                VendorPN: 10131941-2010LF
                VendorRev: 2
                MaxCaseTemp: 0 C
                CC_BASE: 0xe0
                TxCDR: N/A
                TxInpEqFixProg: False
                TxInpEqAutoAdp: False
                TxSquelchImp: False
                RxCDR: N/A
                RxOutpEmphFixProg: False
```

```
                RxOutpAmplFixProg: False
                MemPage02Provided: False
                MemPage01Provided: False
                VendorSN: CN1449FA102L0163
                DateCode: 2014/12/06-
                CC_EXT: 0x68
                CertCableFlag: N
                ReachClass: 4
                CertDataRates: 4x25G
1 Matching FIs Found

4 Connected Switches in Fabric:
0 Matching Switches Found

1 Connected SMs in Fabric:
0 Matching SMs Found
-------------------------------------------------------------------------------
```

3.3.12.11 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 `opareport` can parse.

Note: Intel recommends that you do **not** develop your own tools against this format because it may change in future versions of `opareport`.

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.

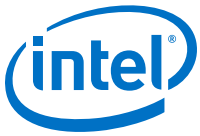
When a snapshot is generated, no additional `-o` options are allowed during the run and certain `opareport` options are ignored. These include: `-F`, `-P`, `-H`, and `-N`. However, the following options are valid:

- `-s` includes port counters in the snapshot.
- `-r` includes switch routing tables in the snapshot.
- `-V` includes QoS VL-related tables in the snapshot.
- `-i`, `-L`, `-a`, and `-C` control the port counters.

Notes: Quarantined nodes cannot be obtained from a snapshot.

- `opareport -o quarantinednodes -X snapshot` does not give the quarantined nodes on a snapshot of the same fabric.
- Use `opareport -o quarantinednodes` to return the quarantined nodes on a fabric with quarantined nodes.

After a snapshot has been generated, it may then be used as input to generate many types of `opareport` 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 `opareport` does not need to be attached to the fabric. Because this is a static report, certain options are not available, including `-i`, `-a`, `-C`, `-h HFI`, and `-p port`.



The report generated from the snapshot includes port counters **only** if the original snapshot was run with the `-s` option. If not, reports such as `-o` errors are not permitted against the snapshot.

Similarly, certain reports are permitted **only** if the original snapshot was run with the `-r` option. This includes: `-o linear`, `-o mcast`,`-o portusage`, `-o pathusage`, `-o treepathusage`, and `-o route`.

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.

Notes: Limitations of `-o route`:

- The Path Records reported may not be complete. The report shows the minimum valid value or an invalid value because certain fields such as `SLID`, `SL`, `PKey`, `MTU`, `Rate`, and `PktLifeTime` are not available. These values do not impact the actual route shown.
- Some routes reported may not be incomplete or not available to applications.

3.3.13 opaverifyhosts

Verifies basic node configuration and performance by running `FF_HOSTVERIFY_DIR/hostverify.sh` on all specified hosts.

Note: Prior to using `opaverifyhosts`, copy the sample file `/usr/lib/opa/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/sysconfig/opa/opafastfabric.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

```
opaverifyhosts [-kc] [-f hostfile] [-u upload_file] [-d upload_dir]
[-h hosts] [-T timelimit] [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/sysconfig/opa/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).
test	Specifies one or more specific tests to run. See `/usr/lib/opa/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

```
opaverifyhosts -c
opaverifyhosts -h 'arwen elrond'
HOSTS='arwen elrond' opaverifyhosts
```

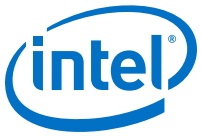
Environment Variables

`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.

3.3.14 opaxlattopology

Generates a topology XML file of a cluster using `topology.csv`, `linksum_swd06.csv`, and `linksum_swd24.csv` as input. The topology file can be used to bring up and verify the cluster.

Note: The `topology.csv` input file must be present in the same directory from which the script operates, but the `linksum` CSV files are read from the `/usr/lib/opa/samples` directory.



For more information, see Sample Files on page 215 and topology.xlsx Overview on page 219.

Syntax

```
opaxlattopology [-d level -v level -i level -K] [-s hfi_suffix]
[source [dest]]
```

Options

`--help` Produces full help text.

`-d level` Specifies the output detail level. Default = 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` Intel® Omni-Path Edge Switch 100 Series topology files.

`2` Rack topology files.

`4` Rack group topology files.

`-v level` Specifies the verbose level. Range = 0 - 8. Default = 2.

`0` No output.

`1` Progress output.

`2` Reserved.

`4` Time stamps.

`8` Reserved.

`-i level` Specifies the output indent level. Range = 0 - 15. Default = 0.

`-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, HFIs, and switches used to create a topology XML file. Temporary files are not typically needed after a topology file is created,

however they are used for creating `linksum_swd06.csv` and `linksum_swd24.csv` files, or can be retained for subsequent inspection or processing.

`-s hfi_suffix` Used on Multi-Rail or Multi-Plane fabrics. Can be used to override the default hfi1_0.

For Multi-Plane fabric, use the tool multiple times with a different hfi-suffix.

For Multi-Rail fabric, specify HostName as "HostName HfiName" in the spreadsheet.

Description

The `opaxlattopology` script reads the `topology.csv` file from the local directory, and reads the other files from `/usr/lib/opa/samples/linksum_swd06.csv` and `/usr/lib/opa/samples/linksum_swd24.csv`. The `topology.csv` file is created from the `topology.xlsx` spreadsheet by saving the Fabric Links tab as a .CSV file to `topology.csv`. A sample `topology.xlsx` is located in the `/usr/lib/opa/samples/` directory. Inspect the `topology.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.

The script outputs one or more topology files starting with `topology.0:0.xml`. The `topology.csv` input file must be present in the same directory from which the script operates, but the `linksum` CSV files are read from the `/usr/lib/opa/samples` directory.

Example

```
opaxlattopology
# reads default input 'topology.csv' and creates default
# output 'topology.0:0.xml'

opaxlattopology  fabric_2.csv
# reads input 'fabric_2.csv' and creates default output
```

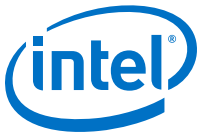
See `topology.xlsx` for examples of links between HFI and Edge SW (rows 4-7), HFI and Core SW (rows 8-11), and Edge SW and Core SW (rows 12-15).

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 = 10240

`MTU_SW_HFI` If set, it overrides default MTU on switch-to-HFI links. Default = 8192



Creating linksum Files

The `linksum_swd06.csv` and `linksum_swd24.csv` files are provided as stand-alone files in the `/usr/lib/opa/samples` directory. However, they can be recreated (or modified) from the spreadsheet, if needed, by performing the following steps:

1. Save each of the following from the `topology.xlsx` file as individual `.csv` files:
   - Internal SWD06 Links tab as `linksum_swd06.csv`
   - Internal SWD24 Links tab as `linksum_swd24.csv`
   - Fabric Links tab as `topology.csv`
2. For each saved `topology.csv` file, run the script with the `-K` option.
3. Upon completion of the script, save the top level `linksum.csv` file as `linksum_swd06.csv` or `linksum_swd24.csv` as appropriate.

Including SM Blocks in Topology

You must manually edit the `/etc/sysconfig/opa/topology.0:0.xml` file to include SM blocks for each SM in your fabric. `opaxlattopology` translates the `topology.csv` file which does not include SM blocks in the xml file by default. If manual edits are not completed, then the command `opareport -o verifyall -T topology.0:0.xml` fails with errors similar to the following:

```
# opareport -o verifyall -T topology.0:0.xml
SMs Topology Verification

SMs Found with incorrect configuration:
NodeGUID          Port PortGUID         Type Name
0x0011750101671ed9   1 0x0011750101671ed9 FI phkp3un86 hfi1_0
NodeDetails: koplab
Unexpected SM

1 of 1 Fabric SMs Checked

SMs Expected but Missing or Duplicate in input:
0 of 0 Input SMs Checked

Total of 1 Incorrect SMs found
0 Missing, 1 Unexpected, 0 Duplicate, 0 Different
```

To retrieve the info for the SM block, use the command `opareport -o nodes -F sm` as shown in the following example.

```
# opareport -o nodes -F sm
Getting All Node Records...
Done Getting All Node Records
Done Getting All Link Records
Done Getting All Cable Info Records
Done Getting All SM Info Records
Node Type Summary
Focused on:
    Port:   1 0x0011750101671ed9
        in Node: 0x0011750101671ed9 FI phkp3un86 hfi1_0

3 Connected FIs in Fabric:
    Name: phkp3un86 hfi1_0
        NodeGUID: 0x0011750101671ed9 Type: FI
        Ports: 1 PartitionCap: 16 SystemImageGuid: 0x0011750101671ed9
        BaseVer: 128 SmaVer: 128 VendorID: 0x1175 DeviceID: 0x24f0 Rev: 0x10
```

```
    1 Connected Ports:
        PortNum:   1 LID: 0x0001 GUID: 0x0011750101671ed9
            Neighbor:  Name: phedfim20
                NodeGUID: 0x0011750102648205 Type: SW PortNum:  46
            Width:    4 Speed:    25Gb Downgraded?  No
1 Matching FIs Found

1 Connected Switches in Fabric:
0 Matching Switches Found

1 Connected SMs in Fabric:
    State: Master      Name: phkp3un86 hfi1_0
        NodeGUID: 0x0011750101671ed9 Type: FI
        PortNum:   1 LID: 0x0001 PortGUID: 0x0011750101671ed9
        SM_Key: 0x0000000000000000 Priority:   0 ActCount: 0x0000bcb2
1 Matching SMs Found
```

At a minimum, the following information must be added in the topology.0:0.xml to avoid the "Unexpected SM" warning. This must be added inside <Nodes> section:

```
<SMs>
<SM>
<NodeDesc>abcxyz</NodeDesc>
</SM>
</SMs>
```

3.3.15 opaxlattopology_cust

Customizable script for documenting cluster topology. Provides an alternative to the standard script (see opaxlattopology on page 102). Edit the sample `topology_cust.xlsx` to represent each external link in a cluster, then modify `opaxlattopology_cust` to translate the alternate CSV form to the standard CSV form used by `opaxlattopology`.

Syntax

```
opaxlattopology_cust [-t topology_prime] [-s topology_second] [-T topology_out]
[-v level] [-i level] [-K]
```

Options

`--help` — Produces full help text.

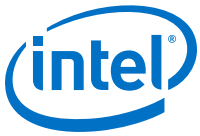
`-t topology_prime` — Specifies the primary topology CSV input file. Specifies the primary CSV input file and must be present.

`-s topology_second` — Specifies the secondary topology CSV input file. Specifies a secondary CSV input file. Appended to the primary for processing.

`-T topology_out` — Specifies the topology CSV output file. Specifies the CSV output file name and must be specified.

`-v level` — Specifies the verbose level. Range = 0 - 8, default = 2.

`0` No output.



	1 Progress output. 2 Reserved. 4 Time stamps. 8 Reserved.
`-i` *level*	Specifies the screen output indent level. Range = 0 - 15, default = 0.
`-K`	Specifies DO NOT clean temporary files. Prevents temporary files from being removed. Temporary files contain CSV data used during processing. Temporary files are not needed after the standard-format CSV file is created, but they can be retained for subsequent inspection or processing.

Description

Each link contains source, destination, and cable fields with one link per row of the spreadsheet. Link fields must not contain commas. Source and Destination fields are each a concatenation of name and port information in the following forms. Names not of the form `ib` or `C` are assumed to be host names.

The following lists the `node type` and source/destination.

`Host`: host*N* where N is a host number.

`Edge Switch`: ib*N*p*N* where N is a switch/port number.

`Core Leaf`: C*n* L*nnn* p*N* where N/n is a host/switch/port number.

Cable values, CableLength, and CableDetails are optional and have no special syntax. If present, they are placed in the standard-format CSV file exactly as they appear. CableLabel is created automatically by `opaxlattopology_cust` as the concatenation of Source and Destination.

Rack Group and Rack are not supported in `topology_cust.xlsx`. Therefore, `opaxlattopology_cust` leaves these fields empty in the standard-format CSV file.

3.4 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.

3.4.1 opaextracterror

Produces a CSV file listing all or some of the errors in the current fabric. `opaextracterror` is a front end to the `opareport` tool. The output from this tool can be imported into a spreadsheet or parsed by other scripts.

Syntax

```
opaextracterror [opareport options]
```

Options

`opareport` *`options`*	Options are passed to `opareport`. See opareport on page 63 for the full set of options.

Examples

```
# List all the link errors in the fabric:
opaextracterror

# List all the link errors related to a switch named "OmniPth00117501ffffffff":
opaextracterror -F "node:OmniPth00117501ffffffff"

# List all the link errors for end-nodes:
opaextracterror -F "nodetype:FI"

# List all the link errors on the 2nd HFI's fabric of a multi-plane fabric:
opaextracterror -h 2
```

3.4.2 opaextractlids

Produces a CSV file listing all or some of the LIDs in the fabric. `opaextractlids` is a front end to the `opareport` tool. The output from this tool can be imported into a spreadsheet or parsed by other scripts.

Syntax

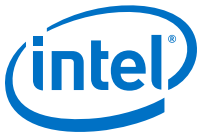
```
opaextractlids [opareport options]
```

Options

`opareport` *`options`*	Options are passed to `opareport`. See opareport on page 63 for the full set of options.

Examples

```
# List all the lids in the fabric:
opaextractlids

# List all the lids of end-nodes:
opaextractlids -F "nodetype:FI"

# List all the lids on the 2nd HFI's fabric of a multi-plane fabric:
opaextractlids -h 2
```



3.4.3 opaextractperf

Provides a report of all performance counters in a CVS format suitable for importing into a spreadsheet or parsed by other scripts for further analysis. It generates a detailed `opareport` component summary report and pipes the result to `opaxmlextract`, extracting element values for NodeDesc, SystemImageGUID, 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 and SMs portions are suppressed).

Syntax

```
opaextractperf [opareport options]
```

Options

`opareport` *`options`*	Options are passed to `opareport`. See opareport on page 63 for the full set of options.

The portion of the script that calls `opareport` and `opaxmlextract` follows:

```
opareport -o comps -s -x -d 10 $@ | opaxmlextract -d \;
-e NodeDesc -e SystemImageGUID -e PortNum -e XmitDataMB
-e XmitData -e XmitPkts -e RcvDataMB -e RcvData -e RcvPkts
-e SymbolErrors -e LinkErrorRecovery -e LinkDowned -e PortRcvErrors
-e PortRcvRemotePhysicalErrors -e PortRcvSwitchRelayErrors
-e PortXmitDiscards -e PortXmitConstraintErrors
-e PortRcvConstraintErrors -e LocalLinkIntegrityErrors
-e ExcessiveBufferOverrunErrors -e VL15Dropped -s Neighbor -s SMs
```

Example .

```
opaextractperf
opaextractperf -h 1 -p 2
```

3.4.4 opaextractstat

Performs an error analysis of a fabric and provides augmented information from a `topology_file`. The report provides cable information as well as symbol error counts.

`opaextractstat` generates a detailed `opareport` errors report that also has a topology file (see opareport on page 63 for more information about topology files). The report is piped to `opaxmlextract` which extracts values for Link, Cable and Port. (The port element names are context-sensitive.) Note that `opaxmlextract` generates two extraction records for each link (one for each port on the link); therefore, `opaextractstat` merges the two records into a single record and removes redundant link and cable information.

`opaextractstat` 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

```
opaextractstat topology_file [opareport options]
```

Options

`topology_file`	Specifies topology_file to use.
`opareport options`	Options are passed to `opareport`. See opareport on page 63 for the full set of options.

The portion of the script that calls `opareport` and `opaxmlextract` follows:

```
opareport -x -d 10 -s -o errors -T $@ | opaxmlextract -d \;
-e Rate -e MTU -e LinkDetails -e CableLength -e CableLabel
-e CableDetails -e Port.NodeDesc -e Port.PortNum -e SymbolErrors.Value
```

Examples

```
opaextractstat topology_file
opaextractstat topology_file -c my_opamon.conf
```

3.4.5 opashowallports

(Switch and Host) Displays basic port state and statistics for all host nodes, chassis, or externally-managed switches.

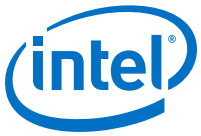
Note: `opareport` and `opareports` are more powerful Intel® Omni-Path Fabric Suite FastFabric commands. For general fabric analysis, use `opareport` or `opareports` with options such as `-o errors` and `-o slowlinks` to perform an efficient analysis of link speeds and errors.

Syntax

```
opashowallports [-C] [-f hostfile] [-F chassisfile] [-h 'hosts']
[-H 'chassis'] [-S]
```

Options

`--help`	Produces full help text.
`-C`	Performs operation against chassis. Default = host.
`-f hostfile`	Specifies the file containing the list of hosts in cluster. Default is `/etc/sysconfig/opa/hosts` file.
`-F chassisfile`	Specifies the file containing the list of chassis in cluster. Default is `/etc/sysconfig/opa/chassis` file.
`-h hosts`	Specifies the list of hosts for which to show ports.



`-H chassis`	Specifies the list of chassis for which to show ports.
`-S`	Securely prompts for password for admin on chassis.

Environment Variables

The following environment variables are also used by this command:

`HOSTS`	List of hosts, used if `-h` option not supplied. See discussion on Selection of Hosts.
`CHASSIS`	List of chassis, used if `-C` is used and `-H` and `-F` options not supplied. See discussion on Selection of Chassis.
`HOSTS_FILE`	File containing list of hosts, used in absence of `-f` and `-h`. See discussion on Selection of Hosts.
`CHASSIS_FILE`	File containing list of chassis, used in absence of `-F` and `-H`. See discussion on Selection of Chassis.
`FF_CHASSIS_LOGIN_METHOD`	How to log into chassis. Can be Telnet or SSH.
`FF_CHASSIS_ADMIN_PASSWORD`	Password for admin on all chassis. Used in absence of `-S` option.

Example

```
opashowallports
opashowallports -h 'elrond arwen'
HOSTS='elrond arwen' opashowallports
opashowallports -C
opashowallports -H 'chassis1 chassis2'
CHASSIS='chassis1 chasssi2' opashowallports
```

Notes

When performing `opashowallports` against hosts, internally SSH is used. The command `opashowallports` requires that password-less SSH be set up between the host running the Intel® Omni-Path Fabric Suite FastFabric Toolset and the hosts `opashowallports` is operating against. The `opasetupssh` FastFabric tool can aid in setting up password-less SSH.

When performing operations against chassis, Intel recommends that you set up SSH keys (see opasetupssh). If SSH keys are not set up, Intel recommends that you use the `-S` option, to avoid keeping the password in configuration files.

When performing `opashowallports` against externally-managed switches, a node with Intel® Omni-Path Fabric Suite FastFabric Toolset installed is required. Typically, this is the node from which `opashowallports` is being run.

3.5 Configuration and Control for Chassis, Switch, and Host

The CLIs described in this section are used for general management of hosts in the fabric, as well as internally- and externally-managed switches. There are also helper programs (for example, `opagenswitches`) to help produce the necessary configuration files.

3.5.1 opagenswitches

Analyzes the present fabric and produces a list of Externally Managed switches in the required format for the `/etc/sysconfig/opa/switches` file.

Syntax

```
opagenswitches [-t portsfile] [-p ports] [-R] [-L switches_file] [-o output_file]
[-T topology_file] [-X snapshot_file] [-s] [-v level] [-K]
```

Options

`--help` — Produces full help text.

`-t` *portsfile* — Specifies the file with list of local HFI ports used to access fabric(s) for analysis. Default is `/etc/sysconfig/opa/ports` file.

`-p` *ports* — Specifies the list of local HFI ports used to access fabrics for counter clear.

Default is first active port. The first HFI in the system is 1. The first port on an HFI is 1. Uses the format hfi:port, for example:

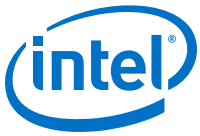
- `0:0` First active port in system.
- `0:y` Port *y* within system.
- `x:0` First active port on HFI *x*.
- `x:y` HFI *x*, port *y*.

`-R` — Does not attempt to get routes for computation of distance.

`-s` — Updates/resolves switch names using topology XML data.

`-L` *switches_file* — Specifies the name of a pre-existing *switches_file* to be used as input in conjunction with a topology file. When specified, the file is used instead of switches data obtained from the actual fabric. The updated switches data is output to stdout (common to all `opagenswitches` operations). Does not generate switches data. Must also use `-s` option.

`-o` *output_file* — Writes switches data to *output_file*. Default is stdout.



`-T topology_file` Specifies *topology_file* to use. May contain `'%P'`. Must also use `-s`.

Link data in the topology file is compared to actual fabric link data (obtained by `opareport -o links` or `opareport -X snapshot -o links`). The data is also matched to a list of switch node GUIDs and the switch NodeDesc values are generated. This list is then applied to the switches data to update NodeDesc values. The comparison of topology link data to actual fabric link data starts with the host names. The host names in the actual fabric must match those in the topology file for the comparison to succeed. However, the comparison logic allows for some mismatches, which could be due to swapped or missing cables. Switch NodeDesc values are matched to GUIDs based on which switch has the greater number of matching links.

`-X snapshot_file` Uses *snapshot_file* XML for fabric link information. May contain `'%P'`. Must also use `-s`.

`-v level` Specifies the verbose level. Default = 0. Values include:

`0` No output.

`1` Progress output.

`2` Reserved.

`4` Time stamps.

`8` Reserved.

`-K` Does not clean temporary files. Temporary files are CSV format and contain lists of links used during script operation. The files are not normally needed after execution, but they can be retained for subsequent inspection or processing.

Environment Variables

The following environment variables are also used by this command:

`PORTS` List of ports, used in absence of `-t` and `-p`.

`PORTS_FILE` File containing list of ports, used in absence of `-t` and `-p`.

`FF_TOPOLOGY_FILE` File containing topology XML data, used in absence of `-T`.

Examples

```
opagenswitches
opagenswitches -p '1:1 2:1'
opagenswitches -o switches
opagenswitches -s -o switches
opagenswitches -L switches -s -o switches
opagenswitches -s -T topology.%P.xml
opagenswitches -L switches -s -T topology.%P.xml -X snapshot.%P.xml
```

3.5.2 opagenchassis

Generates a list of IPv4, IPv6, and/or TCP names in a format acceptable for inclusion in the `/etc/sysconfig/opa/chassis` file.

Syntax

```
opagenchassis [-t portsfile] [-p ports]
```

Options

`--help` Produces full help text.

`-t` *portsfile* Specifies the file with list of local HFI ports used to access fabric for analysis. Default is `/etc/sysconfig/opa/ports` file.

`-p` *ports* Specifies the list of local HFI ports used to access fabrics for counter clear.

Default is first active port. The first HFI in the system is 1. The first port on an HFI is 1. Uses the format hfi:port, for example:

`0:0` First active port in system.

`0:y` Port *y* within system.

`x:0` First active port on HFI *x*.

`x:y` HFI *x*, port *y*.

Environment Variables

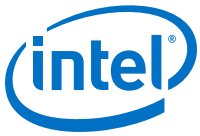
The following environment variables are also used by this command:

`PORTS` List of ports, used in absence of `-t` and `-p`.

`PORTS_FILE` File containing list of ports, used in absence of `-t` and `-p`.

Examples

```
opagenchassis
opagenchassis -p '1:1 1:2 2:1 2:2'
```



3.5.3 opagenesmchassis

Generates a list of chassis IPv4 and IPv6 addresses and/or TCP names where the Embedded Subnet Manager (ESM) is running, in a format acceptable for inclusion in the `/etc/sysconfig/opa/esm_chassis` file. This tool uses `opagenchassis` output to iterate through all the chassis.

Syntax

```
opagenesmchassis [-u user] [-S] [-t portsfile] [-p ports]
```

Options

`--help` Produces full help text.

`-u` *user* Performs command as `user`. For chassis, the default is `admin`.

`-S` Securely prompts for password for user on chassis.

`-t` *portsfile* Specifies the file with a list of local HFI ports used to access fabric(s) for analysis. Default is `/etc/sysconfig/opa/ports`

`-p` *ports* Specifies the list of local HFI ports used to access fabrics.

Default is first active port. The first HFI in the system is 1. The first port on an HFI is 1. Uses the format hfi:port, for example:

`0:0` First active port in system.

`0:y` Port *y* within system.

`x:0` First active port on HFI *x*.

`x:y` HFI *x*, port *y*.

Environment Variables

The following environment variables are also used by this command:

`FF_CHASSIS_ADMIN_PASSWORD` Password for chassis, used in absence of `-S`.

`PORTS` List of ports, used in absence of `-t` and `-p`.

`PORTS_FILE` File containing list of ports, used in absence of `-t` and `-p`.

Examples

```
opagenesmchassis
opagenesmchassis -S -p '1:1 1:2 2:1 2:2'
```

Alternatively, while editing the file, use a `vi` command to include the output such as:

```
:r! opagenesmchassis
```

3.5.4 opachassisadmin

(Switch) Performs a number of multi-step chassis initialization and verification operations, including initial chassis setup, firmware upgrades, chassis reboot, and others.

Syntax

```
opachassisadmin [-c] [-F chassisfile] [-H 'chassis'] [-P packages]
[-a action] [-I fm_bootstate] [-S] [-d upload_dir] [-s securityfiles]
operation ...
```

Options

`--help` — Produces full help text.

`-c` — Overwrites the result files from any previous run before starting this run.

`-F chassisfile` — Specifies the file with chassis in cluster. The default is `/etc/sysconfig/opa/chassis`.

`-H chassis` — Specifies the list of chassis to execute the operation against.

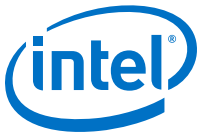
`-P packages` — Specifies the filenames and directories of firmware images to install.

- For directories specified, all `.pkg`, `.dpkg`, and `.spkg` files in directory tree are used. `shell` wild cards may also be used within quotes.
- For `fmconfig`, filename of FM config file is used.
- For `fmgetconfig`, filename to upload to (default `opafm.xml`) is used.

`-a action` — Specifies the action for the supplied file. The default is push.

For chassis upgrade:

`push`	Ensures firmware is in primary or alternate.
`select`	Ensures firmware is in primary.
`run`	Ensures firmware is in primary and running.



For chassis `fmconfig`:	`push`	Ensures the configuration file is in chassis.
	`run`	After push, restarts FM on master, stops on secondary.
	`runall`	After push, restarts FM on all management modules.
For chassis `fmcontrol`:	`stop`	Stops FM on all management modules.
	`run`	Ensures FM running on master, stopped on secondary.
	`runall`	Ensures FM running on all management modules.
	`restart`	Restarts FM on master, stops on secondary.
	`restartall`	Restarts FM on all MM.
For chassis `fmsecurityfiles`:	`push`	Ensures FM security files are in chassis.
	`restart`	After push, restarts FM on master, stop on slave.
	`restartall`	After push, restarts FM on all MM

`-I` *`fm_bootstate`*

Specifies the `fmconfig` and `fmcontrol` install options.

`disable`	Disables FM start at chassis boot.
`enable`	Enables FM start on master at chassis boot.
`enableall`	Enables FM start on all MM at chassis boot.

`-d` *`upload_dir`*

Specifies the directory to upload FM configuration files to; default is `uploads`.

`-S`

Securely prompts for password for user on chassis.

-s *securityFiles*	Specifies the security files to install. Default is *.pem. For Chassis *fmsecurityfiles*, filenames/directories of security files to install. For directories specified, all security files in directory tree are used. Shell wildcards may also be used within quotes. For Chassis *fmgetsecurityfiles*, filename to upload to. Default is *.pem
operation	Specifies the operation to perform. Can be one or more of:

reboot	Reboots chassis, ensures they go down and come back.
configure	Runs wizard to perform chassis configuration.
upgrade	Upgrades install of all chassis.
getconfig	Gets basic configuration of chassis.
fmconfig	FM configuration operation on all chassis.
fmgetconfig	Fetches FM configuration from all chassis.
fmcontrol	Controls FM on all chassis.
fmsecurityfiles	FM security files operation on all chassis.
fmgetsecurityfiles	Fetches FM security files from all chassis.

For more information on the operations that can be performed, see Operation Details on page 119.

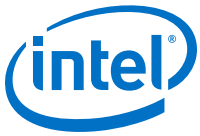
Example

```
opachassisadmin -c reboot
opachassisadmin -P /root/ChassisFw4.2.0.0.1 upgrade
opachassisadmin -H 'chassis1 chassis2' reboot
CHASSIS='chassis1 chassis2' opachassis_admin reboot
opachassisadmin -a run -P '*.pkg' upgrade
```

Environment Variables

The following environment variables are also used by this command:

CHASSIS	List of chassis, used if -H and -F option not supplied. Refer to Selection of Chassis on page 18 for more information.



CHASSIS_FILE	File containing list of chassis, used in absence of -F and -H. Refer to Selection of Chassis on page 18 for more information.
FF_MAX_PARALLEL	Maximum concurrent operations.
FF_SERIALIZE_OUTPUT	Serializes output of parallel operations (yes or no).
UPLOADS_DIR	Directory to upload to, used in absence of -d.

Operation Details

(Switch) All chassis operations log into the chassis as chassis user admin. Intel recommends using the -S option to securely prompt for a password, in which case the same password is used for all chassis. Alternately, the password may be put in the environment or the opafastfabric.conf file using FF_CHASSIS_ADMIN_PASSWORD.

All versions of Intel® Omni-Path Switch 100 Series firmware permit SSH keys to be configured within the chassis for secure password-less login. In this case, there is no need to configure a FF_CHASSIS_ADMIN_PASSWORD and FF_CHASSIS_LOGIN_METHOD can be SSH. Refer to the *Intel® Omni-Path Fabric Switches Command Line Interface Reference Guide* for more information.

upgrade	Upgrades the firmware on each chassis or slot specified. The -P option selects a directory containing .pkg files or provides an explicit list of .pkg files for the chassis and/or slots. The -a option selects the desired minimal state for the new firmware. For each chassis and/or slot selected for upgrade, the .pkg file applicable to that slot is selected and used. If more than one .pkg file is specified of a given card type, the operation is undefined. The upgrade is intelligent and does not upgrade chassis that already have the desired firmware in the desired state (as specified by -a). When the -a option specifies run, chassis that are not already running the desired firmware are rebooted. By selecting the proper FF_MAX_PARALLEL value, a rolling upgrade or a parallel upgrade may be accomplished. In most cases, a parallel upgrade is recommended for expediency. For more information about chassis firmware, refer to the *Intel® Omni-Path Fabric Switches GUI User Guide* and *Intel® Omni-Path Fabric Externally-Managed Switches Release Notes*.
configure	Runs the chassis setup wizard, which asks a series of questions. Once the wizard has finished prompting for configuration information, all the selected chassis are

configured through the CLI interface according to the responses. The following options may be configured for all chassis:

- Syslog server IP address, TCP/UDP port number, syslog facility code, and the chassis LogMode.
- NTP server
- Local time zone
- Link CRC Mode
- Link width supported
- Node description

`reboot` — Reboots the given chassis and ensures they go down and come back up by pinging them during the reboot process.

By selecting the proper `FF_MAX_PARALLEL` value, a rolling reboot or a parallel reboot may be accomplished. In most cases, a parallel upgrade is recommended for expediency.

`getconfig` — Retrieves basic information from a chassis such as syslog, NTP configuration, timezone info, Link CRC Mode, Link Width, and node description.

`fmconfig` — Updates the Fabric Manager configuration file on each chassis specified. The `-P` option selects a file to transfer to the chassis. The `-a` option selects the desired minimal state for the new configuration and controls whether the FM is started/restarted after the file is updated. The `-I` option can be used to configure the FM start at boot for the selected chassis.

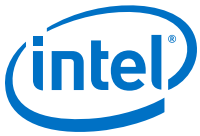
`fmgetconfig` — Uploads the FM configuration file from all selected chassis. The file is uploaded to the selected uploads directory. The `-P` option specifies the desired destination filename within the uploads directory.

`fmcontrol` — Allows the FM to be controlled on each chassis specified. The `-a` option selects the desired state for the FM.

The `-I` option configures the FM start at boot for the selected chassis.

`fmsecurityfiles` — Updates the FM security files on each chassis specified. The `-s` option selects file(s) to transfer to the chassis. The `-a` option selects the desired minimal state for the new security files. In this release, `push` is the only supported action.

`fmgetsecurityfiles` — Uploads the FM security files from all selected chassis. The files are uploaded to the selected uploads directory. The `-s` option specifies the desired destination filename within the uploads directory.



Logging

`opachassisadmin` provides detailed logging of its results. During each run, the following files are produced:

`test.res` This file is appended with summary results of run.

`test.log` This file is appended with detailed results of run.

`save_tmp/` This file is contains a directory per failed test with detailed logs.

`test_tmp*/` This file contains the intermediate results while the test is running.

The `-c` option removes all log files.

ssh Keys

When performing operations against chassis, Intel recommends setting up SSH keys. If SSH keys are not set up, all chassis must be configured with the same admin password. In this case, Intel recommends using the `-S` option. The `-S` option avoids the need to keep the password in configuration files.

Results

Results from `opachassisadmin` are grouped into test suites, test cases, and test items. A given run of `opachassisadmin` represents a single test suite. Within a test suite, multiple test cases occur; typically one test case per chassis 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 `opachassisadmin` run appends to `test.res` and `test.log`, and creates temporary files in `test_tmp$PID` in the current directory. The `test.res` file provides an overall summary of operations performed and their results. The same information is also displayed while `opachassisadmin` 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 `opachassisadmin` are appended to `test.log`. Intel recommends reviewing failures and using the `-c` option to remove old logs before subsequent runs of `opachassisadmin`.

`opachassisadmin` implicitly performs its operations in parallel. However, as for the other tools, `FF_MAX_PARALLEL` can be exported to change the degree of parallelism. Twenty (20) parallel operations is the default.

3.5.5 opaswitchadmin

(Switch) Performs a number of multi-step initialization and verification operations against one or more externally managed Intel® Omni-Path switches. The operations include initial switch setup, firmware upgrades, chassis reboot, and others.

Syntax

```
opaswitchadmin [-c] [-N 'nodes'] [-L nodefile] [-O]
 [-P packages] [-a action] [-t portsfile]
[-p ports] operation ...
```

Options

`-help` Produces full help text.

`-c` Overwrites result files from any previous run before starting this run.

`-N` *nodes* Specifies the list of nodes to execute the operation against.

`-L` *nodefile* Specifies the file with nodes in the cluster. Default is `/etc/sysconfig/opa/switches` file.

`-P` *packages* For upgrades: Specifies the file name or directory where the firmware image is to install. For the directory specified, `.emfw` file in the directory tree is used. `shell` wild cards may also be used within quotes.

`-t` *portsfile* Specifies the file with list of local HFI ports used to access fabrics for switch access. Default is `/etc/sysconfig/opa/ports` file.

`-p` *ports* Specifies the list of local HFI ports used to access fabrics for switch access.

Default is first active port. The first HFI in the system is 1. The first port on an HFI is 1. Uses the format hfi:port, for example:

`0:0` First active port in system.

`0:y` Port *y* within system.

`x:0` First active port on HFI *x*.

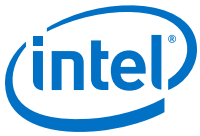
`x:y` HFI *x*, port *y*.

`-a` *action* Specifies an action for firmware file for switch upgrade. The `action` argument can be one or more of the following:

`select` Ensures firmware is in primary (default).

`run` Ensures firmware is in primary and running.

`-O` Specifies the override for firmware upgrades, bypasses the previous firmware version checks, and forces the update unconditionally.



operation	Performs the specified *operation*, which can be one or more of the following:	
	`reboot`	Reboots switches, ensures they go down and come back.
	`configure`	Runs wizard to set up switch configuration.
	`upgrade`	Upgrades installation of all switches.
	`info`	Reports firmware and hardware version, part number, and data rate capability of all nodes.
	`hwvpd`	Completes hardware Vital Product Data (VPD) report of all nodes.
	`ping`	Pings all nodes and tests for presence.
	`fwverify`	Reports integrity of failsafe firmware of all nodes.
	`getconfig`	Gets port configurations of an externally managed switch.

For more information on operations, see Operation Details on page 124.

Example

```
opaswitchadmin -c reboot
opaswitchadmin -P /root/ChassisFwX.X.X.X.X upgrade
opaswitchadmin -a run -P '*.emfw' upgrade
```

Environment Variables

The following environment variables are also used by this command:

`OPASWITCHES`	List of nodes, used in absence of `-N` and `-L` options. See discussion in Selection of Switches on page 20.
`OPASWITCHES_FILE`	File containing list of nodes, used in absence of `-N` and `-L` options. See discussion in Selection of Switches on page 20.
`FF_MAX_PARALLEL`	Maximum concurrent operations.
`FF_SERIALIZE_OUTPUT`	Serialize output of parallel operations (yes or no).

Details

`opaswitchadmin` 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 opaswitchadmin are grouped into test suites, test cases, and test items. A given run of opaswitchadmin represents a single test suite. Within a test suite, multiple test cases occur; typically one test case per chassis 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.

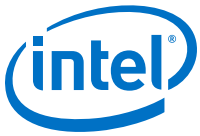
Each opaswitchadmin run appends to test.res and test.log and creates temporary files in test_tmp$PID in the current directory. the test.res file provides an overall summary of operations performed and their results. The same information is also displayed while opaswitchadmin is executing. test.log contains 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 opaswitchadmin are appended to test.log. Intel recommends reviewing failures and using the -c option to remove old logs before subsequent runs of opaswitchadmin. opaswitchadmin also appends to punchlist.csv for failing switches.

opaswitchadmin implicitly performs its operations in parallel. However, as for the other tools, FF_MAX_PARALLEL can be exported to change the degree of parallelism. Twenty (20) parallel operations is the default.

Operation Details

(Switch) All operations against Intel® Omni-Path Fabric externally-managed switches (except ping) securely access the selected switches. If a password has been set, the -S option must be used to securely prompt for a password. In this case, the same password is used for all switches.

reboot	Reboots the given switches. Use the FF_MAX_PARALLEL value to select either a rolling reboot or a parallel reboot. In most cases, a parallel reboot is recommended for expediency.
upgrade	Upgrades the firmware on each specified switch. The -P option selects a directory containing a .emfw file or provides an explicit .emfw file for the switches. If more than one .emfw file is specified, the operation is undefined. The -a option selects the desired minimal state for the new firmware. Only the select and run options are valid for this operation.



When the `-a` option specifies `run`, switches are rebooted. Use the `FF_MAX_PARALLEL` value to select a rolling upgrade or a parallel upgrade. In most cases, a parallel upgrade is recommended for expediency.

The upgrade process also sets the switch name. See discussion on Selection of Devices on page 17.

The upgrade process is used to set, clear, or change the password of the switches using the `-s` option. When this option is specified, you are prompted for a new password to be set on the switches. To reset (clear) the password, press **Enter** when prompted. This option can be used to configure the switches to not require a password for subsequent operations. A change to the password does not take effect until the next reboot of the switch.

For more information about switch firmware, refer to the *Intel® Omni-Path Fabric Switches GUI User Guide* and *Intel® Omni-Path Fabric Externally-Managed Switches Release Notes*.

`configure` Runs the switch setup wizard, which asks a series of questions. Once the wizard has finished prompting for configuration information, all the selected switches are configured according to the entered responses. The following items are configurable for all Intel® Omni-Path Switch 100 Series:

- FM Enabled
- Link CRC Mode
- Link Width Supported
- OPA Node Description

Note: If 4X capability is not enabled in the user selection, 4X capability is added to port 1 for each switch being configured. This provides a *rescue* capability for the switch using FastFabric, in case the link is unable to connect to a link width other than 4X.

Note: Typically, the Node Description is updated automatically during a firmware upgrade, if it is configured properly in the `switches` file. Updating the node description is also available using the configure option without performing a firmware upgrade.

`info` Queries the switches and displays the following information:

- Firmware version
- Hardware version
- Hardware part number, including revision information
- Speed capability
- Fan status
- Power supply status

This operation also outputs a summary of various configuration settings for each switch within a fabric.

For example, in a fabric with seven switches, a report similar to the following is displayed.

```
Summary:
        count - info
        7 - Capability:QDR
        7 - Fan 1 status:Normal/Normal
        7 - Fan 2 status:Normal/Normal
        6 - F/W ver:6.0.2.0.28
        1 - F/W ver:6.1.0.0.72
        7 - H/W pt num:220058-004-E
        7 - H/W ver:004-E
        7 - PS1 Status:N/A
        7 - PS2 Status:ENGAGED
```

`hwvpd` Queries the switches and displays the Vital Product Data (VPD) including:

- Serial number
- Part number
- Model name
- Hardware version
- Manufacturer
- Product description
- Manufacturer ID
- Manufacture date
- Manufacture time

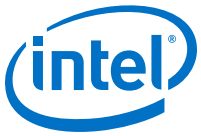
`ping` Issues an inband packet to the switches to test for presence and reports on presence/non-presence of each selected switch.

Note: It is not necessary to supply a password (using `-S`) for this operation.

`fwverify` Verifies the integrity of the firmware images in the EEPROMs of the selected switches.

`getconfig` Gets port configurations of an externally managed switch. This operation also outputs a summary of various configuration settings for each switch within a fabric. For example, in a fabric with seven switches, a report similar to the following is displayed.

```
Summary:
        count - configuration
        7 - Link Speed :  2.5-10Gb
        1 - Link Width :  1-8x
        6 - Link Width :  4x
```



This summary helps determine if all switches have the same configuration, and if not, indicates how many have each value. If some of the values are not as expected, view the `test.res` file to identify which switches have the undesirable values.

3.5.6 opahostadmin

(Host) Performs a number of multi-step host initialization and verification operations, including upgrading software or firmware, rebooting hosts, and other operations. In general, operations performed by `opahostadmin` involve a login to one or more host systems.

Syntax

```
opahostadmin [-c] [-i ipoib_suffix] [-f hostfile] [-h 'hosts']
[-r release] [-I install_options] [-U upgrade_options] [-d dir]
[-T product] [-P packages] [-m netmask] [-S] operation ...
```

Options

`--help`	Produces full help text.
`-c`	Overwrites the result files from any previous run before starting this run.
`-i` *ipoib_suffix*	Specifies the suffix to apply to host names to create IPoIB host names. Default is `-opa`.
`-f` *hostfile*	Specifies the file with the names of hosts in a cluster. Default is `/etc/sysconfig/opa/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® Omni-Path 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 = `IntelOPA-Basic`. Other options include: `IntelOPA-Basic.`*<distro>*, `IntelOPA-IFS.`*<distro>* where *<distro>* is the distribution and CPU.

`-P` *packages*	Specifies the packages to install. Default = `oftools ipoib mpi`
`-m` *netmask*	Specifies the IPoIB netmask to use for `configipoib` operation.
`-S`	Securely prompts for user password on remote system.
operation	Performs the specified *operation*, which can be one or more of the following:

`load`	Starts initial installation of all hosts.
`upgrade`	Upgrades installation of all hosts.
`configipoib`	Creates `ifcfg-ib1` using host IP address from `/etc/hosts` file.
`reboot`	Reboots hosts, ensures they go down and come back.
`sacache`	Confirms `sacache` has all hosts in it.
`ipoibping`	Verifies this host can ping each host through IPoIB.
`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).

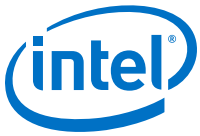
Example

```
opahostadmin -c reboot
opahostadmin upgrade
opahostadmin -h 'elrond arwen' reboot
HOSTS='elrond arwen' opahostadmin reboot
```

Details

`opahostadmin` 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 opahostadmin are grouped into test suites, test cases, and test items. A given run of opahostadmin 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 opahostadmin 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 opahostadmin 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 which 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 opahostadmin are appended to test.log. Intel recommends reviewing failures and using the -c option to remove old logs before subsequent runs of opahostadmin.

opahostadmin implicitly performs its operations in parallel. However, as for the other tools, FF_MAX_PARALLEL can be exported to change the degree of parallelism. Twenty (20) 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.
FF_MAX_PARALLEL	Maximum concurrent operations are performed.
FF_SERIALIZE_OUTPUT	Serialize output of parallel operations (yes or no).

opahostadmin 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 opafastfabric.conf file using FF_PASSWORD and FF_ROOTPASS.

load	Performs an initial installation of Intel® Omni-Path 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® Omni-Path Software configuration. The -I option can be used to select different install packages. Default = oftools ipoib 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,

IntelOPA-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, IntelOPA-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.

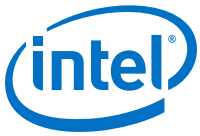
Note: Only components that are currently installed are upgraded. This operation fails for hosts that do not have Intel® Omni-Path Software installed.

configipoib

Creates a ifcfg-ib1 configuration file for each node using the IP address found using the resolver on the node. The standard Linux* resolver is used through the host command. (If running OFA Delta, this option configures ifcfg-ib0.)

If the host is not found, /etc/hosts on the node is checked. The -i option specifies an IPoIB suffix to apply to the host name to create the IPoIB host name for the node. The default suffix is -ib. The -m option specifies a netmask other than the default for the given class of IP address, such as when dividing a class A or B address into smaller IP subnets. IPoIB is configured for a static IP address and is autostarted at boot. For the Intel® OP Software Stack, the default /etc/sysconfig/ipoib.cfg file is used, which provides a redundant IPoIB configuration using both ports of the first HFI in the system.

Note: opahostadmin configipoib now supports DHCP (auto or static options) for configuring the IPoIB interface. You must specify these options in /etc/sysconfig/opa/opafastfabric.conf against the FF_IPOIB_CONFIG variable. If no options are found, the static IP configuration is used by default. If auto is specified, then one IP address from either static or dhcp is chosen. Static is used if the IP address can be obtained out of /etc/hosts or the resolver, otherwise DHCP is used.



`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.
`sacache`	Verifies the given hosts can properly communicate with the SA and any cached SA data that is up to date. To run this command, Intel® Omni-Path Fabric software must be installed and running on the given hosts. The subnet manager and switches must be up. If this test fails: `opacmdall 'opasaquery -o desc'` can be run against any problem hosts. *Note:* This operation requires that the hosts being queried are specified by a resolvable TCP/IP host name. This operation FAILS if the selected hosts are specified by IP address.
`ipoibping`	Verifies IPoIB basic operation by ensuring that the host can ping all other nodes through IPoIB. To run this command, Intel® Omni-Path Fabric software must be installed, IPoIB must be configured and running on the host, and the given hosts, the SM, and switches must be up. The `-i` option can specify an alternate IPoIB hostname suffix.
`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® Omni-Path Fabric 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 HFI model), or fabric issues (for example, symbol errors, incorrect link width, or speed). Assuming `opareport` 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.

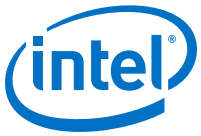
Note: This option is available for the Intel® Omni-Path Fabric Host Software OFA Delta packaging, but is not presently available for other packagings of OFED.

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 HFI model), or fabric issues (for example, symbol errors, incorrect link width, or speed). Assuming `opareport` 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 which 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 `opafastfabric.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 imit 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.

3.5.7 Interpreting the opahostadmin, opachassisadmin, and opaswitchadmin log files

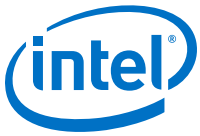
Each run of opahostadmin, opachassisadmin, and opaswitchadmin creates test.log and test.res files in the current directory.

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 chassis, switch, or 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/or chassis and the resulting output from that host and/or chassis before the FAILURE keyword.

If there is a FAILURE message indicating time-out, 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, chassis, or switch 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 or chassis with the expected password behavior.

One common source of time-out 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 time-outs is typographical errors in selected host or chassis names. Verify that the host, chassis, or switch names in the `test.log` file match the intended host names.

When IPoIB host names are used, verify that the correct name is formed based on the `opahostadmin -i '<IPOIB SUFFIX>'` argument. This argument applies a suffix to host names to create IPoIB host names. The default is `-ib`. Use `-i ''` to indicate no suffix.

3.6 Basic Setup and Administration Tools

The tools described in this section are available on a node that has Intel® Omni-Path Fabric Suite installed.

3.6.1 opapingall

(All) Pings a group of hosts or chassis to verify that they are powered on and accessible through TCP/IP ping.

Syntax

```
opapingall [-C] [-p] [-f hostfile] [-F chassisfile] [-h 'hosts'] [-H 'chassis']
```

Options

`--help`	Produces full help text.
`-C`	Performs a ping against a chassis. The default is hosts.
`-p`	Pings all hosts/chassis in parallel.
`-f hostfile`	Specifies the file with hosts in cluster. Default is `/etc/sysconfig/opa/hosts`.
`-F chassisfile`	Specifies the file with chassis in cluster. Default is `/etc/sysconfig/opa/chassis`.
`-h hosts`	Specifies the list of hosts to ping.
`-H chassis`	Specifies the list of chassis to ping.

Example

```
opapingall
opapingall -h 'arwen elrond'
HOSTS='arwen elrond' opapingall
opapingall -C
```

Note: This command pings all hosts/chassis found in the specified host/chassis file. The use of -C option merely 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 chassis.

```
opapingall -C -H 'chassis1 chassis2'
CHASSIS='chassis1 chassis2' opapingall -C
```

Environment Variables

HOSTS	List of hosts, used if -h option not supplied.
CHASSIS	List of chassis, used if -H option not supplied.
HOSTS_FILE	File containing list of hosts, used in absence of -f and -h.
CHASSIS_FILE	File containing list of chassis, used in absence of -F and -H.
FF_MAX_PARALLEL	When -p option is used, maximum concurrent operations are performed.

3.6.2 opasetupssh

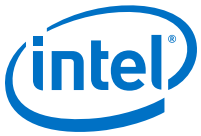
(Linux or Switch) Creates SSH keys and configures them on all hosts or chassis so the system can use SSH and SCP into all other hosts or chassis without a password prompt. Typically, during cluster setup this tool enables the root user on the Management Node to log into the other hosts (as root) or chassis (as admin) using password-less SSH.

Syntax

```
opasetupssh [-C|p|U] [-f hostfile] [-F chassisfile]
[-h 'hosts'] [-H 'chassis'] [-i ipoib_suffix]
[-u user] [-S] [-R|P]
```

Options

--help	Produces full help text.
-C	Performs operation against chassis. Default is hosts.
-p	Performs operation against all chassis or 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/sysconfig/opa/hosts file.
-F *chassisfile*	Specifies the file with chassis in cluster. Default is /etc/sysconfig/opa/chassis file.



`-h` *`hosts`*	Specifies the list of hosts to set up.
`-H` *`chassis`*	Specifies the list of chassis to set up.
`-i` *`ipoib_suffix`*	Specifies the suffix to apply to host names to create IPoIB host names. Default is `-opa`.
`-u` *`user`*	Specifies the user on remote system to allow this user to SSH to. Default is current user code for host(s) and admin for chassis.
`-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

Operations on Hosts

```
opasetupssh -S -i ''
opasetupssh -U
opasetupssh -h 'arwen elrond' -U
HOSTS='arwen elrond' opasetupssh -U
```

Operations on Chassis

```
opasetupssh -C
opasetupssh -C -H 'chassis1 chassis2'
CHASSIS='chassis1 chassis2' opasetupssh -C
```

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`. See discussion on Selection of Hosts.
`CHASSIS_FILE`	File containing list of chassis, used in absence of `-F` and `-H`. See discussion on Selection of Chassis.
`HOSTS`	List of hosts, used if `-h` option not supplied. See discussion on Selection of Hosts.
`CHASSIS`	List of chassis, used if `-C` is used and `-H` and `-F` options not supplied. See discussion on Selection of Chassis.
`FF_MAX_PARALLEL`	When `-p` option is used, maximum concurrent operations.

FF_IPOIB_SUFFIX	Suffix to append to hostname to create IPoIB hostname. Used in absence of -i.
FF_CHASSIS_LOGIN_METHOD	How to log into chassis. Can be Telnet or SSH.
FF_CHASSIS_ADMIN_PASSWORD	Password for admin on all chassis. Used in absence of -S option.

Description

The Intel® Omni-Path Fabric Suite FastFabric Toolset provides additional flexibility in the translation between IPoIB and management network hostnames. Refer to "Configuration of IPoIB Name Mapping" section in the *Intel® Omni-Path Fabric Suite FastFabric User Guide* for more information.

opasetupssh provides an easy way to create SSH keys and distribute them to the hosts or chassis 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, opasetupssh is an important setup step.

This tool also sets up SSH to the local host and the local host's IPoIB name. This capability is required by selected FastFabric Toolset commands and may be used by some applications (such as MPI).

opasetupssh has two modes of operation. The mode is selected by the presence or absence of the -U option. Typically, opasetupssh is first run without the -U option, then it may later be run with the -U option.

Host Initial Key Exchange

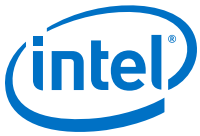
When run without the -U option, opasetupssh performs the initial key exchange and enables password-less SSH and SCP. The preferred way to use opasetupssh for initial key exchange is with the -S option. This requires that all hosts are configured with the same password for the specified "user" (typically 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).

opasetupssh configures password-less SSH/SCP for both the management network and IPoIB. Typically, the management network is used for FastFabric Toolset operations while IPoIB is used for MPI and other applications.

During initial cluster installation, where the Intel® Omni-Path Fabric software is not yet installed on all the hosts, IPoIB is not yet running. In this situation, use the -i option with an empty string as follows:

```
opasetupssh -i ''
```

This causes the last part of the setup of SSH for IPoIB to be skipped.



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 `opasetupssh` with the `-U` option. This option does not transfer the keys, but instead connects to each host (management network and IPoIB) 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, as such no files are transferred to the specified hosts and no passwords should be required.

Typically after completing the installation and booting of Intel® Omni-Path Fabric software, `opasetupssh` must be rerun with the `-U` option to update the `known_hosts` file.

Chassis Initial Key Exchange

When run without the `-U` option, `opasetupssh` performs the initial key exchange and enables password-less SSH and SCP. For chassis, the key exchange uses SCP and the chassis CLI. During this command you log into the chassis using the configured mechanism for chassis login.

The preferred way to use `opasetupssh` for initial key exchange is with the `-S` option. This requires that all chassis are configured with the same password for admin. In this mode, you are prompted for the password once and then the `FF_CHASSIS_LOGIN_METHOD` and SCP are used in conjunction with that password to complete the setup for the chassis. This method also avoids the need to setup the chassis password in `/etc/sysconfig/opa/opafastfabric.conf` (which can be a security risk).

For chassis, the `-i` option is ignored.

Chassis Refreshing Local Systems Known Hosts

If aspects of the chassis have changed, such as IP addresses or MAC addresses, you can refresh the local host's SSH `known_hosts` file by running `opasetupssh` with the `-U` option. This option does not transfer the keys, but instead connects to each chassis to refresh the SSH keys. Existing entries for the specified chassis 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 chassis, because no files are transferred to the specified hosts and no passwords are required.

3.6.3 opacmdall

(Linux and Switch) Executes a command on all hosts or Intel® Omni-Path Chassis. This powerful command can be used for configuring servers or chassis, verifying that they are running, starting and stopping host processes, and other tasks.

Note: `opacmdall` depends on the Linux* convention that utilities return 0 for success and >0 for failure. If `opacmdall` is used to execute a non-standard utility like `diff` or a program that uses custom exit codes, then `opacmdall` may erroneously report "Command execution FAILED" when it encounters a non-zero exit code. However, command output is still returned normally and the error may be safely ignored.

Syntax

```
opacmdall [-CpqPS] [-f hostfile] [-F chassisfile]
[-h hosts] [-H chassis] [-u user]
[-m marker] [-T timelimit] cmd
```

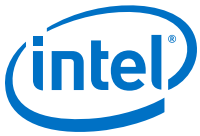
Options

`--help`	Produces full help text.
`-C`	Performs command against chassis. Default is hosts.
`-p`	Runs command in parallel on all hosts/chassis.
`-q`	Quiet mode, do not show command to execute.
`-P`	Outputs the hostname/chassis name as prefix to each output line. This can make script processing of output easier.
`-S`	Securely prompts for password for user on chassis.
`-f` *hostfile*	Specifies the file with hosts in cluster. Default is `/etc/sysconfig/opa/hosts` file.
`-F` *chassisfile*	Specifies the file with chassis in cluster. Default is `/etc/sysconfig/opa/chassis` file.
`-h` *host*	Specifies the list of hosts to execute command on.
`-H` *chassis*	Specifies the list of chassis to execute command on.
`-u` *user*	Specifies the user to perform the command as: • For hosts, the default is current user code. • For chassis, the default is `admin`.
`-m` *marker*	Specifies the marker for end of chassis command output. If omitted, defaults to chassis command prompt. This may be a regular expression.
`-T` *timelimit*	Specifies the time limit in seconds when running host commands. Default is -1 (infinite).

Examples

Operations on Host

```
opacmdall date
opacmdall 'uname -a'
opacmdall -h 'elrond arwen' date
HOSTS='elrond arwen' opacmdall date
```



Operations on Chassis

```
opacmdall -C 'ismPortStats -noprompt'
opacmdall -C -H 'chassis1 chassis2' ismPortStats -noprompt’
CHASSIS='chassis1 chassis2' opacmdall ismPortStats -noprompt’
```

Environment Variables

The following environment variables are also used by this command:

HOSTS	List of hosts, used if -h option not supplied. See discussion on Selection of Devices on page 17.
CHASSIS	List of chassis, used if -C is used and -H and -F options not supplied. See discussion on Selection of Devices on page 17.
HOSTS_FILE	File containing list of hosts, used in absence of -f and -h. See discussion on Selection of Devices on page 17.
CHASSIS_FILE	File containing list of chassis, used in absence of -F and -H. See discussion on Selection of Devices on page 17.
FF_MAX_PARALLEL	When -p option is used, maximum concurrent operations are performed.
FF_SERIALIZE_OUTPUT	Serialize output of parallel operations (yes or no).
FF_CHASSIS_LOGIN_METHOD	How to log into chassis. Can be Telnet or SSH.
FF_CHASSIS_ADMIN_PASSWORD	Password for admin on all chassis. Used in absence of -S option.

Notes

All commands performed with opacmdall must be non-interactive in nature. opacmdall 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.) Similarly, when running chassis commands such as fwUpdateChassis, the -reboot option should not be used because this option causes an immediate reboot and therefore the command never returns. Also, the chassis command reboot should not be executed using opacmdall. Instead, use the opachassisadmin reboot command to reboot one or more chassis. For further information about individual chassis CLI commands, consult the *Intel® Omni-Path Fabric Switches Command Line Interface Reference Guide*. For further information about Linux* operating system commands, consult the man pages.

When performing opacmdall against hosts, internally SSH is used. The command opacmdall requires that password-less SSH be set up between the host running the Intel® Omni-Path Fabric Suite FastFabric Toolset and the hosts opacmdall is operating against. The opasetupssh FastFabric tool can aid in setting up password-less SSH.

When performing opacmdall against a set of chassis, all chassis must be configured with the same admin password. Alternatively, the opasetupssh FastFabric tool can be used to set up password-less SSH to the chassis.

When performing operations against chassis, Intel recommends that you set up SSH keys (see opasetupssh). If SSH keys are not set up, Intel recommends that you use the -S option, to avoid keeping the password in configuration files.

3.6.4 opacaptureall

(Chassis and Host) Captures supporting information for a problem report from all hosts or Intel® Omni-Path Chassis and uploads to this system.

For Hosts: When a host opacaptureall is performed, opacapture 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.

For Chassis: When a chassis opacaptureall is performed, opacapture is run on each chassis and its output is saved to upload_dir/*chassisname*/*file* on the local system. The default file name is chassiscapture.

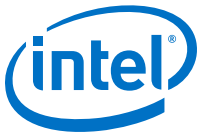
For both host and chassis capture, the uploaded captures are combined into a .tgz file with the file name specified and the suffix .all.tgz added.

Syntax

```
opacaptureall [-C] [-p] [-f hostfile] [-F chassisfile] [-h 'hosts']
[-H 'chassis'] [-t portsfile] [-d upload_dir] [-S] [-D detail_level]
[file]
```

Options

--help	Produces full help text.
-C	Performs capture against chassis. Default is hosts.
-p	Performs capture upload in parallel on all host/chassis. For a host capture, this only affects the upload phase.
-f *hostfile*	Specifies the file with hosts in cluster. Default is /etc/sysconfig/opa/hosts file.
-F *chassisfile*	Specifies the file containing a list of chassis in the cluster. Default is /etc/sysconfig/opa/chassis file.



`-h hosts` Specifies the list of hosts on which to perform a capture.

`-H chassis` Specifies the list of chassis on which to perform a capture.

`-t portsfile` Specifies the file with list of local HFI ports used to access fabric(s) for switch access, default is `/etc/sysconfig/opa/ports` file.

`-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.

`-S` Securely prompts for password for administrator on a chassis.

`-D detail_level` Specifies the level of detail of the capture passed to host opacapture. (Only used for host captures; ignored for chassis captures.)

`1` (Local)	Obtains local information from each host.
`2` (Fabric)	In addition to *Local*, also obtains basic fabric information by queries to the SM and fabric error analysis using `opareport`.
`3` (Fabric +FDB)	In addition to *Fabric*, also obtains the Forwarding Database (FDB), which includes the switch forwarding tables from the SM.
`4` (Analysis)	In addition to *Fabric+FDB*, also obtains `opaallanalysis` results. If `opaallanalysis` has not yet been run, it is run as part of the capture.

Note: Detail levels 2-4 can be used when fabric operational problems occur. If the problem is node-specific, detail level 1 should be sufficient. Detail levels 2-4 require an operational Intel® Omni-Path Fabric Suite 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 2-4, the additional information is only gathered on the node running the `opacaptureall` command. The information is gathered for every fabric specified in the `/etc/sysconfig/opa/ports` file.

file Specifies the name for capture file. The suffix `.tgz` is appended if it is not specified in the name.

Examples

Host Capture Examples

```
opacaptureall
# Creates  a hostcapture directory in upload_dir/hostname/ for each host in
/etc/sysconfig/opa/hosts file, then creates hostcapture.all.tgz.

opacaptureall mycapture
# Creates a mycapture directory in upload_dir/hostname/ for  each  host  in
/etc/sysconfig/opa/hosts file, then creates mycapture.all.tgz.

opacaptureall -h 'arwen elrond' 030127capture
# Gets  the  list  of  hosts  from arwen elrond file and creates
030127capture.tgz file.
```

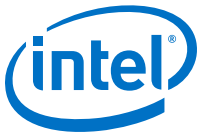
Chassis Capture Examples

```
opacaptureall -C
# Creates a chassiscapture file in upload_dir/chassisname/ for each chassis
in /etc/sysconfig/opa/chassis file, then creates chassiscapture.all.tgz.

opacaptureall -C mycapture
# Creates  a mycapture.tgz file in upload_dir/chassisname/ for each chassis
in /etc/sysconfig/opa/chassis file, then creates mycapture.all.tgz.

opacaptureall -C -H 'chassis1 chassis2' 030127capture
# Captures from chassis1 and chassis2, 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. See discussion on Selection of Devices on page 17.
CHASSIS	List of chassis, used if -C is used and -h option is not supplied. See discussion on Selection of Devices on page 17.
HOSTS_FILE	File containing a list of hosts, used in the absence of -f and -h. See discussion on Selection of Devices on page 17.
CHASSIS_FILE	File containing a list of chassis, used in the absence of -F and -H. See discussion on Selection of Devices on page 17.
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.
FF_CHASSIS_LOGIN_METHOD	How to log into chassis. Can be Telnet or SSH.



FF_CHASSIS_ADMIN_PASSWORD	Password for administrator on all chassis. Used in absence of -S option.

More Information

When performing opacaptureall against hosts, internally SSH is used. The command opacaptureall requires that password-less SSH be set up between the host running Intel® Omni-Path Fabric Suite FastFabric Toolset and the hosts opacaptureall is operating against. The opasetupssh command can aid in setting up password-less SSH.

When performing operations against chassis, set up of SSH keys is recommended (see opasetupssh on page 136). If SSH keys are not set up, all chassis must be configured with the same admin password and use of the -S option is recommended. The -S option avoids the need to keep the password in configuration files.

Note: The resulting host capture files can require significant amounts of space on the Intel® Omni-Path 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® Omni-Path Fabric Suite FastFabric Toolset system. In many cases, it may not be necessary to run opacaptureall against all hosts or chassis; instead, a representative subset may be sufficient. Consult with your support representative for further information.

3.7 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 opasetupssh to set up password-less SSH/SCP.

3.7.1 opascpall

(Linux) 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 opauploadall.
- user@ style syntax cannot be used when specifying filenames.

Syntax

```
opascpall [-p] [-r] [-f hostfile] [-h 'hosts'] [-u user] source_file ... dest_file

opascpall [-t] [-p] [-f hostfile] [-h 'hosts'] [-u user] [source_dir [dest_dir]]
```

Options

`--help`	Produces full help text.
`-p`	Performs copy in parallel on all hosts.
`-r`	Performs recursive copy of directories.
`-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/sysconfig/opa/hosts` file.
`-u` *user*	Specifies the user to perform copy to. Default is current user code.
source_file	Specifies the a file or list of source files to copy.
source_dir	Specifies the name of the source directory to copy. If omitted `.` is used.
dest_file or *dest_dir*	Specifies the name of the destination file or directory to copy to. If more than one source file, this must be a directory. If omitted current directory name is used.

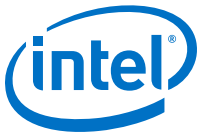
Example

```
# copy a single file
opascpall MPI-PMB /root/MPI-PMB

# efficiently copy an entire directory tree
opascpall -t -p /usr/lib/opa/src/mpi_apps /usr/lib/opa/src/mpi_apps

# copy a group of files
opascpall a b c /root/tools/

# copy to an explitly specified set of hosts
opascpall -h 'arwen elrond' a b c /root/tools
HOSTS='arwen elrond' opascpall 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. See discussion on Selection of Devices on page 17.



HOSTS_FILE	File containing list of hosts; used in absence of -f and -h. See discussion on Selection of Devices on page 17.
FF_MAX_PARALLEL	When the -p option is used, maximum concurrent operations are performed.

3.7.2 opauploadall

(Linux) 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 opadownloadall to upload a host specific configuration file, edit it for each host, and download the new version to all the hosts.

Note: To copy files from this host to hosts in the cluster, use opascpall or opadownloadall. user@ style syntax cannot be used when specifying filenames.

Syntax

```
opauploadall [-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/sysconfig/opa/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 code.
-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*.

A local directory within *upload_dir/* is created for each host. Each uploaded file is copied to upload_dir/*HOSTNAME/dest_file* within the local system. If more than one source file is specified, dest_file is treated as a directory name.

Example

```
# upload two files from 2 hosts
opauploadall -h 'arwen elrond' capture.tgz /etc/init.d/ipoib.cfg .

# upload two files from all hosts
opauploadall -p capture.tgz /etc/init.d/ipoib.cfg .

# upload network config files from all hosts
opauploadall 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. See discussion on Selection of Devices on page 17.
HOSTS_FILE	File containing list of hosts; used in absence of -f and -h. See discussion on Selection of Devices on page 17.
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.

3.7.3 opadownloadall

(Linux) 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 opauploadall to upload a host-specific configuration file, edit it for each host, and download the new version to all the hosts.

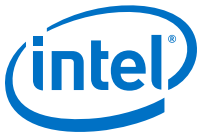
Note: The tool opadownloadall can only copy from this system to a group of hosts in the cluster. To copy files from hosts in the cluster to this host, use opauploadall.

Syntax

```
opadownloadall [-rp] [-f hostfile] [-d download_dir] [-h 'HOSTS']
[-u user] source_file ... dest_file
```

Options

--help	Produces full help text.
-r	Performs recursive download of directories.



`-p`	Performs copy in parallel on all hosts.
`-f` *`hostfile`*	Specifies the file with hosts in cluster. The default is `/etc/sysconfig/opa/hosts`.
`-d` *`download_dir`*	Specifies the directory to download files from. The default is `downloads`. If not specified, the environment variable `DOWNLOADS_DIR` is used. If that is not exported, the default is used.
`-h` *`HOSTS`*	Specifies the list of hosts to download files to.
`-u` *`user`*	Specifies the user to perform the copy. The default is the current user code. *Note:* The `user@` style syntax cannot be used in the arguments to `opadownloadall`.
`source_file`	Specifies the list of source files to copy from the system. 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. 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

```
opadownloadall -h 'arwen elrond' irqbalance vncservers /etc/sysconfig
# Copies two files to 2 hosts

opadownloadall -p irqbalance vncservers /etc/sysconfig
# 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. See discussion on Selection of Devices on page 17.
`HOSTS_FILE`	File containing list of hosts; used in absence of `-f` and `-h`. See discussion on Selection of Devices on page 17.

FF_MAX_PARALLEL	When the -p option is used, the maximum concurrent operations are performed.
DOWNLOADS_DIR	Directory to download from, used in absence of -d.

3.7.4 Simplified Editing of Node-Specific Files

(Linux) The combination of opauploadall and opadownloadall provide a powerful yet simple to use mechanism for reviewing or editing node-specific files without the need to log in to each node.

For example, assume the file /etc/sysconfig/network-scripts/ifcfg-ib1 needs to be reviewed and edited for each host. This file typically contains the IP configuration information for IPoIB 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/sysconfig/network-scripts/ifcfg-ib1 ifcfg-ib1
2. Edit the uploaded files with an editor, such as vi with the command: vi uploads/*/ifcfg-ib1
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: opadownloadall -d uploads ifcfg-ib1 /etc/sysconfig/network-scripts/ifcfg-ib1

Alternatively, you can download the file to a subset of hosts using the -h option or by creating an alternate host list file: opadownloadall -d uploads -h 'host1 host32' ifcfg-ib1 /etc/sysconfig/network-scripts/ifcfg-ib1

Note: When downloading to a subset of hosts, make sure that only the hosts uploaded from are specified.

3.7.5 Simplified Setup of Node-Generic Files

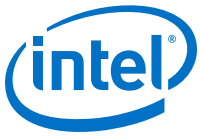
(Linux) opascpall can provide a powerful yet simple to use mechanism for transferring generic files to all nodes.

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: opascpall resolv.conf /etc/resolv.conf

3.8 Fabric Link and Port Control

The CLIs described in this section are used for manipulation of device and port states in the fabric.



3.8.1 opadisableports

(Linux) Accepts a CSV file listing links to disable. For each HFI-SW link, the switch side of the link is disabled. For each SW-SW link, the side of the link with the lower LID (typically, the side closest to the SM) is disabled. This approach generally permits a future `opaenableports` operation to re-enable the port once the issue is corrected or ready to be retested. When using the `-R` option, this tool does not look at the routes, it disables the switch ports with the lower value LID. The list of disabled ports is tracked in `/etc/sysconfig/opa/disabled*.csv`.

Syntax

```
opadisableports [-R] [-h hfi] [-p port]
[reason] < disable.csv
```

Options

`--help` Produces full help text.

`-R` Does not attempt to get routes for computation of distance. Instead, disables switch port with lower LID assuming that it is closer to this node.

`-h` *hfi* Specifies the HFI, numbered 1..n. Using 0 specifies that the `-p` *port* port is a system-wide port number. (Default is 0.)

`-p` *port* Specifies the port, numbered 1..n. Using 0 specifies the first active port. (Default is 0.)

reason Specifies optional text describing why ports are being disabled. If used, text is saved in the reason field of the output file.

`disable.csv` Specifies the input file listing the links to disable. The list is of the form:
`NodeGUID;PortNum;NodeType;NodeDesc;NodeGUID;PortNum;NodeType;NodeDesc;Reason`

For each listed link, the switch port closer to this node is disabled. The *reason* field is optional. An input file such as this can be generated by using `opaextractbadlinks`, `opaextractmissinglinks`, or `opaextractsellinks`.

Information about the links disabled and the reason is saved (in the same format) to an output file named `/etc/sysconfig/opa/disabled:hfi:port.csv` where the `hfi:port` part of the file name is replaced by the HFI number and the port number being operated on (such as 0:0 or 1:2). This CSV file can be used as input to `opaenableports`.

-h and -p options permit a variety of selections:

`-h 0` First active port in system (default).

`-h 0 -p 0` First active port in system.

`-h x` First active port on HFI *x*.

`-h x -p 0` First active port on HFI *x*.

`-h 0 -p y` Port *y* within system (no matter which ports are active).

`-h x -p y` HFI *x*, port *y*.

Examples

```
opadisableports 'bad cable' < disable.csv
opadisableports -h 1 -p 1 'dead servers' < disable.csv
opaextractsellinks -F lid:3 | opadisableports 'bad server'
opaextractmissinglinks -T /etc/sysconfig/opa/topology.0:0.xml | opadisableports
```

3.8.2 opaenableports

(Linux) Accepts a disabled ports input file and re-enables the specified ports. The input file can be `/etc/sysconfig/opa/disabled*.csv` or a user-created subset of such a file. After enabling the port, it is removed from `/etc/sysconfig/opa/disabled*.csv`.

Syntax

```
opaenableports [-h hfi] [-p port] < disabled.csv
```

Options

`--help` Produces full help text.

`-h hfi` Specifies the HFI, numbered 1..n. Using 0 specifies that the `-p port` port is a system-wide port number. (Default is 0.)

`-p port` Specifies the port, numbered 1..n. Using 0 specifies the first active port. (Default is 0.)

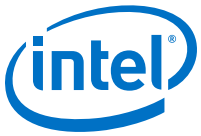
`disabled.csv` Specifies the input file listing the ports to enable. The list is of the form: `NodeGUID;PortNum;NodeType;NodeDesc;Ignored`.

An input file like this is generated in `/etc/sysconfig/opa/disabled*` by `opadisableports`.

-h and -p options permit a variety of selections:

`-h 0` First active port in system (default).

`-h 0 -p 0` First active port in system.



`-h x` First active port on HFI *x*.

`-h x -p 0` First active port on HFI *x*.

`-h 0 -p y` Port *y* within system (no matter which ports are active).

`-h x -p y` HFI *x*, port *y*.

Examples

```
opaenableports < disabled.csv
opaenableports < /etc/sysconfig/opa/disabled:0:0.csv
opaenableports -h 1 -p 1 < disabled.csv
```

3.8.3 opadisablehosts

(Linux) Searches for a set of hosts in the fabric and disables their corresponding switch port.

Syntax

```
opadisablehosts [-h hfi] [-p port] reason host
```

Options

`--help` Produces full help text.

`-h hfi` Specifies the HFI, numbered 1..n. Using 0 specifies that the `-p port` port is a system-wide port number. (Default is 0.)

`-p port` Specifies the port, numbered 1..n. Using 0 specifies the first active port. (Default is 0.)

reason Specifies the text describing the reason hosts are being disabled. *reason* is saved in the `reason` field of the output file.

Information about the links disabled is written to a CSV file. By default, this file is named `/opa/disabled:hfi:port.csv` where the `hfi:port` part of the file name is replaced by the HFI number and the port number being operated on (such as 0:0 or 1:2). This CSV file can be used as input to `opaenableports`.

The list is of the form:
`NodeGUID;PortNum;NodeType;NodeDesc;NodeGUID;PortNum;NodeType;NodeDesc;Reason` For each listed link, the switch port closer to this is the one that has been disabled.

-h and -p options permit a variety of selections:

`-h 0` First active port in system (default).

`-h 0 -p 0` First active port in system.

`-h x` First active port on HFI *x*.

`-h x -p 0` First active port on HFI *x*.

`-h 0 -p y` Port *y* within system (no matter which ports are active).

`-h x -p y` HFI *x*, port *y*.

Examples

```
opadisablehosts 'bad DRAM' compute001 compute045
opadisablehosts -h 1 -p 2 'crashed' compute001 compute045
```

3.8.4 opaswdisableall

(Linux) Disables all unused switch ports.

Syntax

```
opaswdisableall [-t portsfile] [-p ports] [-F focus] [-K mkey]
```

Options

`--help` Produces full help text.

`-t portsfile` Specifies the file with list of local HFI ports used to access fabrics when clearing counters. Default is `/etc/sysconfig/opa/ports` file.

`-p ports` Specifies the list of local HFI ports used to access fabrics for counter clear.

Default is first active port. The first HFI in the system is 1. The first port on an HFI is 1. Uses the format hfi:port, for example:

`0:0` First active port in system.

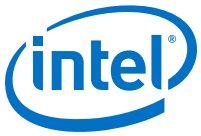
`0:y` Port *y* within system.

`x:0` First active port on HFI *x*.

`x:y` HFI *x*, port *y*.

`-F focus` Specifies the an `opareport`-style focus argument to limit the scope of operation. For more information, see Advanced Focus on page 89.

`-K mkey` Specifies the SM management key to access remote ports.



Examples

```
opaswdisableall
opaswdisableall -p '1:1 1:2 2:1 2:2'
```

Environment Variables

The following environment variables are also used by this command:

`PORTS` List of ports, used in absence of `-t` and `-p`.

`PORTS_FILE` File containing list of ports, used in absence of `-t` and `-p`.

3.8.5 opaswenableall

(Linux) Re-enables all unused (or disabled) switch ports.

Syntax

```
opaswenableall [-t portsfile] [-p ports] [-F focus] [-K mkey]
```

Options

`--help` Produces full help text.

`-t portsfile` Specifies the file with list of local HFI ports used to access fabrics for operation. Default is `/etc/sysconfig/opa/ports` file.

`-p ports` Specifies the list of local HFI ports used to access fabrics for operation.

Default is first active port. The first HFI in the system is 1. The first port on an HFI is 1. Uses the format hfi:port, for example:

`0:0` First active port in system.

`0:y` Port *y* within system.

`x:0` First active port on HFI *x*.

`x:y` HFI *x*, port *y*.

`-F focus` Specifies an `opareport`-style focus argument to limit the scope of operation. For more information, see Advanced Focus on page 89.

`-K mkey` Specifies the SM management key to access remote ports.

Examples

```
opaswenableall
opaswenableall -p '1:1 1:2 2:1 2:2'
```

Environment Variables

The following environment variables are also used by this command:

PORTS	List of ports, used in absence of -t and -p.
PORTS_FILE	File containing list of ports, used in absence of -t and -p.

3.8.6 opaledports

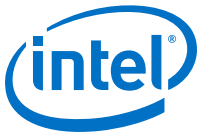
Toggles the beaconing LED state of HFIs, switches, and switch ports. opaledports is a useful aid for finding specific physical nodes in a crowded data center. It supports the CSV link format provided by opaextractsellinks.

Syntax

```
opaledports [-h hfi] [-p port] [-C] [-s|-d] [on|off] < portlist.csv
```

Options

--help	Produces full help text.
-h *hfi*	Specifies the HFI, numbered 1..n. Using 0 specifies that the -p *port* port is a system- number. (Default is 0.)
-p *port*	Specifies the port, numbered 1..n. Using 0 specifies the first active port. (Default is 0.)
-C	Clears beaconing LED on all ports. *Note:* If -C is entered, no other options are valid.
-s	Affects source side (first node) of link only.
-d	Affects destination side (second node) of link only.
on|off	Turns on or off the beaconing LED. Options include: on Turns on beaconing LED. off Turns off beaconing LED.
portlist.csv	Specifies the file listing the links to process. The list is of the form: NodeGUID;PortNum;NodeType;NodeDesc;NodeGUID;PortNum;NodeType;NodeD



Examples

```
echo "0x001175010165ac1d;1;FI;phkpstl035 hfi1_0"|opaledports on
opaledports on < portlist.csv
opaextractsellinks -F led:on | opaledports off
opaledports -C
```

3.9 Fabric Debug

The CLIs described in this section are used for gathering various fabric information from the FM for debug and analysis purposes.

3.9.1 opafequery

(All) Used for testing or debugging performance administration (PA) operations to the Fabric Executive (FE). This tool performs custom PA client/server queries. The output formats and arguments are very similar to `opapaquery`.

Syntax

```
opafequery [-v] [-a ipAddr | -h hostName] [-E] [-T paramsfile] -o type
[SA options | PA options]
```

General Options

`--help`	Produces full help text.
`-v/--verbose`	Specifies the verbose output.
`-a/--ipAddr` *ipAddr*	Specifies the IP address of node running the FE. This options supports IPv4 and IPv6 addresses with port number; for example, *127.0.0.1:3245* or *[::1]:3245*.
`-h/--hostName` *hostName*	Specifies the host name of node running the FE. This option supports host name with port number; for example, *localhost:3245*.
`-o/--output` *output*	Specifies the output type. See SA Output Types and PA Output Types for details.
`-E/--feEsm` *ESMName*	Specifies the ESM FE name.
`-T/--sslParmsFile` *filename*	Specifies the SSL/TLS parameters XML file. Default = `/etc/sysconfig/opa/opaff.xml`

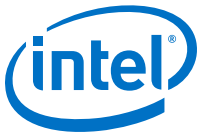
SA Specific Options

`-I/--IB`	Issues query in legacy InfiniBand* format.

-l/--lid *lid*	Queries a specific LID.
-k/--pkey *pkey*	Queries a specific pkey.
-i/--vfindex *vfindex*	Queries a specific vfindex.
-S/--serviceId *serviceId*	Queries a specific service ID.
-L/--SL *SL*	Queries by service level.
-t/--type *type*	Queries by node type.
-s/--sysguid *guid*	Queries by system image GUID.
-n/--nodeguid *guid*	Queries by node GUID.
-p/--portguid *guid*	Queries by port GUID.
-u/--portgid *gid*	Queries by port GID.
-m/--mcgid *gid*	Queries by multicast GID.
-d/--desc *name*	Queries by node name/description.
-P/--guidpair *'guid guid'*	Queries by a pair of port GUIDs.
-G/--gidpair *'gid gid'*	Queries by a pair of GIDs.
-B/--guidlist *'sguid ...;dguid ...'*	Queries by a list of port GUIDs.
-A/--gidlist *'sgid ...;dgid ...'*	Queries by a list of GIDs.
-x/--sourcegid *gid*	Specifies a source GID for certain queries.

PA Specific Options

-g/--groupName *groupName*	Queries by group name for groupInfo.
-l/--lid *lid*	Queries by LID of node for portCounters.
-N/--portNumber	Queries by port number for portCounters.
-f/--delta	Queries by delta flag for portCounters. Values include: 0 or 1.
-j/--begin *date_time*	Obtains portCounters over an interval beginning at *date_time*.



date_time may be a time entered as HH:MM[:SS] or date as mm/dd/YYYY, dd.mm.YYYY, YYYY-mm-dd or date followed by time; for example, "2016-07-04 14:40". Relative times are taken as "x [second|minute|hour|day](s) ago.

`-q/--end` *date_time*

Obtains portCounters over an interval ending at *date_time*.

date_time may be a time entered as HH:MM[:SS] or date as mm/dd/YYYY, dd.mm.YYYY, YYYY-mm-dd or date followed by time; for example, "2016-07-04 14:40". Relative times are taken as "x [second|minute|hour|day](s) ago.

`-U/--userCntrs`

Queries by user-controlled counters flag for portCounters.

`-e/--select`

Specifies the 32-bit select flag for clearing port counters select bits. 0 is least significant (rightmost).

Bit descriptions are listed below in the order "mask - bit - location":

- 0x80000000 - 31 - Transmit Data (XmitData)
- 0x40000000 - 30 - Receive Data (RcvData)
- 0x20000000 - 29 - Transmit Packets (XmitPkts)
- 0x10000000 - 28 - Receive Packets (RcvPkts)
- 0x08000000 - 27 - Multicast Transmit Packets (MulticastXmitPkts)
- 0x04000000 - 26 - Multicast Receive Packets (MulticastRcvPkts)
- 0x02000000 - 25 - Transmit Wait (XmitWait)
- 0x01000000 - 24 - Congestion Discards (CongDiscards)
- 0x00800000 - 23 - Receive FECN (RcvFECN)
- 0x00400000 - 22 - Receive BECN (RcvBECN)
- 0x00200000 - 21 - Transmit Time Congestion (XmitTimeCong)
- 0x00100000 - 20 - Transmit Time Wasted BW (XmitWastedBW)
- 0x00080000 - 19 - Transmit Time Wait Data (XmitWaitData)
- 0x00040000 - 18 - Receive Bubble (RcvBubble)
- 0x00020000 - 17 - Mark FECN (MarkFECN)
- 0x00010000 - 16 - Receive Constraint Errors (RcvConstraintErrors)
- 0x00008000 - 15 - Receive Switch Relay (RcvSwitchRelayErrors)
- 0x00004000 - 14 - Transmit Discards (XmitDiscards)
- 0x00002000 - 13 - Transmit Constraint Errors (XmitConstraintErrors)

- 0x00001000 - 12 - Receive Remote Physical Errors (RcvRemotePhysicalErrors)
- 0x00000800 - 11 - Local Link Integrity (LocalLinkIntegrityErrors)
- 0x00000400 - 10 - Receive Errors (RcvErrors)
- 0x00000200 - 9 - Excessive Buffer Overrun (ExcessiveBufferOverruns)
- 0x00000100 - 8 - FM Configuration Errors (FMConfigErrors)
- 0x00000080 - 7 - Link Error Recovery (LinkErrorRecovery)
- 0x00000040 - 6 - Link Error Downed (LinkDowned)
- 0x00000020 - 5 - Uncorrectable Errors (UncorrectableErrors)

`-c/--focus` *`focus`*
: Specifies the focus select value for getting focus ports. Values include:

  `utilhigh` — Sorted by utilization - highest first.

  `pktrate` — Sorted by packet rate - highest first.

  `utillow` — Sorted by utilization - lowest first.

  `integrity` — Sorted by integrity errors - highest first.

  `congestion` — Sorted by congestion errors - highest first.

  `smacongesion` — Sorted by SMA congestion errors - highest first.

  `bubbles` — Sorted by bubble errors - highest first.

  `security` — Sorted by security errors - highest first.

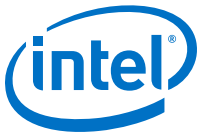
  `routing` — Sorted by routing errors - highest first.

`-w/--start`
: Specifies the start of window for focus ports - should always be 0.

`-r/--range` *`range`*
: Specifies the size of window for focus ports list.

`-b/--imgNum`
: Specifies the 64-bit image number. May be used with `groupInfo`, `groupConfig`, `portCounters (delta)` outputs.

`-O/--imgOff`
: Specifies the image offset. May be used with `groupInfo`, `groupConfig`, `portCounters (delta)` outputs.



`-y/--imgTime`	Specifies the image time. May be used with `imageinfo`, `groupInfo`, `groupInfo`, `groupConfig`, `freezeImage`, `focusPorts`, `vfInfo`, `vfConfig`, and `vfFocusPorts` . Will return closest image within image interval if possible. See `--begin/--end` above for format.
`-F/--moveImgNum`	Specifies the 64-bit image number. Used with `moveFreeze` output to move a freeze image.
`-M/--moveImgOff` *ImgOff*	Specifies the image offset. May be used with `moveFreeze` output to move a freeze image.
`-V/--vfName`	Queries by VF name for vfInfo.

SA Output Types

Output types include:

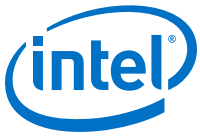
`saclassPortInfo`	Specifies the class port info.
`systemguid`	Lists the system image GUIDs.
`nodeguid`	Lists the node GUIDs.
`portguid`	Lists the port GUIDs.
`lid`	Lists the LIDs.
`desc`	Lists the node descriptions/names.
`path`	Lists the path records.
`node`	Lists the node records.
`portinfo`	Lists the port info records.
`sminfo`	Lists the SM info records.
`swinfo`	Lists the switch info records.
`link`	Lists the link records.
`scsc`	Lists the SC to SC mapping table records.
`slsc`	Lists the SL to SC mapping table records.
`scsl`	Lists the SC to SL mapping table records.
`scvlt`	Lists the SC to VLt table records.

`scvlnt`	Lists the SC to VLnt table records.
`vlarb`	Lists the VL arbitration table records.
`pkey`	Lists the PKey table records.
`service`	Lists the service records.
`mcmember`	Lists the multicast member records.
`inform`	Lists the inform info records.
`linfdb`	Lists the switch linear forwarding database (FDB) records.
`mcfdb`	Lists the switch multicast FDB records.
`trace`	Lists the trace records.
`vfinfo`	Lists the vFabrics.
`vfinfocsv`	Lists the vFabrics in CSV format.
`vfinfocsv2`	Lists the vFabrics in CSV format with enums.
`fabricinfo`	Provides a summary of fabric devices.
`quarantine`	Lists the quarantined nodes.
`conginfo`	Lists the Congestion Info Records.
`swcongset`	Lists the Switch Congestion Settings.
`hficongset`	Lists the HFI Congestion Settings.
`hficongcon`	Lists the HFI Congestion Control Settings.
`bfrctrl`	Lists the buffer control tables.
`cableinfo`	Lists the Cable Info records.
`portgroup`	Lists the AR Port Group records.
`portgroupfdb`	Lists the AR Port Group FWD records.

PA Output Types

Output types include:

`paclassPortInfo`	Specifies the class port info.



`groupList`	Lists the PA groups.
`groupInfo`	Provides a summary statistics of a PA group. Requires `-g` option for *`groupName`*.
`groupConfig`	Specifies the configuration of a PA group. Requires `-g` option for *`groupName`*.
`portCounters`	Specifies the port counters of fabric port. Requires `-l` *`lid`* and `-N` *`port`* options. Optionally, use the `-f` *`delta`* option.
`clrPortCounters`	Clears port counters of fabric port. Requires `-l` *`lid`*, `-N` *`port`*, and `-e` *`select`* options.
`clrAllPortCounters`	Clears all port counters in fabric.
`pmConfig`	Retrieves PM configuration information.
`freezeImage`	Creates freeze frame for image ID. Requires `-b` *`imgNum`*.
`releaseImage`	Releases freeze frame for image ID. Requires `-b` *`imgNum`*.
`renewImage`	Renews lease for freeze frame for image ID. Requires `-b` *`imgNum`*.
`moveFreeze`	Moves freeze frame from image ID to new image ID. Requires `-b` *`imgNum`* and `-F` *`moveImgNum`*.
`focusPorts`	Gets sorted list of ports using utilization or error values (from group buckets).
`imageInfo`	Gets information about a PA image (timestamps and other details). Requires `-b` *`imgNum`*.
`vfList`	Lists the virtual fabrics.
`vfInfo`	Provides a summary statistics of a virtual fabric. Requires `-V` *`vfName`* option.
`vfConfig`	Specifies the configuration of a virtual fabric. Requires `-V` *`vfName`* option.
`vfPortCounters`	Specifies the port counters of fabric port. Requires `-V` *`vfName`*, `-l` *`lid`*, and `-N` *`port`* options. Optionally, use the `-f` *`delta`* option.
`vfFocusPorts`	Gets sorted list of virtual fabric ports usingutilization or error values (from VF buckets). Requires `-V` *`vfName`* option.

clrVfPortCounters	Clears VF port counters of fabric port. Requires -l *lid*, -N *port*, -e *select*, and -V *vfName* options.

Examples

```
opafequery -o saclassPortInfo
opafequery -h stewie -o paclassPortInfo
opafequery -a 172.21.2.155  -o saclassPortInfo
opafequery -o groupList
opafequery -o groupInfo -g All
opafequery -o groupConfig -g All
opafequery -h stewie -o groupInfo -g All
opafequery -a 172.21.2.155 -o groupInfo -g All
opafequery -o portCounters -l 1 -N 1 -d 1
opafequery -o portCounters -l 1 -N 1 -d 1 -e 0x20000000d02 -O 1
opafequery -o pmConfig
opafequery -o freezeImage 0x20000000d02
opafequery -o releaseImage -b 0xd01
opafequery -o renewImage -b 0xd01
opafequery -o moveFreeze -b 0xd01 -m 0x20000000d02 -M -2
opafequery -o focusPorts -g All -f 0x00030001 -w 0 -r 20
opafequery -o imageInfo -b 0x20000000d02
```

3.9.2 opapaquery

(All) Performs various queries of the performance management (PM)/performance administration (PA) agent and provides details about fabric performance. Refer to the *Intel® Omni-Path Fabric Suite Fabric Manager User Guide* for a description of the operation and client services of the PM/PA.

By default, opapaquery queries the most recent data. However, if an image number (*imgNum*) and/or image offset (*imgOff*) is provided, the query returns previous sweep data. Queries that access previous sweep data return with the absolute image number representing that data, and therefore have an image offset of zero.

opapaquery's operation is dependent on an Intel® Omni-Path Fabric Suite Fabric Manager version 6.0 or greater running as master SM/PM in the fabric.

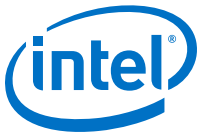
By default, opapaquery uses the first active port on the local system. However, if the Fabric Management Node is connected to more than one fabric (for example, a subnet), the HFI and port may be specified to select the fabric whose PA is to be queried.

Syntax

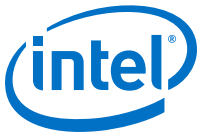
```
opapaquery [-v] [-h hfi] [-p port] [-o type] [-g groupName] [-l nodeLid]
[-P portNumber] [-d delta] [-j date_time] [-q date_time] [-U] [-s select]
[-f focus] [-S start] [-r range] [-n imgNum] [-O imgOff] [-y imgTime]
[-m moveImgNum] [-M moveImgOff] [-V vfName]
```

Options

-v/--verbose	Specifies the verbose output.
-h/--hfi *hfi*	Specifies the HFI, numbered 1..n. Using 0 specifies that the -p *port* port is a system-wide port number. (Default is 0.)



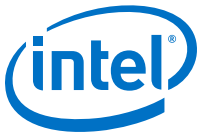
`-p/--port` *port*	Specifies the port, numbered 1..n. Using 0 specifies the first active port. (Default is 0.)
`-o/--output` *type*	Specifies the output type, default is groupList. See Output Types on page 168.
`-g/--groupName` *groupName*	Specifies the group name for `groupInfo` query.
`-l/--lid` *lid*	Specifies the LID of node for `portCounters` query.
`-P/--portNumber` *portNumber*	Specifies the port number for `portCounters` query.
`-d/--delta` *delta*	Specifies the delta flag for `portCounters` query - 0 or 1.
`-j/--begin` *date_time*	Obtains portCounters over an interval beginning at *date_time*. *date_time* may be a time entered as HH:MM[:SS] or date as mm/dd/YYYY, dd.mm.YYYY, YYYY-mm-dd or date followed by time; for example, "2016-07-04 14:40". Relative times are taken as "x [second|minute|hour|day](s) ago.
`-q/--end` *date_time*	Obtains portCounters over an interval ending at *date_time*. *date_time* may be a time entered as HH:MM[:SS] or date as mm/dd/YYYY, dd.mm.YYYY, YYYY-mm-dd or date followed by time; for example, "2016-07-04 14:40". Relative times are taken as "x [second|minute|hour|day](s) ago.
`-U/--userCntrs`	Queries by user-controlled counters flag for portCounters.
`-s/--select` *select*	Specifies the 32-bit select flag for clearing port counters. Select bits for `clrPortCounters`. 0 is the least significant bit (rightmost). The `clrPortCounters` bit descriptions are listed in the order "mask - bit - location" below: • 0x80000000 - 31 - Transmit Data (XmitData) • 0x40000000 - 30 - Receive Data (RcvData) • 0x20000000 - 29 - Transmit Packets (XmitPkts) • 0x10000000 - 28 - Receive Packets (RcvPkts) • 0x08000000 - 27 - Multicast Transmit Packets (MulticastXmitPkts) • 0x04000000 - 26 - Multicast Receive Packets (MulticastRcvPkts) • 0x02000000 - 25 - Transmit Wait (XmitWait) • 0x01000000 - 24 - Congestion Discards (CongDiscards)



- 0x00800000 - 23 - Receive FECN (RcvFECN)
- 0x00400000 - 22 - Receive BECN (RcvBECN)
- 0x00200000 - 21 - Transmit Time Congestion (XmitTimeCong)
- 0x00100000 - 20 - Transmit Time Wasted BW (XmitWastedBW)
- 0x00080000 - 19 - Transmit Time Wait Data (XmitWaitData)
- 0x00040000 - 18 - Receive Bubble (RcvBubble)
- 0x00020000 - 17 - Mark FECN (MarkFECN)
- 0x00010000 - 16 - Receive Constraint Errors (RcvConstraintErrors)
- 0x00008000 - 15 - Receive Switch Relay (RcvSwitchRelayErrors)
- 0x00004000 - 14 - Transmit Discards (XmitDiscards)
- 0x00002000 - 13 - Transmit Constraint Errors (XmitConstraintErrors)
- 0x00001000 - 12 - Receive Remote Physical Errors (RcvRemotePhysicalErrors)
- 0x00000800 - 11 - Local Link Integrity (LocalLinkIntegrityErrors)
- 0x00000400 - 10 - Receive Errors (RcvErrors)
- 0x00000200 - 9 - Excessive Buffer Overrun (ExcessiveBufferOverruns)
- 0x00000100 - 8 - FM Configuration Errors (FMConfigErrors)
- 0x00000080 - 7 - Link Error Recovery (LinkErrorRecovery)
- 0x00000040 - 6 - Link Error Downed (LinkDowned)
- 0x00000020 - 5 - Uncorrectable Errors (UncorrectableErrors)

Select bits for `clrVfPortCounters`. 0 is the least significant bit (rightmost). The `clrVfPortCounters` bit descriptions are listed in the order "mask - bit - location" below:

- 0x80000000 - 31 - VL Transmit Data (VLXmitData)
- 0x40000000 - 30 - VL Receive Data (VLRcvData)
- 0x20000000 - 29 - VL Transmit Packets (VLXmitPkts)
- 0x10000000 - 28 - VL Receive Packets (VLRcvPkts)
- 0x08000000 - 27 - VL Transmit Discards (VLXmitDiscards)
- 0x04000000 - 26 - VL Congestion Discards (VLCongDiscards)



- 0x02000000 - 25 - VL Transmit Wait (VLXmitWait)
- 0x01000000 - 24 - VL Receive FECN (VLRcvFECN)
- 0x00800000 - 23 - VL Receive BECN (VLRcvBECN)
- 0x00400000 - 22 - VL Transmit Time Congestion (VLXmitTimeCong)
- 0x00200000 - 21 - VL Transmit Wasted BW (VLXmitWastedBW)
- 0x00100000 - 20 - VL Transmit Wait Data (VLXmitWaitData)
- 0x00080000 - 19 - VL Receive Bubble (VLRcvBubble)
- 0x00040000 - 18 - VL Mark FECN (VLMarkFECN)
- Bits 17-0 reserved

`-f/--focus` *`focus`* Specifies the focus select value for getting *`focus`* ports. *`focus`* select values are:

`utilhigh`	Sorted by utilization - highest first.
`pktrate`	Sorted by packet rate - highest first.
`utillow`	Sorted by utilization - lowest first.
`integrity`	Sorted by integrity errors - highest first.
`congestion`	Sorted by congestion errors - highest first.
`smacongestion`	Sorted by SMA congestion errors - highest first.
`bubbles`	Sorted by bubble errors - highest first.
`security`	Sorted by security errors - highest first.
`routing`	Sorted by routing errors - highest first.

`-S/--start` *`start`* Specifies the start of window for focus ports, should always be 0.

`-r/--range` *`range`* Specifies the size of window for focus ports list.

`-n/--imgNum` *`imgNum`* Specifies the 64-bit image number. Can be used with `groupInfo`, `groupConfig`, `portCounters` (delta).

`-O/--imgOff` *`imgOff`* Specifies the image offset. Can be used with `groupInfo`, `groupConfig`, `portCounters` (delta).

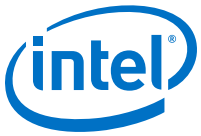
`-y/--imgTime`	Specifies the image time. May be used with `imageinfo`, `groupInfo`, `groupInfo`, `groupConfig`, `freezeImage`, `focusPorts`, `vfInfo`, `vfConfig`, and `vfFocusPorts`. Will return closest image within image interval if possible. See `--begin/--end` above for format.
`-m/--moveImgNum` *moveImgNum*	Specifies the 64-bit image number. Used with `moveFreeze` to move a freeze image.
`-M/--moveImgOff` *moveImgOff*	Specifies the image offset. Can be used with `moveFreeze` to move a freeze image.
`-V/--vfName`*vfName*	Specifies the VF name for vfInfo query.

-h and -p options permit a variety of selections:

`-h 0`	First active port in system (default).
`-h 0 -p 0`	First active port in system.
`-h` *x*	First active port on HFI *x*.
`-h` *x* `-p 0`	First active port on HFI *x*.
`-h 0 -p` *y*	Port *y* within system (no matter which ports are active).
`-h` *x* `-p` *y*	HFI *x*, port *y*.

Output Types

`classPortInfo`	Specifies the class port info.
`groupList`	Specifies the list of PA groups.
`groupInfo`	Specifies the summary statistics of a PA group. Requires `-g` option for *groupName*.
`groupConfig`	Specifies the configuration of a PA group. Requires `-g` option for *groupName*.
`portCounters`	Specifies the port counters of fabric port. Requires `-l` *lid* and `-P` *port* options, `-d` *delta* is optional.
`clrPortCounters`	Clears port counters of fabric port. Requires `-l` *lid* and `-P` *port*, and `-s` *select* options.
`clrAllPortCounters`	Clears all port counters in fabric.
`pmConfig`	Retrieves PM configuration information.



`freezeImage`	Creates freeze frame for image ID. Requires `-n` *`imgNum`*.
`releaseImage`	Releases freeze frame for image ID. Requires `-n` *`imgNum`*.
`renewImage`	Renews lease for freeze frame for image ID. Requires `-n` *`imgNum`*.
`moveFreeze`	Moves freeze frame from image ID to new image ID. Requires `-n` *`imgNum`* and `-m` *`moveImgNum`*.
`focusPorts`	Gets sorted list of ports using utilization or error values (from group buckets). Requires `-g` *`groupname`*, `-f` *`focus`*, `-S` *`start`*, `-r` *`range`*.
`imageInfo`	Gets configuration of a PA image (timestamps, etc.). Requires `-n` *`imgNum`*.
`vfList`	Specifies the list of virtual fabrics.
`vfInfo`	Specifies the summary statistics of a virtual fabric. Requires `-V` option for *`vfName`*.
`vfConfig`	Specifies the configuration of a virtual fabric. Requires `-V` option for *`vfName`*.
`vfPortCounters`	Specifies the port counters of fabric port. Requires `-V` *`vfName`*, `-l` *`lid`* and `-P` *`port`* options, `-d` *`delta`* is optional.
`vfFocusPorts`	Gets sorted list of virtual fabric ports using utilization or error values (from VF buckets). Requires `-V` *`vfname`*, `-f` *`focus`*, `-S` *`start`*, `-r` *`range`*.
`clrVfPortCounters`	Clears VF port counters of fabric port. Requires `-l` *`lid`*, `-P` *`port`*, `-s` *`select`*, and `-V` *`vfname`* options.

Examples

```
opapaquery -o classPortInfo
opapaquery -o groupList
opapaquery -o groupInfo -g All
opapaquery -o groupConfig -g All
opapaquery -o portCounters -l 1 -P 1 -d 1
opapaquery -o portCounters -l 1 -P 1 -d 1 -n 0x20000000d02 -O 1
opapaquery -o portCounters -l 1 -P 1 -d 1 -j 13:30 -q 14:20
opapaquery -o clrPortCounters -l 1 -P 1 -s 0xC0000000
   #clears XmitData & RcvData
opapaquery -o clrAllPortCounters -s 0xC0000000
   #clears XmitData & RcvData on all ports
opapaquery -o PMConfig
opapaquery -o freezeImage -n 0x20000000d02
opapaquery -o releaseImage -n 0xd01
opapaquery -o renewImage -n 0xd01
opapaquery -o moveFreeze -n 0xd01 -m 0x20000000d02 -M -2
opapaquery -o focusPorts -g All -f 0x00030001 -S 0 -r 20
```

```
opapaquery -o imageInfo -n 0x20000000d02
opapaquery -o imageInfo -y "1 hour ago"
opapaquery -o vfList
opapaquery -o vfInfo -V Default
opapaquery -o vfConfig -V Default
opapaquery -o vfPortCounters -l 1 -P 1 -d 1 -V Default
opapaquery -o clrVfPortCounters -l 1 -P 1 -s 0xC0000000
   #clears VLXmitData & VLRcvData
opapaquery -o vfFocusPorts -V Default -f 0x00030001 -S 0 -r 20
```

3.9.3 opasaquery

(All) Performs various queries of the subnet manager/subnet agent and provides detailed fabric information.

`opareport` and `opareports` can provide a more powerful tool, however, in some cases `opasaquery` is preferred, especially when dealing with virtual fabrics, service records, and multicast.

The command `opasaquery` is installed on all hosts as part of the Intel® Omni-Path Fabric Host Software, but it is also included in Intel® Omni-Path Fabric Suite FastFabric Toolset.

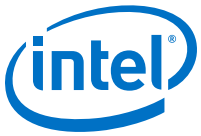
By default, `opasaquery` uses the first active port on the local system. However, if the node is connected to more than one fabric (for example, a subnet), the Intel® Omni-Path Host Fabric Interface (HFI) and port may be specified to select the fabric whose SA is to be queried.

Syntax

```
opasaquery [-v [-v] [-v]] [-I] [-h hfi] [-p port] [-o type] [-l lid]
[-t type] [-s guid] [-n guid] [-g guid] [-k pkey] [-i vfIndex]
[-S serviceId] [-L sl] [-u gid] [-m gid] [-d name] [-P 'guid guid']
[-G 'gid gid'] [-a 'sguid...;dguid...'] [-A 'sgid...;dgid...']
```

Options

`--help`	Produces full help text.
`-v/--verbose`	Returns verbose output. A second invocation activates `openib` debugging, a third invocation activates `libibumad` debugging.
`-I/--IB`	Issues query in legacy InfiniBand* format.
`-h/--hfi` *hfi*	Specifies the HFI, numbered 1..n. Using 0 specifies that the `-p` *port* port is a system-wide port number. (Default is 0.)
`-p/--port` *port*	Specifies the port, numbered 1..n. Using 0 specifies the first active port. (Default is 0.)



`-o` *type*	Output type for query (default is `node`). See Output Types for details.
`-l/--lid` *lid*	Query a specific LID.
`-t/--type` *node_type*	Queries by node type. See Node Types for details.
`-s/--sysguid` *system_image_guid*	Queries by system image GUID.
`-n/--nodeguid` *node_guid*	Queries by node GUID.
`-g/--portguid` *port_guid*	Queries by port GUID.
`-k/--pkey` *pkey*	Queries a specific PKey.
`-i/--vfindex` *vfIndex*	Queries a specific vfindex.
`-S/--serviceId` *serviceId*	Queries a specific service ID.
`-L/--SL` *SL*	Queries by service level.
`-u/--portgid` *port_gid*	Queries by port GID. See GIDs for details.
`-m/--mcgid` *multicast_gid*	Queries by multicast GID. See GIDs for details.
`-d/--desc` *node_description*	Queries by node name/description.
`-P/--guidpair` *guid guid*	Queries by a pair of port GUIDs.
`-G/--gidpair` *gid gid*	Queries by a pair of GIDs. See GIDs for details.
`-a/--guidlist` *sguid ...;dguid ...*	Queries by a list of port GUIDs.
`-A/--gidlist` *sgid ...;dgid ...*	Queries by a list of GIDs. See GIDs for details.

-h and -p options permit a variety of selections:

`-h 0`	First active port in system (default).
`-h 0 -p 0`	First active port in system.
`-h` *x*	First active port on HFI *x*.
`-h` *x* `-p 0`	First active port on HFI *x*.

`-h 0 -p y` Port *y* within system (no matter which ports are active).

`-h x -p y` HFI *x*, port *y*.

Node Types

`fi` Fabric Interface

`sw` Switch

GIDs

Specifies a 64-bit subnet and 64-bit interface ID in the form:

`subnet:interface`

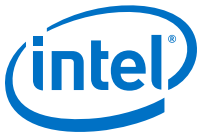
Note: In the following example, the GID corresponds to a PortGID. In this case, the interface ID coincides with the lower 64-bits of the GUID of the card. The interface ID will be different if the GID is a MGID (that is, multicast GID). See opafm.xml for MGID examples.

```
0xfe80000000000000:0x00117500a0000380
```

Output Types

Default is `node`.

`classportinfo`	Specifies the `classportinfo` of the SA.
`systemguid`	Lists the system image GUIDs.
`nodeguid`	Lists the node GUIDs.
`portguid`	Lists the port GUIDs.
`lid`	Lists the LIDs.
`desc`	Lists the node descriptions/names.
`path`	Lists the path records.
`node`	Lists the node records.
`portinfo`	Lists the port info records.
`sminfo`	Lists the SM info records.
`swinfo`	Lists the switch info records.
`link`	Lists the link records.



`scsc`	Lists the SC to SC mapping table records.
`slsc`	Lists the SL to SC mapping table records.
`scsl`	Lists the SC to SL mapping table records.
`scvlt`	Lists the SC to VLt table records.
`scvlnt`	Lists the SC to VLnt table records.
`vlarb`	Lists the VL arbitration table records.
`pkey`	Lists the PKey table records.
`service`	Lists the service records.
`mcmember`	Lists the multicast member records.
`inform`	Lists the inform info records.
`linfdb`	Lists the switch linear forwarding database (FDB) records.
`mcfdb`	Lists the switch multicast FDB records.
`trace`	Lists the trace records.
`vfinfo`	Lists the vFabrics.
`vfinfocsv`	Lists the vFabrics in CSV format.
`vfinfocsv2`	Lists the vFabrics in CSV format with enums.
`fabricinfo`	Specifies the summary of fabric devices.
`quarantine`	Lists the quarantined nodes.
`conginfo`	Lists the Congestion Info Records.
`swcongset`	Lists the Switch Congestion Settings.
`swportcong`	Lists the Switch Port Congestion Settings.
`hficongset`	Lists the HFI Congestion Settings.
`hficongcon`	Lists the HFI Congestion Control Settings.
`bfrctrl`	Lists the buffer control tables.
`cableinfo`	Lists the Cable Info records.

`portgroup`	Lists the AR Port Group records.
`portgroupfdb`	Lists the AR Port Group FWD records.

The `vfinfocsv` and `vfinfocsv2` output formats are designed to make it easier to script vfinfo queries. One line is output per vFabric of the form:

```
name:index:pkey:sl:mtu:rate:optionflag
```

The only difference between these two formats is how the MTU and rate are output. `vfinfocsv` outputs MTU and rate in human/text readable format. `vfinfocsv2` outputs MTU and rate as enumerations defined for the SMA protocol. The `opagetvf` command is based on this capability of `opasaquery`. For more information, see opagetvf on page 199.

Example

```
opasaquery -o desc -t fi
opasaquery -o portinfo -l 2
opasaquery -o sminfo
opasaquery -o pkey
```

Input Options vs. Output Permitted

The following list shows the input (assorted query by options) and outputs (`-o`) that are permitted.

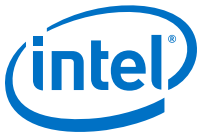
Note: In this release, the combinations displayed in **bold** are currently not available.

None

`-o` output permitted	`systemguid, nodeguid, portguid, lid, desc, path, node, portinfo, sminfo, swinfo, link, vlarb, pkey, service, mcmember, inform, linfdb, mcfdb, vfinfo, vfinfocsv, vfinfocsv2, scsc, slsc, scvlt, scnlmt, linfdb, classportinfo, fabricinfo, quarantine, conginfo, swcongset, swportcong, hficongset, hficongcon, bfrctl, cableinfo, portgroup, portgroupfdb`
`-o` output not permitted	`trace`

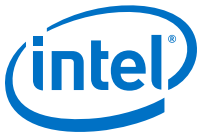
`-t` *`node_type`*

`-o` output permitted	`systemguid, nodeguid, portguid, lid, desc, node`



	-o output not permitted	portinfo, sminfo, swinfo, vlarb, pkey, service, mcmember, inform, linfdb, mcfdb, trace, vfinfo, vfinfocsv, vfinfocsv2
-l *lid*		
	-o output permitted	systemguid, nodeguid, portguid, lid, desc, path, node, portinfo, swinfo, slvl, vlarb, pkey, service, mcmember, linfdb, mcfdb
	-o output not permitted	sminfo, link, inform, trace, vfinfo, vfinfocsv, vfinfocsv2
-k *pkey*		
	-o output permitted	mcmember, path, vfinfo, vfinfocsv, vfinfocsv2
	-o output not permitted	systemimageguid, nodeguid, portguid, lid, desc, node, portinfo, sminfo, swinfo, link, vlarb, pkey, service, mcmember, inform, linfdb, mcfdb
-i *vfindex*		
	-o output permitted	vfinfo, vfinfocsv, vfinfocsv2
	-o output not permitted	systemimageguid, nodeguid, portguid, lid, desc, node, portinfo, sminfo, swinfo, link, vlarb, pkey, service, mcmember, inform, linfdb, mcfdb
-s *system_image_guid*		
	-o output permitted	systemguid, nodeguid, portguid, lid, desc, node
	-o output not permitted	portinfo, sminfo, swinfo, link, vlarb, pkey, service, mcmember, inform, linfdb, mcfdb, trace, vfinfo, vfinfocsv, vfinfocsv2

`-n node_guid`		
	-o output permitted	`systemguid, nodeguid, portguid, lid, desc, node`
	-o output not permitted	`portinfo, sminfo, swinfo, link, vlarb, pkey, service, mcmember, inform, linfdb, mcfdb, trace, vfinfo, vfinfocsv, vfinfocsv2`
`-g port_guid`		
	-o output permitted	`systemguid, nodeguid, portguid, lid, desc, path, node, service, mcmember, inform, trace`
	-o output not permitted	`portinfo, sminfo, swinfo, link, vlarb, pkey, linfdb, mcfdb, vfinfo, vfinfocsv, vfinfocsv2`
`-u port_gid`		
	-o output permitted	`path, service, mcmember, inform, trace`
	-o output not permitted	`systemguid, nodeguid, portguid, lid, desc, node, portinfo, sminfo, swinfo, link, vlarb, pkey, linfdb, mcfdb, vfinfo, vfinfocsv, vfinfocsv2`
`-m multicast_gid`		
	-o output permitted	`mcmember, vfinfo, vfinfocsv, vfinfocsv2`
	-o output not permitted	`systemguid, nodeguid, portguid, lid, desc, path, node, portinfo, sminfo, swinfo, link, vlarb, pkey, service, inform, linfdb, mcfdb, trace`
`-d name`		
	-o output permitted	`systemguid, nodeguid, portguid, lid, desc, node`
	-o output not permitted	`trace`



`-P port_guid_pair`

`-o` output permitted: `path, trace`

`-o` output not permitted: `systemguid, nodeguid, portguid, lid, desc, node, portinfo, sminfo, swinfo, link, vlarb, pkey, service, mcmember, inform, linfdb, mcfdb`

`-S serviceId`

`-o` output permitted: `path, vfinfo, vfinfocsv, vfinfocsv2`

`-o` output not permitted: `systemimageguid, nodeguid, portguid, lid, desc, node, portinfo, sminfo, swinfo, link, vlarb, pkey, service, mcmember, inform, linfdb, mcfdb`

`-L SL`

`-o` output permitted: `path, vfinfo, vfinfocsv, vfinfocsv2`

`-o` output not permitted: `systemimageguid, nodeguid, portguid, lid, desc, node, portinfo, sminfo, swinfo, link, vlarb, pkey, service, mcmember, inform, linfdb, mcfdb`

`-G gid_pair`

`-o` output permitted: `path, trace`

`-o` output not permitted: `systemguid, nodeguid, portguid, lid, desc, node, portinfo, sminfo, swinfo, link, vlarb, pkey, service, mcmember, inform, linfdb, mcfdb`

`-a port_guid_list`

`-o` output permitted: `path`

`-o` output not permitted: `systemguid, nodeguid, portguid, lid, desc, node, portinfo, sminfo, swinfo,`

		link, vlarb, pkey, service, mcmember, inform, linfdb, mcfdb, trace
-A *gid_list*	-o output permitted	path
	-o output not permitted	systemguid, nodeguid, portguid, lid, desc, node, portinfo, sminfo, swinfo, link, vlarb, pkey, service, mcmember, inform, linfdb, mcfdb, trace

3.9.4 opashowmc

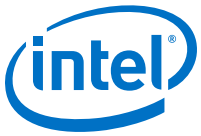
(Linux) Displays the Intel® Omni-Path Multicast groups created for the fabric along with the Intel® Omni-Path Host Fabric Interface (HFI) ports which are a member of each multicast group. This command can be helpful when attempting to analyze or debug Intel® Omni-Path multicast usage by applications or ULPs such as IPoIB.

Syntax

```
opashowmc [-v] [-t portsfile] [-p ports]
```

Options

--help	Produces full help text.
-v	Returns verbose output and shows name of each member.
-t *portsfile*	Specifies the file with list of local HFI ports used to access fabric(s) for analysis. Default is /etc/sysconfig/opa/ports file.
-p *ports*	Specifies the list of local HFI ports used to access fabric(s) for analysis. Default is first active port. The first HFI in the system is 1. The first port on an HFI is 1. Uses the format hfi:port, for example: 0:0 First active port in system. 0:y Port *y* within system. x:0 First active port on HFI *x*. x:y HFI *x*, port *y*.



Examples

```
opashowmc
opashowmc -p '1:1 1:2 2:1 2:2'
```

Environment Variables

The following environment variables are also used by this command:

PORTS	List of ports, used in absence of -t and -p.
PORTS_FILE	File containing list of ports, used in absence of -t and -p.

3.9.5 opasmaquery

(All) Performs Intel® Omni-Path Architecture-defined SMA queries and displays the resulting response. Each query is issued directly to the SMA and does not involve SM interaction.

Syntax

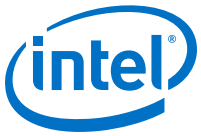
```
opasmaquery [-v] [-d detail][-g] [-l lid] [-h hfi] [-p port] [-K mkey] [-o otype]
[-m port|port1,port2] [-f flid] [-b block[,count]] [hop hop ...]
```

Options

--help	Produces full help text.
-v	Returns verbose output. Can be specified more than once for additional openib and libibumad debugging.
-d *detail*	Specifies the output detail level for cableinfo only. Range = 0 - n. Default = 2. An upper limit for detail level is not enforced. After a maximum amount of output is reached, a larger detail value has no effect.
-g	Displays line-by-line format. Default is summary format.
-l *lid*	Specifies the destination LID. Default is local port.
-h/--hfi *hfi*	Specifies the HFI, numbered 1..n. Using 0 specifies that the -p *port* port is a system-wide port number. (Default is 0.)
-p/--port *port*	Specifies the port, numbered 1..n. Using 0 specifies the first active port. (Default is 0.)
-K *mkey*	Specifies the SM management key to access remote ports.
-o *otype*	Specifies the output type. Default is nodeinfo. Refer to otype Options Vary by Report on page 182 for supported options.

Valid output types are:

`bfrctrl`	Specifies buffer control tables. `[-m dest_port] [-m port1,port2]`
`cableinfo`	Specifies cable information. `[-d detail] [-m dest_port] [-b block[,count]]`
`conginfo`	Specifies congestion information.
`desc` or `nodedesc`	Specifies node descriptions/names.
`hficongcon`	Specifies HFI congestion control settings. `[-b block[,count]] [-f flid]`
`hficonglog`	Specifies HFI congestion logs. `[-b block[,count]]`
`hficongset`	Specifies HFI congestion settings.
`linfdb`	Specifies switch linear forwarding database (FDB) tables. `[-b block[,count]] [-f flid]`
`mcfdb`	Specifies switch multicast FDB tables. `[-m dest_port] [-b block[,count]] [-f flid]`
`portgroup`	Specifies Adaptive Routing port groups. `[-b block[,count]]`
`portgroupfdb`	Specifies Adaptive Routing port group FWD tables. `[-b block[,count]] [-f flid]`
`nodeaggr`	Specifies node information and node descriptions.
`node` or `nodeinfo`	Specifies node information. `[-m dest_port]`
`portinfo`	Specifies port information.



	`[-m dest_port]`
`pstateinfo`	Specifies switch port state information. `[-m dest_port] [-m port1,port2]`
`pkey`	Specifies P-Key tables. `[-m dest_port] [-b block[,count]]`
`slsc`	Specifies SL to SC mapping tables.
`scsl`	Specifies SC to SL mapping tables.
`scsc`	Specifies SC to SC mapping tables. `[-m dest_port] [-m port1,port2]`
`scvlt`	Specifies SC to VLt tables. `[-m dest_port] [-m port1,port2]`
`scvlnt`	Specifies SC to VLnt tables. `[-m dest_port] [-m port1,port2]`
`sminfo`	Specifies SM information.
`swaggr`	Specifies node information and switch information.
`swconglog`	Specifies switch congestion logs. `[-b block[,count]]`
`swcongset`	Specifies switch congestion settings.
`swinfo`	Specifies switch information.
`swportcong`	Specifies switch congestion settings. `[-b block[,count]]`
`vlarb`	Specifies VL arbitration tables. `[-m dest_port]`
`ibnodeinfo`	Specifies IB node information.
`ledinfo`	Specifies LED information. `[-m dest_port]`

-h and -p options permit a variety of selections:

`-h 0`	First active port in system (default).
`-h 0 -p 0`	First active port in system.
`-h x`	First active port on HFI *x*.
`-h x -p 0`	First active port on HFI *x*.
`-h 0 -p y`	Port *y* within system (no matter which ports are active).
`-h x -p y`	HFI *x*, port *y*.

otype Options Vary by Report

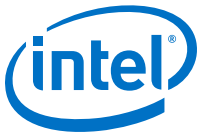
`-m port`	Specifies the port in destination device to query.
`-m port1,port2`	For some reports, specifies a range of ports between *port1* and *port2*. For others, this describes an inport/outport pair.
`-f flid`	Specifies the LID to look up in forwarding table to select which LFT or MFT block to display. Default is to show entire table.
`-b block[,count]`	Specifies the block number of either GUIDs or pkey, and the number of blocks to display. Default is to show entire table.

For example:

`-b block`	Displays all of block *block* of a larger table.
`-b block,count`	Displays *count* blocks of data starting with block *block*.
`-b, count`	Displays *count* blocks of data starting with block 0.

Examples

```
opasmaquery -o desc -l 6
# get nodedesc via lid routed

opasmaquery -o nodedesc 1 3
# get nodedesc via directed route (2 dr hops)

opasmaquery -o nodeinfo -l 2 3
# get nodeinfo via a combination of lid routed and
# directed route (1 dr hop)

opasmaquery -o portinfo
# get local port info
```



```
opasmaquery -o portinfo -l 6 -m 1
# get port info of port 1 of lid 6

opasmaquery -o pkey -l 2 3
# get pkey table entries starting (lid routed to lid 2,
# then 1 dr hop to port 3)

opasmaquery -o vlarb -l 6
# get vlarb table entries from lid 6

opasmaquery -o swinfo -l 2
# get switch info

opasmaquery -o sminfo -l 1
# get SM info

opasmaquery -o slsc -l 3
# get sl2sc table entries from lid 3

opasmaquery -o scsl -l 3
# get sc2sl table entries from lid 3
```

3.9.6 opapmaquery

(All) Performs individual PMA queries against a specific LID. It is very useful in displaying port runtime information.

Syntax

```
opapmaquery [-v] [-s sl] [-l lid] [-h hfi] [-p port]
[-o otype] [-m port] [-n mask] [-e mask] [-w mask]
```

Options

`--help` — Produces full help text.

`-v` — Specifies the verbose output. Can be specified more than once for additional openib debugging and libibumad debugging.

`-s sl` — Specifies different service level. Default is 0.

`-l lid` — Specifies the destination LID. Default is local port.

`-h/--hfi hfi` — Specifies the HFI, numbered 1..n. Using 0 specifies that the `-p port` port is a system-wide port number. (Default is 0.)

`-p/--port port` — Specifies the port, numbered 1..n. Using 0 specifies the first active port. (Default is 0.)

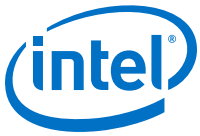
`-o otype` — Specifies the output type. Default is `getportstatus`. Refer to otype options vary by report on page 185 for supported options.

Valid output types are:

`getportstatus`	Specifies the list of port status records. Supported options: `[-m port] [-w vl mask]`
`classportinfo`	Specifies the list of port info records.
`clearportstatus`	Clears the port status. Supported options: `[-n port mask] [-e counter mask] [-w vl mask]`
`getdatacounters`	Specifies the list of data counters. Supported options: `[-n port mask] [-w vl mask]`
`geterrorcounters`	Specifies the list of error counters. Supported options: `[-n port mask] [-w vl mask]`
`geterrorinfo`	Specifies the list of error info. Supported options: `[-n port mask]`
`clearerrorinfo`	Clears the error info. Supported options: `[-n port mask] [-e counter mask]`

-h and -p options permit a variety of selections:

`-h 0`	First active port in system (default).
`-h 0 -p 0`	First active port in system.
`-h` *x*	First active port on HFI *x*.
`-h` *x* `-p 0`	First active port on HFI *x*.
`-h 0 -p` *y*	Port *y* within system (no matter which ports are active).
`-h` *x* `-p` *y*	HFI *x*, port *y*.



otype options vary by report

`-m` *`port`* Specifies the port in destination device to query/clear. Required when using `-l` option for all but `-o classportinfo`.

`-n` *`mask`* Specifies the port mask, in hexadecimal. Bits represent ports 63-0. For example: `0x2` for port 1, `0x6` for ports 1, 2.

`-e` *`mask`* Specifies the counter/error select mask, in hexadecimal. The following lists "Mask - Bit - Location for Counters". Where applicable, location "for Error Info" is presented. The default is all bits set (`0xfffffffe0`).

- 0x80000000 - 31 - Transmit Data (XmitData)

  For Error Info: Receive Error Info
- 0x40000000 - 30 - Receive Data (RcvData)

  For Error Info: Excessive Buffer Overrun
- 0x20000000 - 29 - Transmit Packets (XmitPkts)

  For Error Info: Transmit Const Error Info
- 0x10000000 - 28 - Receive Packets (RcvPkts)

  For Error Info: Receive Const Error Info
- 0x08000000 - 27 - Multicast Transmit Packets (MulticastXmitPkts)

  For Error Info: Receive Switch Relay Error Info
- 0x04000000 - 26 - Multicast Receive Packets (MulticastRcvPkts)

  For Error Info: Uncorrectable Error Info
- 0x02000000 - 25 - Transmit Wait (XmitWait)

  For Error Info: FM Configuration Error Info
- 0x01000000 - 24 - Congestion Discards (CongDiscards)
- 0x00800000 - 23 - Receive FECN (RcvFECN)
- 0x00400000 - 22 - Receive BECN (RcvBECN)
- 0x00200000 - 21 - Transmit Time Congestion (XmitTimeCong)
- 0x00100000 - 20 - Transmit Time Wasted BW (XmitWastedBW)
- 0x00080000 - 19 - Transmit Time Wait Data (XmitWaitData)
- 0x00040000 - 18 - Receive Bubble (RcvBubble)
- 0x00020000 - 17 - Mark FECN (MarkFECN)
- 0x00010000 - 16 - Receive Constraint Errors (RcvConstraintErrors)
- 0x00008000 - 15 - Receive Switch Relay (RcvSwitchRelayErrors)
- 0x00004000 - 14 - Transmit Discards (XmitDiscards)
- 0x00002000 - 13 - Transmit Constraint Errors (XmitConstraintErrors)
- 0x00001000 - 12 - Receive Remote Physical Errors (RcvRemotePhysicalErrors)
- 0x00000800 - 11 - Local Link Integrity (LocalLinkIntegrityErrors)

- 0x00000400 - 10 - Receive Errors (RcvErrors)
- 0x00000200 - 9 - Excessive Buffer Overrun (ExcessiveBufferOverruns)
- 0x00000100 - 8 - FM Configuration Errors (FMConfigErrors)
- 0x00000080 - 7 - Link Error Recovery (LinkErrorRecovery)
- 0x00000040 - 6 - Link Error Downed (LinkDowned)
- 0x00000020 - 5 - Uncorrectable Errors (UncorrectableErrors)

`-w` *`mask`* Specifies the Virtual Lane Select Mask, in hexadecimal. Bits represent VL number 31-0. For example, `0x1` for VL 0, `0x3` for VL 0,1. Default is none.

Examples

```
opapmaquery -o classportinfo

opapmaquery -o getportstatus
# get data and error counts, local port

opapmaquery -o getdatacounters -n 0x2
# get data counts, local port 1

opapmaquery -o geterrorcounters -n 0x2
# get error counts, local port 1

opapmaquery -o clearportstatus -n 0x2
# clear all counters local port 1

opapmaquery -o geterrorinfo -n 0x2
# get error info for local port 1

opapmaquery -o clearerrorinfo -n 0x2
# clear all error info, local port 1
```

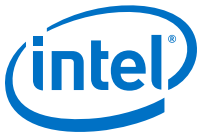
```
opapmaquery -o getdatacounters -l 6 -n 0x7e -w 0x1
# for device at LID 6, get data counters on ports 1-6, inclusive of VL 0 data

opapmaquery -o clearportstatus -l 6 -n 0x2 -e 0x1ffff
# for device at LID 6, on port 1, clear only error counters

opapmaquery -o clearerrorinfo -l 6 -n 0x2 -e 0x04000000
# for device at LID 6, on ports 1, clear uncorrectable error info
```

3.10 Basic Single Host Operations

The tools described in this section are available on each host where the Intel® Omni-Path Fabric Host Software stack tools have been installed. The tools enable FastFabric toolset operations against cluster nodes, however, they can also be directly used on an individual host.

3.10.1 opaconfig

(Switch and Host) Configures the Intel® Omni-Path Fabric Suite FastFabric.



Syntax

```
opaconfig [-r root] [-v|-vv] [-u|-s|-e comp] [-E comp] [-D comp]
[--user_queries|--no_user_queries] [--answer keyword=value]
```

or

```
opaconfig -C
```

or

```
opaconfig -V
```

Options

No option	Starts the Intel® Omni-Path Software TUI.
`--help`	Produces full help text.
`-r` *root*	Specifies alternate root directory; default is `/`.
`-v`	Specifies verbose logging.
`-vv`	Specifies very verbose debug logging.
`-u`	Uninstalls all ULPs and drivers with default options.
`-s`	Enables autostart for all installed drivers.
`-e` *comp*	Uninstalls the given component with default options. This option can appear more than once on the command line.
`-E` *comp*	Enables autostart of a given component. This option can appear with `-D` or more than once on the command line.
`-D` *comp*	Disables autostart of given component. This option can appear with `-E` or more than once on the command line.
`-C`	Outputs list of supported components. Supported components include: `opa_stack ibacm mpi_selector intel_hfi oftools opa_stack_dev fastfabric delta_ipoib opafm mvapich2 openmpi gasnet openshmem mvapich2_gcc_hfi mvapich2_intel_hfi openmpi_gcc_hfi openmpi_intel_hfi delta_mpisrc delta_debug` Supported component name aliases include: `opa ipoib mpi mpisrc opadev`
`-V`	Outputs version.

`--user_queries`	Permits non-root users to query the fabric (default).
`--no_user_queries`	Prohibits non-root users from querying the fabric.
`--answer` *keyword=value*	Provides an answer to a question that may occur during the operation. Answers to questions not asked are ignored. Invalid answers result in prompting for interactive installs, or using default options for non-interactive installs.

Possible Questions (*keyword=value*):

`UserQueries`	Allow non-root users to access the UMAD interface?
	Note: Allowing access to UMAD device files may present a security risk. However, this allows tools such as `opasaquery` and `opaportinfo` to be used by non-root users.
`IrqBalance`	Set IrqBalance to Exact?

Example

```
# opaconfig
Intel OPA x.x.x.x.x Software

   1) Show Installed Software
   2) Reconfigure OFED IP over IB
   3) Reconfigure Driver Autostart
   4) Generate Supporting Information for Problem Report
   5) FastFabric (Host/Chassis/Switch Setup/Admin)
   6) Uninstall Software

   X) Exit
```

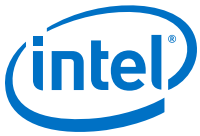
3.10.2 opacapture

(Host) Captures critical system information into a zipped tar file. The resulting tar file should be sent to Customer Support along with any Intel® Omni-Path Fabric problem report regarding this system.

Note: The resulting host capture file can require significant amounts of space on the host. The actual size varies, but sizes can be multiple megabytes. Intel recommends ensuring that adequate disk space is available on the host system.

Syntax

```
opacapture [-d detail] output_tgz_file
```



Options

`--help` Produces full help text.

`-d detail` Captures level of detail:

`1` (Local)	Obtains local information from host. Default if no options are entered.
`2` (Fabric)	In addition to *Local*, also obtains basic fabric information by queries to the SM and fabric error analysis using `opareport`.
`3` (Fabric +FDB)	In addition to *Fabric*, also obtains the Forwarding Database (FDB), which includes the switch forwarding tables from the SM and the server multicast membership.
`4` (Analysis)	In addition to *Fabric+FDB*, also obtains `opaallanalysis` results. If `opaallanalysis` has not yet been run, it is run as part of the capture.

Note: Detail levels 2 – 4 can be used when fabric operational problems occur. If the problem is node-specific, detail level 1 should be sufficient. Detail levels 2 – 4 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 – 4, the additional information is only available on a node with Intel® Omni-Path Fabric Suite FastFabric Toolset installed. The information is gathered for every fabric specified in the `/etc/sysconfig/opa/ports` file.

`output_tgz_file` Specifies the name of a file to be created by `opacapture`. The file name specified is overwritten if it already exists. Intel recommends using the `.tgz` suffix in the file name supplied. If the filename given does not have a `.tgz` suffix, the `.tgz` suffix is added.

Examples

```
opacapture mycapture.tgz
opacapture -d 3 030127capture.tgz
```

3.10.3 opahfirev

(Linux) Scans the system and reports hardware and firmware information about all the HFIs in the system.

Syntax

```
opahfirev
```

Options

no option Returns information about all of the HFIs in the system.

`--help` Produces full help text.

Example

```
# opahfirev
######################
server.intel.com  - HFI 02:00.0
Board: ChipABI 3.0, WFR_ID 0x1, ChipRev 7.0 patch 0x00000003, SW Compat 3
SN:    0x0057501a
Bus:   PCIe,8000MHz,x16
GUID:  0011:7501:0157:501a
######################
```

3.10.4 opainfo

Provides summary information for local HFI port(s).

Syntax

```
opainfo [-h hfi] [-p port] [-o type] [-g] [-d detail] [-v [-v]...]
```

Options

`--help` Produces full help text.

`-h` *`hfi`* Specifies the HFI, numbered 1..n. Using 0 specifies that the `-p` *`port`* port is a system-wide port number. (Default is 0.)

`-p` *`port`* Specifies the port, numbered 1 to n. Using 0 specifies the first active port across all HFIs/ports. Default = 1 which indicates all ports. If selected, the tool returns information on all available ports if `p` is not defined.

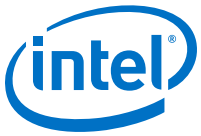
`-o` *`type`* Specifies the output type and can appear more than once.

Note: Behavior without -o gives a brief summary of portinfo, counters and cableinfo.

`info` Outputs detailed portinfo.

`stats` Outputs detailed port counters.

`-g` Output is displayed in line-by-line format. Default = summary format.



`-d detail` Output detail level. Range = 0 - 4. CableInfo only. Default = 0.

Note: -d option is ignored when used with -o type.

0 Minimal crucial information (for example, cable length, vendor)

1 Brief summary

2 Extended brief summary

3 Verbose output

4 Verbose and debug output. Lists all settings both statically and dynamically configured in the cable.

-h and -p options permit a variety of selections:

`-h 0` First active port in system (default).

`-h 0 -p 0` First active port in system.

`-h x` First active port on HFI *x*.

`-h x -p 0` First active port on HFI *x*.

`-h 0 -p y` Port *y* within system (no matter which ports are active).

`-h x -p y` HFI *x*, port *y*.

Debug Options

`-v` Specifies the verbose output. Additional invocations (–v –v ...) turn on debugging, `openib` debugging, and `libibumad` debugging.

Examples

```
opainfo
   hfi1_0:1                  PortGID:0xfe80000000000000:001175010165b19c
   PortState:     Active
   LinkSpeed      Act: 25Gb         En: 25Gb
   LinkWidth      Act: 4            En: 4
   LinkWidthDnGrd ActTx: 4  Rx: 4   En: 3,4
   LCRC           Act: 14-bit       En: 14-bit,16-bit,48-bit       Mgmt: True
   LID: 0x00000001-0x00000001       SM LID: 0x00000002 SL: 0
   QSFP: PassiveCu, 1m   FCI Electronics   P/N 10131941-2010LF   Rev 5
   Xmit Data:           22581581 MB Pkts:           5100825193
   Recv Data:           18725619 MB Pkts:           4024569756
   Link Quality: 5 (Excellent)
```

3.10.5 opaportconfig

(Host or Switch) Controls the configuration and state of a specified Intel® Omni-Path Host Fabric Interface (HFI) port on the local host or a remote switch.

Syntax

```
opaportconfig [-l lid [-m dest_port]] [-h hfi] [-p port] [-r secs] [-z]
[-S state] [-P physstate] [-s speed] [-w width] [-c LTPCRC]  [-K mkey]
[-v] [-x] [-L lid] [<sub command>]
```

Options

`--help` Produces full help text.

`-l` *lid* Specifies the destination LID. Default is local port.

`-m` *dest_port* Specifies the destination port. Default is port with given LID. Used to access switch ports.

`-h` *hfi* Specifies the HFI to send through/to. Default is first HFI.

`-p` *port* Specifies the port to send through/to. Default is first port.

`-K` *mkey* Specifies the SM management key to access remote ports.

sub command Specifies the one of the following choices:

`enable` Enables port.

`disable` Disables port.

`bounce` Bounces port.

Note: Bouncing remote ports may cause timeouts.

`ledon` Turns port LED on.

`ledoff` Turns port LED off.

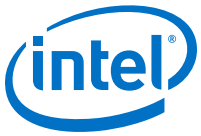
Configuration Options

`-r` *secs* Repeats to keep the port down for the specified amount of seconds.

`-S` *state* Specifies the new state. Default is 0.

`0` No-op.

`1` Down.



`2` Initiate.

`3` Armed.

`4` Active.

`-P` *`physstate`* Specifies the new physical state. Default is 0.

Note: All transitions are valid.

`0` No-op.

`2` Polling.

`3` Disabled.

`11` Phy-Test. Current physstate must be `disabled`.

`-s` *`speed`* Specifies the new link speeds enabled. Default is 0. To enable multiple speeds, use the sum of the desired speeds.

`0` No-op.

`2` 0x0002 - 25 Gb/s.

`-w` *`width`* Specifies the new link widths enabled. Default is 0. To enable multiple widths, use sum of desired widths.

`0` No-op.

`1` 0x01 - 1x.

`2` 0x02 - 2x.

`4` 0x04 - 3x.

`8` 0x08 - 4x.

`-c` *`LTPCRC`* Specifies the new LTP CRCs enabled. Default is 0. To enable multiple LTP CRCs, use sum of desired LTP CRCs.

`0` No-op.

`1` 0x1 - 14-bit LTP CRC mode.

`2` 0x2 - 16-bit LTP CRC mode.

`4` 0x4 - 48-bit LTP CRC mode.

`8` 0x8 - 12/16 bits per lane LTP CRC mode.

-h and -p options permit a variety of selections:

`-h 0` First active port in system (default).

`-h 0 -p 0` First active port in system.

`-h x` First active port on HFI *x*.

`-h x -p 0` First active port on HFI *x*.

`-h 0 -p y` Port *y* within system (no matter which ports are active).

`-h x -p y` HFI *x*, port *y*.

Debug Options

`-v` Verbose output. Additional invocations turn on debugging, `openib` debugging, and `libibumad` debugging.

`-z` Does not get port information first; clears most port attributes.

`-L lid` Sets PortInfo.LID = `lid`.

Examples

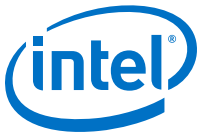
```
opaportconfig -w 1
opaportconfig -p 1 -h 2 -w 3
```

Description

Port configuration is transient in nature. If the given host is rebooted or its Intel® Omni-Path Fabric Stack is restarted, the port reverts to its default configuration and state. Typically, the default state is to have the port enabled with all speeds and widths supported by the given HFI port.

To access switch ports using this command, the `-l` and `-m` options must be given. The `-l` option specifies the lid of switch port 0 (the logical management port for the switch) and `-m` specifies the actual switch port to access. If SMA mkeys are used, the `-K` option is also needed. However, the Intel® Omni-Path Fabric Suite Fabric Manager does not use SMA mkeys by default, therefore this option may not be required.

Note: The `/etc/init.d/opaportconfig` script is provided as an example of changing port speed every time the server boots. This script can be edited, then scheduled, using `chkconfig` to control link settings on any set of HFI ports.



Caution: When using this command to disable or reconfigure switch ports, if the final port in the path between the Fabric Management Node and the switch is disabled or fails to come online, then `opaenableports` is not able to reenable it. In this case, the switch CLI and/or a switch reboot may be needed to correct the situation.

3.10.6 opaportinfo

(Host or Switch) Displays configuration and state of a specified Intel® Omni-Path Host Fabric Interface (HFI) port on the local host or a remote switch.

Syntax

```
opaportinfo [-l lid [-m dest_port]] [-h hfi] [-p port] [-K mkey] [-v]
```

Options

`-l` *`lid`* Specifies the destination LID. Default is local port.

`-m` *`dest_port`* Specifies the destination port. Default is port with given LID. Useful to access switch ports.

`-h` *`hfi`* Specifies the HFI to send through/to. Default is first HFI.

`-p` *`port`* Specifies the port to send through/to. Default is first port.

`-K` *`mkey`* Specifies the SM management key to access remote ports.

-h and -p options permit a variety of selections:

`-h 0` First active port in system (default).

`-h 0 -p 0` First active port in system.

`-h` *`x`* First active port on HFI *x*.

`-h` *`x`* `-p 0` First active port on HFI *x*.

`-h 0 -p` *`y`* Port *y* within system (no matter which ports are active).

`-h` *`x`* `-p` *`y`* HFI *x*, port *y*.

Debug Options

`-v` Specifies the verbose output. Additional invocations (–v –v ...) turn on debugging, `openib` debugging, and `libibumad` debugging.

Examples

```
opaportinfo -p 1
opaportinfo -p 2 -h 2 -l 5 -m 18
```

Description

To access switch ports using this command, the -l and -m options must be given. The -l option specifies the LID of switch port 0 (the logical management port for the switch) and -m specifies the actual switch port to access. If SMA mkeys are used, the -K option is also needed. However, the Intel® Omni-Path Fabric Suite Fabric Manager does not use SMA mkeys by default, therefore this option may not be required.

3.10.7 opapacketcapture

Starts capturing packet data.

To stop capture and trigger dump, use SIGINT or SIGUSR1. Program dumps packets to file and exits.

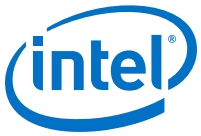
Note: Using opapacketcapture with large amounts of traffic can cause performance issues on the given host. Intel recommends you use opapacketcapture on hosts with lower packet rates and bandwidth.

Syntax

```
opapacketcapture [-o outfile] [-d devfile] [-f filterfile] [-t triggerfile]
[-l triggerlag][-a alarm] [-p packets] [-s maxblocks] [-v [-v]]
```

Options

--help	Produces full help text.
-o *outfile*	Specifies the output file for captured packets. Default = packetDump.pcap
-d *devfile*	Specifies the device file for capturing packets. Default = /dev/hfi1_diagpkt0
-f *filterfile*	Specifies the file used for filtering. If absent, no filtering is done.
-t *triggerfile*	Specifies the file used for triggering a stop capture. If absent, normal triggering is performed.
-l *triggerlag*	Specifies the number of packets to collect after trigger condition is met, before dumping data and exiting. Default = 10.
-a *alarm*	Specifies the number of seconds for alarm trigger to dump capture and exit.
-p *packets*	Specifies the number of packets for alarm trigger to dump capture and exit.
-s *maxblocks*	Specifies the number of blocks to allocate for ring buffer. Value is in Millions. Default = 2 which corresponds to 128 MiB because 1 block = 64 Bytes.



`-v` Produces verbose output. (Use verbose Level 1+ to show levels.)

Example .

```
# opapacketcapture
opapacketcapture: Capturing packets using 128 MiB buffer
^C
opapacketcapture: Triggered
Number of packets stored is 100
```

In the example above, `opapacketcapture` operates until **CTRL+C** is entered.

3.10.8 opa-arptbl-tuneup

Adjusts kernel ARP/neighbor table sizes for very large subnets based on configured IPv4/IPv6 network interface netmask values. Normally executes once on boot by `opa.service`; however, `opa-arptbl-tuneup` can be invoked with user discretion for a changed subnet configuration.

Note: Must execute as `root`.

Syntax

```
opa-arptbl-tuneup [start | stop | restart | force-reload | status]
```

Options

`--help`	Produces full help text.
`start`	Adjusts kernel ARP table size.
`stop`	Restores previous configuration.
`restart`	Stops then starts.
`force-reload`	Stops then starts. (Identical to `restart` option.)
`status`	Checks if original table size was changed.

3.10.9 opa-init-kernel

Initializes the OPA extensions to the RDMA stack. This script is typically run by the system at boot time and is not intended to be run by hand.

Syntax

```
opa-init-kernel [--help]
```

Option

`--help` Produces full help text.

3.10.10 opatmmtool

(Host) Manages and updates the firmware on the Thermal Management Microchip (TMM).

Syntax

```
opatmmtool [-v] [-h hfi] [-f file] operation
```

Options

`--h` Produces full help text.

`-v` Produces verbose output.

`-h hfi` Specifies the HFI, numbered 1..n. Default = 1.

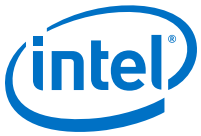
`-f file` Specifies the firmware file or output file.

operation Includes one of the following:

`reboot`	Reboots the TMM.
`fwversion`	Reports the current firmware version.
`fileversion`	Reports the file version. Requires `-f fw_file` option.
`update`	Performs a firmware update. Requires `-f fw_file` option.
`dumpotp`	Dumps the one-time programmable (OTP) region. Requires `-f output_file` option.
`lockotp`	Locks the one-time programmable (OTP) region.
`status`	Displays the current GPIO pin status.

Examples

```
opatmmtool -h 1 fwversion
opatmmtool -v -f main.signed.bin update
opatmmtool reboot
```



3.11 FastFabric Utilities

The CLIs described in this section are used for miscellaneous information about the fabric. They are also available for custom scripting.

3.11.1 opagetvf

Used for scripting application use of vFabrics, such as for mpirun parameters. You can query by VF Name, VF Index, Service ID, MGID, PKey, or SL. Fetches the Virtual Fabric info in a delimited format. Returns exactly one matching VF. When multiple VFs match the query, it prefers non-default VFs in which the calling server is a full member. If multiple choices remain, it returns the one with the lowest VF Index. Uses the same algorithm as the Distributed SA Plug-in (DSAP).

The tool can be used with additional scripts to help set PKey, SL, MTU, and Rate when running MPI jobs. Internally, this tool is based on the `opasaquery -o vfinfocsv` command. For more information, see opasaquery on page 170.

Syntax

```
opagetvf [-h hfi] [-p port] [-e] [-d vfname | -S serviceId | -m mcgid |
-i vfIndex | -k pkey | -L sl]
```

Options

`--help`	Produces full help text.
`-h` *hfi*	Specifies the HFI to send by. Default is first HFI.
`-p` *port*	Specifies the port to send by. Default is first active port.
`-e`	Outputs MTU and rate as enum values.
`-d` *vfname*	Queries by VirtualFabric Name.
`-S` *serviceId*	Queries by Application ServiceId.
`-m` *gid*	Queries by Application Multicast GID.
`-i` *vfindex*	Queries by VirtualFabric Index.
`-k` *pkey*	Queries by VirtualFabric PKey.
`-L` *SL*	Queries by VirtualFabric SL.

Examples

```
opagetvf -d 'Compute'
opagetvf -h 2 -p 2 -d 'Compute'
```

Sample Outputs

The output is of the form: `name:index:pkey:sl:mtu:rate:optionflag` as shown in the following example.

Option flag (bitmask) values include:

- `0x00` Indicates no bits are set. Specifically, no QoS, no Security, and no flow control disabled (which means flow control is enabled).
- `0x01` Security
- `0x02` QoS
- `0x04` Flow Control Disable

```
# opagetvf -d Default
Default:0:0xffff:0:unlimited:unlimited:0x0
```

3.11.2 opagetvf_env

Provides `opagetvf_func` and `opagetvf2_func` shell functions that query the parameters of a vFabric. Also exports values that indicate the PKEY, SL, MTU, and RATE associated with the vFabric. The typical usage of this tool is to include it in a shell script as:

```
. /usr/sbin/opagetvf_env
```

A usage example is provided in: `/usr/lib/opa/src/mpi_apps/openmpi.params`

Note: `opagetvf_func` and `opagetvf2_func` have a similar usage. The difference is whether the MTU and RATE are returned as absolute values or enum values, respectively.

Function Syntax

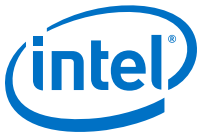
```
opagetvf_func "arguments to opagetvf" pkey_env_var_name sl_env_var_name
[mtu_env_var_name [rate_env_var_name]]
```

or

```
opagetvf2_func "arguments to opagetvf" pkey_env_var_name
sl_env_var_name [mtu_env_var_name [rate_env_var_name]]
```

Function Options

`"arguments to opagetvf"`	Specifies a set of arguments to pass to `opagetvf` to select a virtual fabric. See `opagetvf` on page 199 for more information.
`pkey_env_var_name`	Specifies the environment variable to fill in with pkey for the selected virtual fabric. The variable given will be exported with the hex numeric value for the pkey.



If a variable name of "" is provided, pkey is not saved.

`sl_env_var_name` — Specifies the environment variable to fill in with service level (sl) for the selected virtual fabric. The variable given will be exported with the numeric value for the sl.

If a variable name of "" is provided, sl is not saved.

`mtu_env_var_name` — Specifies the environment variable to fill in with maximum MTU for the selected virtual fabric. The variable given will be exported with the value for the MTU.

If a variable name of "" is provided, MTU is not saved.

For `opagetvf_func`, MTU is returned as an absolute value of 2048, 4096, 8192, or 10240.

For `opagetvf2_func`, MTU is returned as an enumerated value of 4, 5, 6, or 7 corresponding to the absolute values above, respectively.

If the selected virtual fabric does not have a limitation specified for MTU, the variable will be unaltered.

`rate_env_var_name` — Specifies the environment variable to fill in with maximum static rate for the selected virtual fabric. The variable given will be exported with the value for the rate.

If a variable name of "" is provided, rate is not saved.

For `opagetvf_func`, rate is returned as an absolute value of 25g, 50g, 75g or 100g.

For `opagetvf2_func`, rate is returned as an enumerated value of 15, 12, 9, or 16 corresponding to the absolute values above, respectively.

If the selected virtual fabric does not have a limitation specified for rate, the variable will be unaltered.

Function Example

```
. /usr/sbin/opagetvf_env

# ensure values are empty in case they are not specified for the virtual fabric
MTU=
RATE=
opagetvf_func "-d 'Compute'" PKEY SERVICE_LEVEL MTU RATE
echo "The Compute Virtual Fabric has pkey: $PKEY SL:$SERVICE_LEVEL MTU: $MTU
rate:$RATE"
```

Note: Additional examples may be found in `/usr/lib/opa/src/mpi_apps/openmpi.params` and `/usr/lib/opa/src/mpi_apps/mvapich2.params`. Those scripts use `opagetvf_func` and `opagetvf2_func` to get virtual fabric parameters and then pass them into `openmpi` and `mvapich2`, respectively.

3.11.3 opaexpandfile

(Linux) Expands a Intel® Omni-Path Fabric Suite FastFabric hosts, chassis, 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

```
opaexpandfile file
```

Options

`--help` Produces full help text.

`file` Specifies the FastFabric file to be processed.

Example

```
opaexpandfile allhosts
```

3.11.4 opafirmware

Returns firmware information.

Syntax

```
opafirmware [--showVersion | --showType] [firmwareFile]
```

Options

`--help` Produces full help text.

`--showVersion` Specifies the version of the firmware file.

`--showType` Specifies the type of the firmware file.

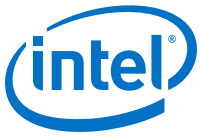
firmwareFile Specifies the firmware filename.

Examples

```
# opafirmware --showVersion STL1.q7.10.0.0.0.spkg
10.0.0.0
# opafirmware --showType STL1.q7.10.0.0.0.spkg
Omni_Path_Switch_Products.q7
```

3.11.5 oparesolvehfiport

(Host) Permits the Intel® Omni-Path Fabric Host Software style Intel® Omni-Path Host Fabric Interface (HFI) number and port number arguments to be converted to a Host Software style HFI name and physical port number.



Syntax

```
oparesolvehfiport [-o output] [hfi] [port]
```

Options

`--help` Produces full help text.

`-o` *output* Specifies the output type.

`devname` Prints the device name, in the format hfiname:portnum (default).

`hfinum` Prints the hfi number.

hfi Specifies the HFI, numbered 1..n. Using 0 specifies that the -p port port is a system-wide port number. (Default is 0.)

port Specifies the port, numbered 1..n. Using 0 specifies the first active port. (Default is 0.)

The HFI and port permit a variety of selections:

`0 0` First active port in system.

x `0` First active port on HFI *x*.

`0` *y* Port *y* within system (no matter which ports are active).

x y HFI *x*, port *y*

Examples

```
oparesolvehfiport 0 1                #Output: hfi1_0:1
oparesolvehfiport -o devname 0 1     #Output: hfi1_0:1
oparesolvehfiport -o hfinum 0 1      #Output: 1
```

3.11.6 opasorthosts

Sorts its standard input in a typical host name order and sorts to standard output. 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 or cable tests that place hosts in order by name.

Syntax

```
opasorthosts <hostlist> output_file
```

Options

`--help`	Produces full help text.
hostlist	Specifies the list of host names.
output_file	Specifies the sorted list output.

```
opasorthosts < host.xml > Sorted_host
```

Standard Input

```
opasorthosts
osd04
osd1
compute20
compute3
mgmt1
mgmt2
login
```

Standard Output

```
compute3
compute20
login
mgmt1
mgmt2
osd1
osd04
```

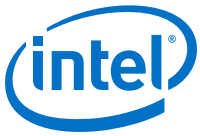
3.11.7 opaxmlextract

(Linux) Extracts element values from XML input and outputs the data in CSV format. `opaxmlextract` is intended to be used with `opareport`, 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. `opaxmlextract` 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 `opareport` with a call to `opaxmlextract` with commonly used parameters.

Syntax

```
opaxmlextract [-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 the 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 FIs.Node:id` can be used. This will return the value of the id attribute of any Node elements within FIs element. • If desired, a specific element can be selected by its attribute value, such as `-e MulticastFDB.Value:LID:0xc000` which will return the value of the Value element within Multicast FDB element where the Value element has an attribute of LID with a value of `0xc000`. • A given element can be specified multiple times each with a different AttrName or attrValue.
`-s/--suppress` *suppress_element*	Specifies the name of the 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 opareport output filtered by opaxmlextract:

```
# opareport -o comps -s -x | opaxmlextract -d \; -e NodeDesc
 -e SystemImageGUID -e NumPorts -s Neighbor
Getting All Node Records...
Done Getting All Node Records
Done Getting All Link Records
Done Getting All Cable Info Records
Done Getting All SM Info Records
Getting All Port Counters...
Done Getting All Port Counters
NodeDesc;SystemImageGUID;NumPorts
phs1fnivd13u07n4 hfi1_0;0x00117501016033c7;1
phs1fnivd13u07n2 hfi1_0;0x00117501016033ef;1
phs1fnivd13u07n1 hfi1_0;0x001175010160347a;1
phs1fnivd13u07n3 hfi1_0;0x0011750101603593;1
phs1swivd13u21;0x00117501ff6a5619;48
phs1fnivd13u07n1 hfi1_0;;
```

Details

opaxmlextract 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.

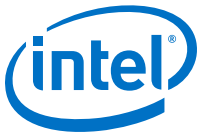
opaxmlextract takes the XML input stream from either stdin or a specified input file. opaxmlextract does not use or require a connection to a fabric.

opaxmlextract 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 which 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 which 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 NodeGUID inside a Neighbor specification of opareport.

opaxmlextract 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 opaxmlextract to produce multiple records.

opaxmlextract 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.

`opaxmlextract` 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, `opaxmlextract`, 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 `opaxmlextract` 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.

3.11.8 opaxmlfilter

Processes an XML file and removes all specified XML tags. The remaining tags are output and indentation can also be reformatted. `opaxmlfilter` is the opposite of `opaxmlextract`.

Syntax

```
opaxmlfilter [-t|-k] [-l] [-i indent] [-s element] [-P param_file] [input_file]
```

Options

`--help`	Produces full help text.
`-t`	Trims leading and trailing whitespace in tag contents.
`-k`	In tags with purely whitespace that contain newlines, keeps newlines as-is. Default is to format as an empty list.
`-l`	Adds comments with line numbers after each end tag. This can make 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 (in any order).
`-P` *param_file*	Uses input command line options (parameters) from `param_file`.
input_file	Specifies the XML file to read. Default is `stdin`.

3.11.9 opaxmlindent

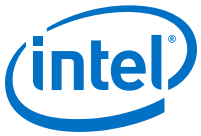
(Linux) Takes well-formed XML as input, filters out comments, and generates a uniformly-indented equivalent XML file. Use `opaxmlindent` to reformat files for easier reading and review, also to reformat a file for easy comparison with diff.

Syntax

```
opaxmlindent [-t|-k] [-i indent] [input_file]
```

Options

`--help`	Produces full help text.
`-t`	Trims leading and trailing whitespace in tag contents.
`-k`	In tags with purely whitespace that contain newlines, keeps newlines as-is. 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`.



3.11.10 opaxmlgenerate

(Linux) Takes comma-separated-values (CSV) data 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

```
opaxmlgenerate [-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 value for *element* 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*	Name of the XML element that is the enclosing header start tag.
`-e/--end` *element*	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*	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

`opaxmlgenerate` 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. `opaxmlgenerate` does not use nor require a connection to an Intel® Omni-Path Fabric.

`opaxmlgenerate` 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 (3) 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. `opaxmlgenerate` does **not** check for matching start and end tags or proper nesting of tags.

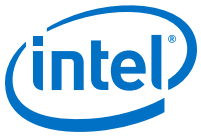
Options (parameters) to `opaxmlgenerate` can be specified on the command line, with a parameter file, or both. A parameter file is specified with `-P` *`param_file`*. When a parameter file specification is encountered on the command line, option processing on the command line is suspended, the parameter 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 opaxmlgenerate to Create Topology Input Files

`opaxmlgenerate` can be used to create scripts to translate from user-specific format into the `opareport topology_input` file format. `opaxmlgenerate` 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 which translates from a user-specific format into `opareport topology_input` would be:

- As needed, reorganize the data into link and node data CSV files, in a sequencing similar to that used by `opareport topology_input`. One link record per line in one temporary file, one node record per line in another temporary file and one SM per line in a third temporary file.
- The script must directly output the boilerplate for XML version, etc.
- `opaxmlgenerate` can be used to output the Link section of the `topology_input`, using the link record temporary file.
- `opaxmlgenerate` 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 HFIs, Switches, and Routers.
- `opaxmlgenerate` can be used to output the SM section of the `topology_input`, if desired.
- The script must directly output the closing XML tags to complete the `topology_input` file.



3.11.11 opacheckload

Returns load information on hosts in the fabric.

Syntax

```
opacheckload [-f hostfile] [-h 'hosts'] [-r] [-a|-n numprocs] [-d uploaddir]
```

Options

`--help`	Produces full help text.
`-f` *hostfile*	Specifies the file with hosts to check. Default = `/etc/sysconfig/opa/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*	Shows the specified number of top *numprocs* hosts. Default is 10.
`-a`	Shows all hosts. Default is 10.
`-d` *upload_dir*	Specifies the target directory to upload `loadavg`. Default is `uploads`.

Examples

```
opacheckload
opacheckload -h 'arwen elrond'
HOSTS='arwen elrond' opacheckload
```

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.

3.12 Address Resolution Tools

These tools allow you to verifiy and diagnose the `ibacm` distributed SA plug-in.

3.12.1 opa_osd_dump

Prints the current contents of the distributed SA shared memory database.

Syntax

```
opa_osd_dump [--verbose arg | -v arg]
```

Options

`--help`	Produces full help text.
`--verbose/-v arg`	Specifies the Kernel logging level to perform. Range = 1 - 7.

Example

```
opa_osd_dump >opasadb_contents
```

3.12.2 opa_osd_exercise

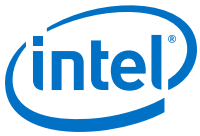
Performs stress test on SM and distributed SA query system.

Syntax

```
opa_osd_exercise [-d | -s | -r | -x | -X | -D | -p | -S | -t |
-e] guidlist
```

Options

`--help`	Produces full help text.
`-d debug level`	Sets debugging level.
`-s seconds`	Specifies running for at least *seconds* seconds.
`-r remote`	Specifies the host running the fabric simulator.
`-x count`	Number of destinations to toggle up or down after each pass. Maximum = MAX_TOGGLES.
`-X count`	Specifies how often to toggle a source port up or down (in seconds).
`-D seconds`	Specifies how long to sleep after each pass. This value gives the Subnet Manager time to process port events.
`-p pkey`	Specifies to include *pkey* in the searches. Can be specified up to 8 times.
`-S sid`	Specifies to include SID in the searches.



`-t error threshold`	Cancels the test if the number of path errors to a single destination exceeds *`error threshold`*. The count is reset to zero when a correct result is retrieved. Can be specified up to 8 times. Note that providing both SIDs and pkeys may cause problems.
`-e`	Instructs simulator to enable all ports before starting.
`guidlist`	Text file that lists the source and destination GUIDs and LIDs. *`guidlist`* format is: • `lid_0;guid_0;node_desc_0` • `lid_1;guid_1;node_desc_1` • and so on.

Example

```
opa_osd_exercise -p 0x9001 guidtable
```

3.12.3 opa_osd_perf

Tests the performance of the distributed SA shared memory database.

Syntax

```
opa_osd_perf [-q | -p | -S] guidtable
```

Options

`--help`	Produces full help text.
`-q queries`	Runs at least the specified number of queries.
`-p pkey`	Specifies to include *`pkey`* in the searches. Can be specified up to 8 times.
`-S sid`	Specifies to include SID in the searches. Can be specified up to 8 times. Note that providing both SIDs and pkeys may cause problems.
`guidtable`	Text file that lists the destination GUIDs and LIDs. For example, from a `build_table.pl` file.

Example

```
opa_osd_perf -q 100000 -p 0x8001  guidtable
```

3.12.4 opa_osd_query

Queries the opasadb for path records. This tool allows you to create an arbitrary path query and view the result.

Syntax

```
opa_osd_query [-v | -verbose] | [-s | --slid] | [-d | --dlid] |
[-S | --sgid] | [-D | --dgid]| [-k | --pkey] | [-i | --sid] |
[-h | --hfi] | [-p | --port]
```

Options

All arguments are optional, but ill-formed queries can be expected to fail. You must provide at least a pair of LIDs or a pair of GIDs.

If you have multiple HFIs, the same LID can appear on more than one HFI, therefore you must specify which HFI to use when searching by LIDs.

Numbers can be in decimal, hex, or octal.

`--help`	Produces full help text.
`-v/--verbose` *arg*	Sets debugging level. Range = 1 - 7.
`-s/--slid` *arg*	Specifies source LID.
`-d/--dlid` *arg*	Specifies destination LID.
`-S/--sgid` *arg*	Specifies source GID in GID format (`0x00000000:0x00000000`) or in Inet6 format (`x:x:x:x:x:x:x:x`).
`-D/--dgid` *arg*	Specifies destination GID in GID format (`0x00000000:0x00000000`) or in Inet6 format (`x:x:x:x:x:x:x:x`).
`-k/--pkey` *arg*	Specifies partition key.
`-i/--sid` *arg*	Specifies service ID.
`-h/--hfi` *arg*	Specifies the HFI to use. Default = first HFI. The HFI can be identified by name, for example, `hfi1_0` or by number, for example, `1, 2, 3, ....`
`-p/--port` *arg*	Specifies the port to use. Default = first port.

Example

```
opa_osd_query -s2 -d4
```

4.0 Sample Files

This section describes the files that are installed in the `/usr/lib/opa/samples` directory, including the `opagentopology` sample script.

4.1 List of Files

This section describes the files that are installed in the `/usr/lib/opa/samples` directory.

Configuration and Control Files

Files used by commands that analyze the fabric and perform multi-step initialization and verification operations. See Configuration and Control for Chassis, Switch, and Host for command details.

- `allhosts-sample`: all hosts in fabric, including management nodes. See `opahostadmin` for details.
- `chassis-sample`: all internally managed switches (reachable out-of-band). See `opagenchassis` and `opachassisadmin` for details.
- `esm_chassis-sample`: all internally managed switches running the embedded FM. See `opagenesmchassis` for details.
- `hosts-sample`: all hosts in the fabric. See `opahostadmin` for details.
- `ports-sample`: HFI and port configuration to use for communication with fabric. See `opagenswitches`, `opagenchassis`, `opagenesmchassis` and `opaswitchadmin` for details.
- `switches-sample`: all externally managed switches (reachable in-band). See `opagenswitches` and `opaswitchadmin` for details.

Packet Capture Files

Files for use with `opapacketcapture`:

- `filterFile.txt` - packet filter configuration.
- `triggerFile.txt` - trigger configuration for condition to terminate packet capture.

Topology Files

Files related to topology:

- `README.topology`
- `README.xlat_topology`
- `opagentopology` - script to generate topology file. See `opagentopology` for details.

- opatopology_links.txt - text CSV values for LinkSummary information.
- opatopology_FIs.txt - text CSV values for HFI Nodes information.
- opatopology_SWs.txt - text CSV values for Switch Nodes information.
- opatopology_SMs.txt - text CSV values for SM information.
- linksum_swd06.csv, linksum_swd24.csv - sample CSV configurations. See README.xlat_topology for explanation.
- topology.xlsx, topology_cust.xlsx - topology MS Excel files.
- opamon.conf-sample, opamon.si.conf-sample - port counter threshold files for use with opareport.

Miscellaneous Files

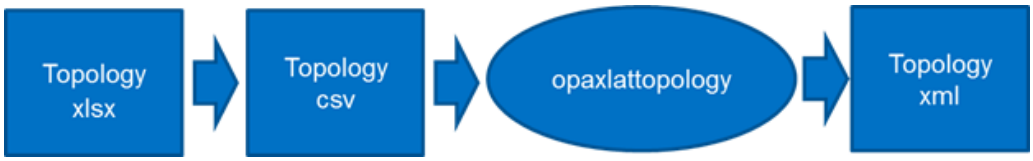
- hostverify.sh - bash script to help verify configuration and performance of host nodes.
- mac_to_dhcp - script to help generate DHCP stanzas to append to dhcpd.conf. Uses host and MAC addresses.
- opafastfabric.conf-sample - configuration file for opafastfabric. Used in /etc/sysconfig/opa.

4.2 opagentopology

Generates sample topology verification XML. Provides an example of using opaxmlgenerate and is a prototype for customization.

A multi-step process generates the required topology.xml file, as shown in the following figure. You must edit the sample topology.xlsx file, save the edited information in CSV format, and run the opaxlattopology script, which produces the topology.xml file.

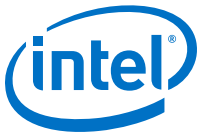
Figure 1. Topology Workflow



Uses CSV input files opatopology_links.txt, opatopology_FIs.txt, and opatopology_SWs.txt to generate LinkSummary, Node FIs, 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 opagentopology (not opaxmlgenerate) generates the XML version string as well as the <Topology> and <LinkSummary> lines. Also note that the indent level is at the default value of zero (0). The portions of the script that call opaxmlgenerate follow:

```
opaxmlgenerate -X /usr/lib/opa/samples/opatopology_1.txt -d \; -h Link
-g Rate -g Rate_Int -g MTU -g LinkDetails -h Cable -g CableLength -g CableLabel
-g CableDetails -e Cable -h Port -g NodeGUID -g PortNum -g NodeDesc -g PortGUID
```



```
-g NodeType -g NodeType_Int -g PortDetails -e Port -h Port -g NodeGUID -g PortNum
-g NodeDesc -g PortGUID -g NodeType -g NodeType_Int -g PortDetails -e Port -e Link

opaxmlgenerate -X /usr/lib/opa/samples/opatopology_2.txt -d \;
-h Node -g NodeGUID -g NodeDesc -g NodeDetails -g HostName -g NodeType
-g NodeType_Int -g NumPorts -e Node
```

opatopology_links.txt

This file can be found in `/usr/lib/opa/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;bender
HFI-1;0x0002c9020020e004;FI;Some info about port;0x0011750007000df6;7;Switch 1234
Leaf 4;;SW;

25g;;0;;;;;0x0002c9020025a678;1;mindy2 HFI-1;;FI;;0x0011750007000e6d;4;Switch
2345 Leaf 5;;SW;
```

opatopology_FIs.txt

This file can be found in `/usr/lib/opa/samples/`. For brevity, this sample shows only two nodes.

```
0x0002c9020020e004;bender HFI-1;More details about node
0x0002c9020025a678;mindy2 HFI-1;Node details
```

opatopology_SWs.txt

This file can be found in `/usr/lib/opa/samples/`. For brevity, this sample shows only two nodes.

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

opatopology_SMs.txt

This file can be found in `/usr/lib/opa/samples/`. For brevity, this sample shows only one node.

```
0x0002c9020025a678;1;mindy2 HFI-1;0x0011750007000e6d;FI;details about SM
```

Example

When run against the supplied topology input files, `opagentopology` produces:

```
<?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>
<NodeGUID>0x0002c9020020e004</NodeGUID>
<PortNum>1</PortNum>
<NodeDesc>bender HFI-1</NodeDesc>
<PortGUID>0x0002c9020020e004</PortGUID>
<NodeType>FI</NodeType>
<PortDetails>Some info about port</PortDetails>
</Port>
<Port>
<NodeGUID>0x0011750007000df6</NodeGUID>
<PortNum>7</PortNum>
<NodeDesc>Switch 1234 Leaf 4</NodeDesc>
<NodeType>SW</NodeType>
</Port>
</Link>
<Link>
<Rate>25g</Rate>
<Internal>0</Internal>
<Cable>
</Cable>
<Port>
<NodeGUID>0x0002c9020025a678</NodeGUID>
<PortNum>1</PortNum>
<NodeDesc>mindy2 HFI-1</NodeDesc>
<NodeType>FI</NodeType>
</Port>
<Port>
<NodeGUID>0x0011750007000e6d</NodeGUID>
<PortNum>4</PortNum>
<NodeDesc>Switch 2345 Leaf 5</NodeDesc>
<NodeType>SW</NodeType>
</Port>
</Link>
</LinkSummary>
<Nodes>
<FIs>
<Node>
<NodeGUID>0x0002c9020020e004</NodeGUID>
<NodeDesc>bender HFI-1</NodeDesc>
<NodeDetails>More details about node</NodeDetails>
</Node>
<Node>
<NodeGUID>0x0002c9020025a678</NodeGUID>
<NodeDesc>mindy2 HFI-1</NodeDesc>
<NodeDetails>Node details</NodeDetails>
</Node>
</FIs>
<Switches>
<Node>
<NodeGUID>0x0011750007000df6</NodeGUID>
<NodeDesc>Switch 1234 Leaf 4</NodeDesc>
</Node>
<Node>
<NodeGUID>0x0011750007000e6d</NodeGUID>
<NodeDesc>Switch 2345 Leaf 5</NodeDesc>
</Node>
</Switches>
<SMs>
<SM>
<NodeGUID>0x0002c9020025a678</NodeGUID>
<PortNum>1</PortNum>
<NodeDesc>mindy2 HFI-1</NodeDesc>
```

```
<PortGUID>0x0011750007000e6d</PortGUID>
<NodeType>FI</NodeType>
<SMDetails>details about SM</SMDetails>
</SM>
</SMs>
</Nodes>
</Topology>
```

4.3 topology.xlsx Overview

This section describes the `topology.xlsx` file that is installed in the `/usr/lib/opa/samples` directory.

`topology.xlsx` provides a standard format for representing each external link in a cluster. Each link contains **Source**, **Destination**, and **Cable** fields with one link per row of the spreadsheet. The cells cannot contain commas.

Figure 2. 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	host101	gw		FI	row1	rack1	opasw11		1	SW	host101 opasw11P1	1m	Cable CU
5			host102						opasw12		2		host102 opasw12P2	1m	Cable CU
6			host103						opasw13		3		host103 opasw13P3	1m	Cable CU
7			host104						opasw14		4		host104 opasw14P4	1m	Cable CU
8		rack2	host105			FI		rack3	opacore1	L101	1	CL	host105 opacore1L101P1	5m	Cable Fiber
9			host106						opacore1	L102	2		host106 opacore1L102P2	5m	Cable Fiber
10			host107						opacore1	L103	3		host107 opacore1L103P3	5m	Cable Fiber
11			host108						opacore1	L104	4		host108 opacore1L104P4	5m	Cable Fiber
12		rack1	opasw11		19	SW		rack3	opacore1	L108	9	CL	opasw11P19 opacore1L108	1m	Cable CU
13			opasw12		20				opacore1	L108	10		opasw12P20 opacore1L108	1m	Cable CU
14			opasw13		21				opacore1	L108	11		opasw13P21 opacore1L108	1m	Cable CU
15			opasw14		22				opacore1	L108	12		opasw14P22 opacore1L108	1m	Cable CU
16	row2	rack4	host201	lsw		FI	row2	rack4	opasw21		1	SW	host201 opasw21P1	1m	Cable CU
17			host202						opasw22		2		host202 opasw22P2	1m	Cable CU
18			host203						opasw23		3		host203 opasw23P3	1m	Cable CU
19			host204						opasw24		4		host204 opasw24P4	1m	Cable CU
20		rack5	host205			FI		rack6	opacor2	L101	1	CL	host205 opacor2L101P1	5m	Cable Fiber
21			host206						opacor2	L102	2		host206 opacor2L102P2	5m	Cable Fiber
22			host207						opacor2	L103	3		host207 opacor2L103P3	5m	Cable Fiber
23			host208						opacor2	L104	4		host208 opacor2L104P4	5m	Cable Fiber
24		rack4	opasw21		19	SW		rack6	opacor2	L108	9	CL	opasw21P19 opacor2L108P	1m	Cable CU
25			opasw22		20				opacor2	L108	10		opasw22P20 opacor2L108P	1m	Cable CU
26			opasw23		21				opacor2	L108	11		opasw23P21 opacor2L108P	1m	Cable CU
27			opasw24		22				opacor2	L108	12		opasw24P22 opacor2L108P	1m	Cable CU
28	Xrow	Xrack	Xhost			FI	Xrow	Xrack	Xswitch		1	SW			
29															
30	Core Name:opacore1	Core Group:row1	Core Rack:rack3	Core Size:288	Core Full:1										
31	Core Name:opacor2	Core Group:row2	Core Rack:rack6	Core Size:1152	Core Full:0										
32															

The previous figure shows examples of links between HFI and Intel® OP Edge Switch 100 Series (rows 4-7), HFI 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)

  User-defined primary name of host or switch. Intel recommends that host names match the host names configured in `/etc/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)

  Port contains the port number of the HFI or Switch port. If the Port field is empty, the script defaults to 1.

- Type

  Type contains the device type. When creating the spreadsheet to verify external links, use the following values for type: `FI`, `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**: FI for HFI adapter
  - **Edge Switch**: SW
  - **Core Leaf**: CL for Director switch core leaf module

- Cable (optional)

  Cable values are optional and have no special syntax.

  The **Cable** fields have the following columns:

  - Label - Max characters = 57. (In release 10.2 and earlier, this field is limited to 20 characters.)
  - Length
  - Details

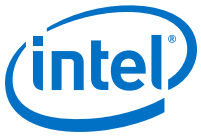
Core Full Statement

At the bottom of the `/usr/lib/opa/sample/topology.xlsx` file, there is a core full statement to indicate if the Intel® OP Director Class Switch 100 Series is fully populated with all spine and leaf modules installed. If there are multiple 6-slot or 24-slot Director switches in the fabric, each Director switch should have an entry in the `topology.xlsx` file as shown in the following table.

Table 3. Core Full Statement Definitions

`Core Name:Core01`	`Core Group:row1`	`Core Rack:rack01`	`Core Size:1152`	`Core Full:0`
`Core Name:Core02`	`Core Group:row1`	`Core Rack:rack02`	`Core Size:1152`	`Core Full:0`
`Core Name: Specified in "Name" Column of topology.xlsx`	`Core Group: Specified in "Rack Group" Column of topology.xlsx`	`Core Rack: Specified in "Rack" Column of topology.xlsx`	`Core Size: Set to 1152 for 24 slot Director switch, 288 for 6 slot Director switch. Represents all internal (spine) links`	`0: Use for partially populated director.` `1: Use for fully populated director.` **`Known Issue:`** `Currently when Core Full is 0, none of the`

continued...



			for fully populated Director.	internal links are put in the output xml file. This means that "partially populated" is not working correctly. This will be added in a future release.

5.0 MPI Sample Applications

5.1 Overview

As part of a Intel® Omni-Path Fabric Suite FastFabric Toolset installation, sample MPI applications and benchmarks are installed in `/usr/lib/opa/src/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:

- Latency/bandwidth deviation test
- OSU latency (3 versions)
- OSU bandwidth (3 versions)
- OSU bidirectional bandwidth
- HPL2
- Intel® MPI Benchmarks (IMB)
- Pallas MPI Benchmarks (PMB)

5.1.1 Building MPI Sample Applications

Perform the following procedure to build the applications:

1. Type `export MPICH_PREFIX=/usr/mpi/X/Y`

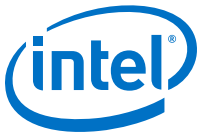
   where:

   - `X` is a compiler such as `gcc`
   - `Y` is an MPI variation such as `openmpi-1.2.5`

   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/lib/opa/src/mpi_apps/get_selected_mpi.sh`

   This will show you the currently selected MPI and set the `MPICH_PREFIX` variable to match.
2. Type `cd /usr/lib/opa/src/mpi_apps`
3. Type `make clean`
4. Type `make full` which builds all of the sample applications.

Note: The MPI used does not have to be in the `/usr/mpi` directory. The default MPIs installed with the Intel® OP Software are located here, however, you can also export `MPICH_PREFIX` to point to any location where you have another third party MPI installed.



The Intel® Omni-Path Fabric Suite FastFabric TUI can assist with building the MPI sample applications by providing a simple way to select the MPI to use for the build.

Alternatives include:

- `opa-base` - Builds applications in core RPM: Deviation, group stress, and `mpi_check`.
- `quick` – Builds everything in `opa-base`, plus OSU1 Latency, OSU1 Bandwidth, OSU2, OSU3.8, Intel® MPI Benchmarks (IMB), Deviation, HPL2, Group Stress
- `full` - Builds everything from `quick`.
- `all` – Builds everything from `full`.

5.1.2 Running MPI Sample Applications

To run the applications, an `mpi_hosts` file must be created in `/usr/lib/opa/src/mpi_apps` that provides the names of the hosts on which processes should be run. Either IPoIB or Ethernet names can be specified. Typically, use of IPoIB names provides faster job startup, especially on larger clusters. 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 `mpi_hosts` is used.

If a host has more than one real CPU, its name may appear in the MPI hosts file once per CPU.

Note: Intel® Xeon® Processors support hyper-threading; however, it significantly impacts performance for floating point intensive MPI applications, such as HPL2. For this reason, Intel recommends that you disable hyper-threading.

Note: When running the applications, all hosts listed in `MPI_HOSTS` must have a copy of the applications compiled for the same value of `MPICH_PREFIX`, for example, the same variation and version of MPI.

When the `run_*` scripts are used to execute the applications, the variation of MPI used to build the applications is detected and the proper mpirun is used to start the application.

To determine which variation of MPI the applications have been built, use the command:

```
cat /usr/lib/opa/src/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 a file in `/usr/lib/opa/src/mpi_apps/logs`. The file name includes the date and time of the run for uniqueness.

The run_* scripts automatically use the ofed.openmpi.params or ofed.mvapich2.params files to set up parameters for mpirun. These files have various samples of setting parameters such as vFabric selection, dispersive routing, etc. These parameter files can also set the MPI_CMD_ARGS variable to provide additional arguments to mpirun.

The current run_* scripts include:

- run_allhfilatency Checks the latencies of every pair of HFIs in the fabric.
- run_alltoall3 Runs the OSU3 all-to-all benchmark.
- run_batch_cabletest Runs the run_cabletest script on every node in the fabric, but runs them in batches to reduce the load on the fabric.
- run_bcast2 Runs OSU2 broadcast test.
- run_bcast3 Runs OSU3 broadcast test.
- run_bibw3 Runs OSU3 bidirectional bandwidth test.
- run_bw Runs OSU1 bandwidth test.
- run_bw2 Runs OSU2 bandwidth test.
- run_bw3 Runs OSU3 bandwidth test.
- run_cabletest Stresses groups of nodes in the fabric to discover possible bad cables.
- run_deviation Runs the deviation test.
- run_hpl2 Runs HPL V2.
- run_imb Runs Intel® MPI Benchmarks (IMB).
- run_lat Runs OSU1 latency test.
- run_lat2 Runs OSU2 latency test.
- run_lat3 Runs OSU3 latency test.
- run_mbw_mr3 Runs OSU3 mbw_mr test (multibandwidth message rate test).
- run_mpicheck Simple test to validate that MPI is passing data correctly.
- run_multi_lat3 Runs OSU3 multi-latency test.
- run_multibw Runs OSU1 multi-bandwidth test.

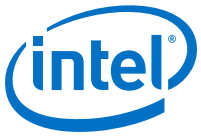
5.2 Latency/Bandwidth Deviation Test

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® Omni-Path Fabric Suite FastFabric Toolset Check MPI performance TUI menu item. It can also be invoked using opahost mpiperfdeviation.

Perform the following procedure to use the script provided to run this application:

1. Type cd /usr/lib/opa/src/mpi_apps
2. Type ./run_deviation *NP*



where:

NP is the number of processes to run or `all`, such as:

```
./run_deviation 4
```

This runs a quick latency and bandwidth test against pairs of the hosts specified in `mpi_hosts`. 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, regardless of how many CPUs the host has.

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

  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
```

5.3 OSU Tests

5.3.1 OSU Latency

This 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 /usr/lib/opa/src/mpi_apps`
2. Type `./run_lat`

This 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.

This benchmark only uses the first two nodes listed in `MPI_HOSTS`.

5.3.2 OSU Latency2

This is a simple performance test of end-to-end latency for various MPI message sizes. The values reported are one-direction latency.

Perform the following steps:

1. Type `cd /usr/lib/opa/src/mpi_apps`
2. Type `./run_lat2`

This runs assorted latencies from 0 to 4 Megabytes.

This benchmark only uses the first two nodes listed in `MPI_HOSTS`.

5.3.3 OSU Latency 3

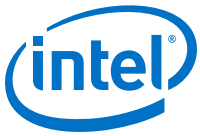
This is a simple performance test of end-to-end latency for various MPI message sizes. The values reported are one-direction latency.

Perform the following steps:

1. Type `cd /usr/lib/opa/src/mpi_apps`
2. Type `./run_lat3`

This runs assorted latencies from 0 to 4 Megabytes.

This benchmark only uses the first two nodes listed in `MPI_HOSTS`.



5.3.4 OSU Multi Latency3

This is a simple performance test of end-to-end latency for multiple concurrent pairs of hosts for various MPI message sizes. The values reported are average one-direction latency.

Perform the following steps:

1. Type `cd /usr/lib/opa/src/mpi_apps`
2. Type `./run_multi_lat3 NP`

   where:

   NP is the number of processes to run or `all`, such as:

```
./run_multi_lat3 4
```

This runs assorted latencies from 0 to 4 Megabytes.

This benchmark only uses the first *NP* nodes listed in `MPI_HOSTS`.

5.3.5 OSU Bandwidth

This is a simple benchmark of maximum unidirectional bandwidth.

Perform the following steps:

1. Type `cd /usr/lib/opa/src/mpi_apps`
2. Type `./run_bw`

This runs assorted bandwidths from 4K to 4Mbytes. To run a different set of message sizes, an optional argument specifying the maximum message size can be provided.

This benchmark only uses the first two nodes listed in `MPI_HOSTS`.

5.3.6 OSU Bandwidth2

This is a simple benchmark of maximum unidirectional bandwidth.

Perform the following steps:

1. Type `cd /usr/lib/opa/src/mpi_apps`
2. Type `./run_bw2`

This runs assorted bandwidths from 1 byte to 4Mbytes.

This benchmark only uses the first two nodes listed in `MPI_HOSTS`.

5.3.7 OSU Bandwidth3

This is a simple benchmark of maximum unidirectional bandwidth.

Perform the following steps:

1. Type `cd /usr/lib/opa/src/mpi_apps`
2. Type `./run_bw3`

This runs assorted bandwidths from 1 byte to 4Mbytes.

This benchmark only uses the first two nodes listed in `MPI_HOSTS`.

5.3.8 OSU Multi Bandwidth3

This is a simple benchmark of aggregate unidirectional bandwidth and messaging rate for multiple concurrent pairs of nodes.

Perform the following steps:

1. Type `cd /usr/lib/opa/src/mpi_apps`
2. Type `./run_mbw_mr3` *NP*

   where:

   NP is the number of processes to run or `all`, such as:

```
./run_mbw_mr3 4
```

This runs assorted messaging rates from 1 byte to 4Mbytes.

5.3.9 OSU Bidirectional Bandwidth

This is a simple benchmark of maximum bidirectional bandwidth.

Perform the following steps:

1. Type `cd /usr/lib/opa/src/mpi_apps`
2. Type `./run_bibw2`

This runs assorted bandwidths from 1 byte to 4Mbytes.

This benchmark only uses the first two nodes listed in `MPI_HOSTS`.

5.3.10 OSU Bidirectional Bandwidth3

This is a simple benchmark of maximum bidirectional bandwidth.

Perform the following steps:

1. Type `cd /usr/lib/opa/src/mpi_apps`
2. Type `./run_bibw3`

This runs assorted bandwidths from 1 byte to 4Mbytes.

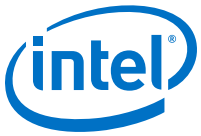
This benchmark only uses the first two nodes listed in `MPI_HOSTS`.

5.3.11 OSU All to All 3

This is a simple benchmark of AllToAll latency.

Perform the following steps:

1. Type `cd /usr/lib/opa/src/mpi_apps`
2. Type `./run_alltoall3` *NP*



where:

NP is the number of processes to run or `all`, such as:

```
./run_alltoall3 4
```

This runs assorted latencies from 1 byte to 1Mbytes.

5.3.12 OSU Broadcast 3

This is a simple benchmark of Broadcast latency.

Perform the following steps:

1. Type `cd /usr/lib/opa/src/mpi_apps`
2. Type `./run_bcast3 NP`

   where:

   NP is the number of processes to run or `all`, such as:

```
./run_bcast3 4
```

This runs assorted latencies from 1 byte to 16K bytes.

5.3.13 OSU Multiple Bandwidth/Message Rate

The Multiple Bandwidth / Message Rate Test (`osu_mbw_mr`) is intended to be used with block assigned ranks. This means that all processes on the same machine are assigned ranks sequentially.

Note: All benchmarks are run using two processes, except for `osu_bcast` and `osu_mbw_mr` which can use more than two processes.

If you're using `mpd` with MVAPICH2, you must specify the number of processes on each host in the host file, otherwise `mpd` assigns ranks in a cyclic fashion. Refer to the following table for rank assignments.

Table 4. Rank Assignment

Rank	Block	Cyclic
0	host1	host1
1	host1	host2
2	host1	host1
3	host1	host2
4	host2	host1
5	host2	host2
6	host2	host1
7	host2	host2

Here is an example of MPD HOSTFILE:

```
host1:4
host2:4

MPI-1
-----
osu_bcast       - Broadcast Latency Test
osu_bibw        - Bidirectional Bandwidth Test
osu_bw          - Bandwidth Test
osu_latency     - Latency Test
osu_mbw_mr      - Multiple Bandwidth / Message Rate Test
osu_multi_lat   - Multi-pair Latency Test

MPI-2
-----
osu_acc_latency - Accumulate Latency Test
osu_get_bw      - One-Sided Get Bandwidth Test
osu_get_latency - One-Sided Get Latency Test
osu_latency_mt  - Multi-threaded Latency Test
osu_put_bibw    - One-Sided Put Bidirectional Test
osu_put_bw      - One-Sided Put Bandwidth Test
osu_put_latency - One-Sided Put Latency Test
```

5.4 Latency Tests

The latency tests are carried out in a ping-pong fashion. The sender sends a message with a certain data size to the receiver and waits for a reply from the receiver. The receiver receives the message from the sender and sends back a reply with the same data size. Many iterations of this ping-pong test are carried out and average one-way latency numbers are obtained. Blocking version of MPI functions (MPI_Send and MPI_Recv) are used in the tests.

5.4.1 Multi-Threaded Latency Test

The multi-threaded latency test performs a ping-pong test with a single sender process and multiple threads on the receiving process. In this test, the sending process sends a message of a given data size to the receiver and waits for a reply from the receiver process. The receiving process has a variable number of receiving threads (set by default to 2), where each thread calls MPI_Recv and upon receiving a message sends back a response of equal size. Many iterations are performed and the average one-way latency numbers are reported.

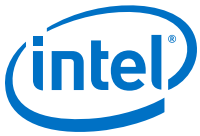
Note: This test is only applicable for MVAPICH2 with threading support enabled.

5.4.2 Multi-Pair Latency Test

This test is very similar to the latency test, except that multiple pairs are performing the same test simultaneously. In order to perform the test across just two nodes, the hostnames must be specified in block fashion.

5.4.3 Broadcast Latency Test

This test is carried out in the following manner.



After doing an MPI_Bcast, the root node waits for an acknowledgment from the last receiver. This acknowledgment is in the form of a zero byte message from the receiver to the root. This test is carried out for a large number (1000) of iterations. The Broadcast latency is obtained by subtracting the time taken for the acknowledgment from the total time. The acknowledgment time is computed by doing a ping-pong test.

5.4.4 One-Sided Put Latency Test

The sender (origin process) calls MPI_Put (ping) to directly place a message of certain data size in the receiver window. The receiver (target process) calls MPI_Win_wait to make sure the message has been received. Then the receiver initiates a MPI_Put (pong) of the same data size to the sender, which is now waiting on a synchronization call. Several iterations of this test are carried out, and the average put latency is obtained.

Note: This test is only applicable for MVAPICH2.

5.4.5 One-Sided Get Latency Test

The origin process calls MPI_Get (ping) to directly fetch a message of certain data size from the target process window to its local window. It then waits on a synchronization call (MPI_Win_complete) for local completion. After the synchronization call, the target and origin processes are switched for the pong message. Several iterations of this test are carried out and the average get latency is obtained.

Note: This test is only applicable for MVAPICH2.

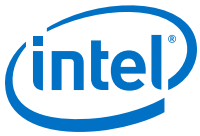
5.4.6 One-Sided Accumulate Latency Test

The origin process calls MPI_Accumulate to combine the data moved to the target process window with the data that resides at the remote window. The combining operation used in the test is MPI_SUM. The origin process then waits on a synchronization call (MPI_Win_complete) for local completion. After the synchronization call, the target and origin process are switched for the pong message. Several iterations of this test are carried out, and the average accumulate latency number is obtained.

Note: This test is only applicable for MVAPICH2.

5.5 Bandwidth Tests

The bandwidth tests are carried out by having the sender sending out a fixed number (equal to the window size) of back-to-back messages to the receiver and then waiting for a reply from the receiver. The receiver sends the reply only after receiving all these messages. This process is repeated for several iterations and the bandwidth is calculated based on the elapsed time (from the time sender sends the first message until the time it receives the reply back from the receiver) and the number of bytes sent by the sender. The objective of these bandwidth tests is to determine the maximum sustained date rate that can be achieved at the network level. Non-blocking versions of MPI functions (MPI_Isend and MPI_Irecv) are used in the test.



5.5.1 Bidirectional Bandwidth Test

The bidirectional bandwidth test is similar to the bandwidth test, except that both nodes send out a fixed number of back-to-back messages and wait for the reply. This test measures the maximum sustainable aggregate bandwidth by two nodes.

5.5.2 Multiple Bandwidth / Message Rate Test

The multi-pair bandwidth and message rate test evaluates the aggregate uni-directional bandwidth and message rate between multiple pairs of processes. Each of the sending processes sends a fixed number of messages (the window size) back-to-back to the paired receiving process before waiting for a reply from the receiver. This process is repeated for several iterations. The objective of this benchmark is to determine the achieved bandwidth and message rate from one node to another node with a configurable number of processes running on each node.

5.5.3 One-Sided Put Bandwidth Test

The bandwidth tests are carried out by the origin process calling a fixed number of back-to-back Puts, and then waiting on a synchronization call (MPI_Win_complete) for completion. This process is repeated for several iterations, then the bandwidth is calculated, based on the elapsed time and the number of bytes sent by the origin process.

Note: This test is only applicable for MVAPICH2.

5.5.4 One-Sided Get Bandwidth Test

The bandwidth tests are carried out by an origin process calling a fixed number of back-to-back Gets, and then waiting on a synchronization call (MPI_Win_complete) for completion. This process is repeated for several iterations, then the bandwidth is calculated based on the elapsed time and the number of bytes sent by the origin process.

Note: This test is only applicable for MVAPICH2.

5.5.5 One-Sided Put Bidirectional Bandwidth Test

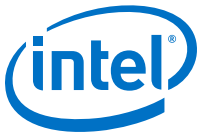
The bidirectional bandwidth test is similar to the bandwidth test, except that both nodes send out a fixed number of back-to-back Put messages and wait for their completion. This test measures the maximum sustainable aggregate bandwidth by two nodes.

Note: This test is only applicable for MVAPICH2.

5.6 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 you and your local peers. 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_processes] [mpi_stress arguments]
```

Options

mpi_stress arguments

- `-a` *INT* – desired alignment for buffers (must be power of 2)
- `-b` *BYTE* – byte value to initialize non-random send buffers (otherwise 0)
- `-c` – enable CRC checksums
- `-D` *INT* – set max data amount per msg size (default 1073741824)
- `-d` – enable data checksums (otherwise headers only)
- `-e` – exercise the interconnect with random length messages
- `-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` – display this help page
- `-i` – include local ranks as destinations (only for all-to-all)
- `-I` *INT* – set msg size increment (default power of 2)
- `-l` *INT* – set min msg size (default 0)
- `-L` *INT* – set min msg count (default 100)
- `-m` *INT* – set max msg size (default 4194304)
- `-M` *INT* – set max msg count (default 10000)
- `-n` *INT* – number of times to repeat (default 1)
- `-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 – uni-directional traffic (only for grid)
- -v – enable verbose mode (more -v for more verbose)
- -w *INT* – number of send/recv in window (default 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

5.7 High Performance Linpack (HPL2)

This test is a standard benchmark for Floating Point Linear Algebra performance. Version 2.0 is provided, which includes the Dr K. Goto Linear Algebra library. If desired, you can modify the HPL2 `makefiles` to use alternate libraries. Atlas source code and the open source math library are also provided in `/usr/lib/opa/src/mpi_apps/ATLAS`. On RHEL systems, HPL2 attempts to use the Atlas package provided in the distribution.

Note: The Linear Algebra Library is highly optimized for a given CPU model. When running in a fabric with mixed CPU models, the HPL2 application must be rebuilt for each CPU model and that version must be used on all CPUs of the given type. Attempting to run a CPU with a library that is not optimized for the given CPU results in less than optimal performance. In some cases (such as trying to run an AMD CPU-optimized library on an Intel CPU), HPL2 may fail or produce incorrect results.

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).

Prior to running this application, an `HPL.dat` file must be installed in `/usr/lib/opa/src/mpi_apps/hpl2/bin/ICS.${ARCH}.${CC}` on all nodes. The `config_hpl2` script and some sample configurations are included.

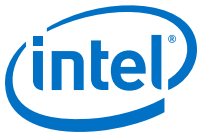
The `config_hpl2` script can select from one of the assorted `HPL.dat` files in `opt/opa/src/mpi_apps/hpl-config`. These files are a good starting point for most clusters, and should get within 10-20% of the optimal performance for the cluster. The problem sizes used assume a cluster with 1GB of physical memory per processor. For each cluster size, 4 files are provided:

- `t` – A very small test run (5000 problem size)
- `s` – A small problem size on the low end of optimal problem sizes
- `m` – A medium problem size
- `l` – A large problem size

These can be selected using `config_hpl2`. The following command displays the pre-configured problem sizes available:

```
./config_hpl2
```

For example, to quickly confirm that HPL2 runs on the 16 nodes in the `/usr/lib/opa/src/mpi_apps/mpi_hosts` file:



1. Type `./config_hpl2 16t`.

   This command edits the `HPL.dat` file on the local host for a 16 host "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 /usr/lib/opa/src/mpi_apps`

3. Type `./run_hpl2 NP`

   where:

   NP is the number of processors for the run, or `all`. For example:

```
./run_hpl2 16
```

For more information about HPL2, refer to the `README`, `TUNING`, and assorted HTML files in the `/usr/lib/opa/src/mpi_apps/hpl2` directory.

5.8 Intel® MPI Benchmarks (IMB)

Use the `run_imb` sample script in `/usr/lib/opa/src/mpi_apps` to run the Intel® MPI Benchmarks (IMB).

1. Type `cd /usr/lib/opa/src/mpi_apps`
2. Type `./run_imb NP`

   where:

   NP is the number of processes to run, or `all`. A minimum of two processes is required. For example:

```
./run_imb 4
```

5.9 Pallas MPI Benchmark (PMB)

The Pallas MPI benchmark performs exhaustive benchmarking of latency and bandwidth for assorted message sizes for many MPI primitives. This benchmark is a good tool for evaluating and tuning small clusters, or a subset of a large cluster.

PMB has known scalability limitations, particularly in its **AllToAll** phase. This phase can simultaneously perform up to 4 MB transfers to and from all nodes at once. However, there is a downside in that a system must have approximately 10*NP MB of memory available per process for Pallas data to run this benchmark. Therefore, for a small cluster (approximately 16 processors or less), the memory requirement is modest at 160 MB. However, for a larger cluster (approximately 256 processors or greater), the memory requirement is rather large at 2.5 GB.

Intel recommends that you use PMB for smaller runs (2-32 processes), since the benchmark is likely to fail at larger process counts. Depending upon the amount of memory in the system and the numbers of processes to run, the `VIADEV_MEM_REG_MAX` parameter in `/usr/lib/opa/src/mpi_apps/mpi.param.pallas` may need to be edited.

To run the benchmark:

1. Type `cd /usr/lib/opa/src/mpi_apps`
2. Type `./run_pmb` *NP*

   where:

   NP is the number of processes to run, or `all`. For example:

```
./run_pmb 4
```

5.10 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 performance issues in the fabric, such as bad cables.

5.10.1 All HFI Latency

The All HFI Latency test is a specialized stress test for large fabrics. It iterates through every possible pairing of the HFIs 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 `/usr/lib/opa/src/mpi_apps`.

   `cd /usr/lib/opa/src/mpi_apps`
2. Run the All HFI Latency test

   `./run_allhfilatency`

This test runs a 60 second test on the first two nodes listed in the `mpi_hosts` file.

To change the default behavior, specify up to three optional arguments, for example:

`./run_allhfilatency` *NP MN SS*

where:

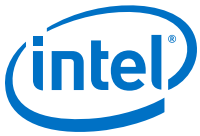
NP is the number of processes to run, or `all`.

MN is the number of minutes the test should run.

SS is the size of the messages to use when testing (between 1 byte and 4 megabytes).

For example, to run a 30 minute test on 64 nodes with 4 kilobyte messages, the following command would be used from the `/usr/lib/opa/src/mpi_apps` directory:

```
./run_allhfilatency 64 30 4096
```



Once 30 minutes has elapsed, the test completes as soon as the current round of testing has completed.

If you want the tests to repeat indefinitely, use the duration `infinite` as shown in the following CLI command:

```
./run_allhfilatency 64 infinite 4096
```

There are three options, `-c`, `-h`, and `-v` available:

- `-h` / `--help` Provides some help text, then terminates.
- `-c` / `--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` / `--verbose` Runs the test in a verbose mode that shows more information.

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 interswitch link causing those nodes to experience degraded performance.

5.10.2 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 on `gen_group_hosts`, refer to gen_group_hosts on page 240.

By default, `run_cabletest` runs for 60 minutes and uses 4-megabyte 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 `/usr/lib/opa/src/mpi_apps`.

   ```
   cd /usr/lib/opa/src/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-megabyte messages, enter the following CLI command:

```
./run_cabletest 1
```

Once one minute has elapsed, the test completes when the current round of testing completes.

If you want the tests to repeat indefinitely, use `infinite` as the duration, 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 megabytes). The following example tests message sizes between 1 and 4 megabytes, and runs for 24 hours:

```
./run_cabletest 1440 1048576 4194304
```

There are two options available, `-h` and `-v`:

- `-h` / `--help` – provides this help text.
- `-v` / `--verbose` – runs the test in a verbose mode that shows you how the nodes were grouped.

5.10.3 run_batch_cabletest

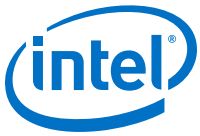
The run_batch_cabletest in `/usr/lib/opa/src/mpi_apps` makes it easier to run the `run_cabletest` stress test (see run_cabletest on page 237). The `run_batch_cabletest` script runs separate jobs for each `BATCH_SIZE` hosts, and can generate the `mpi_group_hosts` files needed using a single `mpi_hosts` file, which lists each host to be tested once, in topology order. For many clusters, `opasorthosts` may help put a list of hosts in topology order, or `opafindgood` may be used to identify candidate hosts. 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_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–megabyte 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 `/usr/lib/opa/src/mpi_apps`.

   ```
   cd /usr/lib/opa/src/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, use `infinite` as the duration, 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 megabytes, 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` – how many minutes to run. Default is 60.
- *`minmsg`* – smallest message to use. Must be between 16384 and 4194304.
- *`maxmsg`* – largest message to use. Must be between 16384 and 4194304.

Default *`minmsg`* and *`maxmsg`* is 4 Megabytes.

Each run_cabletest MPI job has its output saved to a corresponding `/tmp/nohup.#.out` file.

Environment Variables

- MPI_HOSTS - mpi_hosts file to use. The default is 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
```

5.10.4 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.

The first question asks for 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 /usr/lib/opa/src/mpi_apps/mpi_hosts.

The second question asks how big your groups are. 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.

The third question asks how many processes you want to run per node. 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 /usr/lib/opa/src/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.

5.10.5 run_multibw

run_multibw runs mpi_multibw, which performs a multi-core pairwise bandwidth test. mpi_multibw is based on OSU bw and multi-lat.

1. Change directory to /usr/lib/opa/src/mpi_apps.

   ```
   cd /usr/lib/opa/src/mpi_apps
   ```

2. Run the run_multibw test including the number of processes on which to run the test.

   ```
   ./run_multibw processes
   ```

   where: *processes* is the number of processes on which to run the test. All indicates the test should be run for every process in the mpi_hosts file.

5.10.6 run_nxnlatbw

run_nxnlatbw runs mpi_nxnlatbw, which is an NxN latency bandwidth test.

1. Change directory to /usr/lib/opa/src/mpi_apps.

   ```
   cd /usr/lib/opa/src/mpi_apps
   ```

2. Run the run_nxnlatbw test, including the number of processes on which to run the test.

   ```
   ./run_nxnlatbw processes
   ```

   where: *processes* is the number of processes on which to run the test. All indicates the test should be run for every process in the mpi_hosts file.

5.11 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 /usr/lib/opa/src/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 – Force an even number of hosts in the final batch by skipping the last one.
- run_script – A run_* script from this directory
- *args* – 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 is only used for scripts that use MPI_HOSTS.

To run run_cabletest, use run_batch_cabletest.

Environment Variables

- MPI_HOSTS – 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

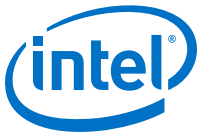
```
./run_batch_script run_deviation NP ff
BATCH_SIZE=2 MPI_HOSTS=good ./run_batch_script run_lat2
BATCH_SIZE=16 MPI_HOSTS=good ./run_batch_script run_deviation ff
MIN_BATCH_SIZE=16 BATCH_SIZE=16 ./run_batch_script run_hpl2 16
```

5.11.1 SHMEM Batch run_* scripts

Scripts for various SHMEM benchmarks included with SHMEM are contained in /usr/lib/opa/src/shmem_apps. The behavior of these scripts is very similar to those in mpi_apps.

Each SHMEM application/benchmark has an accompanying run_* script, which assumes the existence of a local mpi_hosts file. The provided run_* scripts include the following:

- run_alltoall
- run_barrier
- run_reduce
- run_get[put]_bw
- run_get[put]_bibw



6.0 Port Counters Overview

Each port in an Intel® Omni-Path Fabric maintains a set of port counters to indicate both traffic and error counts. These counters can be grouped into the categories described in this section. Each port stops incrementing when the max value is reached, irrespective of counter size. Most of the counters are 64-bits in size. Exceptions are noted.

6.1 Utilization

These counters reflect the normal utilization of the port and Virtual Lane when present.

Several of these counters are used during the calculation of Congestion, SMA Congestion, and the Bubble Categories. The Utilization metrics provide a way of giving some of the other counters context by comparing them to the amount of data or packets that were transmitted or received.

6.1.1 PortXmitData & PortVLXmitData[n]

These counters indicate the total number of fabric packet flits transmitted. This does not include idle nor other LF command flits.

6.1.2 PortRcvData & PortVLRcvData[n]

These counters indicate the total number of fabric packet flits received.

6.1.3 PortMulticastXmitPkts

This counter indicates the number of multicast and collective packets transmitted.

6.1.4 PortMulticastRcvPkts

This counter indicates the number of multicast and collective packets received.

6.2 Link Integrity

These counters reflect errors in the Physical (PHY) and Link Layers, as well as errors in firmware. In some cases, these errors are benign and can be ignored. However in other cases, excessive link integrity errors can indicate a hardware problem such as a poor connection, marginal cable, incorrect length/model cable for signal rate, or damaged/broken hardware, such as bad connectors.

When a bad packet is detected, one of these counters is incremented and the Link Layer may either discard or replay the packet.

During the link training sequence, assorted errors may be observed. This is a normal part of the link training and clock synchronization process. Hence, errors observed as part of rebooting nodes or moving cables should not be considered a problem.

The category is calculated as a weighted sum of the counters in the group. With the exception of ExcessiveBufferOverrunErrors, the counters in this group report on the receive side of the link. However, the counter can indicate a problem on either side of the link.

6.2.1 Link Quality Indicator (LQI)

This is a status indicator, similar to the signal strength bar display on a mobile phone, that enumerates link quality as a range of 0-5, with 5 being very good. Values in the lower part of the range may indicate hardware problems such as port, cable, and others that surface as signal integrity issues, leading to performance and other problems.

Table 5. Link Quality Values and Description

Link Quality Value	Description
5	Working at or above preferred link quality, no action needed.
3	Working on low end of acceptable link quality, recommended corrective action on next maintenance window.
2	Working below acceptable link quality, recommend timely corrective action.
1	Working far below acceptable link quality, recommend immediate corrective action.
0	Link down

6.2.2 LocalLinkIntegrityErrors Counter

This counter indicates the number of retries initiated by a link transfer layer receiver.

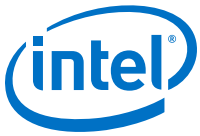
The retry rate is represented by the Link Quality Indicator. A link that is meeting electrical performance requirements has a Link Quality of 5, which corresponds to 1000 or fewer replays per second.

6.2.3 PortRcvErrors Counter

This counter indicates the total number of packets containing an error that were received by the port, including Link Layer protocol violations and malformed packets. It indicates possible misconfiguration of a port, either by the SM or, more likely, by user intervention. It can also indicate hardware issues or extremely poor signal integrity for a link.

6.2.4 ExcessiveBufferOverrunErrors Counter

This counter, associated with credit management, indicates an input buffer overrun. It indicates possible misconfiguration of a port, either by the SM or, more likely, by user intervention. It can also indicate hardware issues or extremely poor signal integrity for a link.



6.2.5 LinkErrorRecovery Counter

This counter indicates the number of times the link has successfully completed the link error recovery process.

Link Quality Indicator is the primary indicator for link quality to use. This counter is factored into the value reported for Link Quality Indicator. This counter may be non-zero for a properly functioning link.

6.2.6 LinkDowned Counter

This counter indicates the total number of times the port has failed the link error recovery process and downed the link. These events can cause disruptions to fabric traffic.

6.2.7 UncorrectableErrors Counter

This counter indicates the number of unrecoverable internal device errors. It indicates a severe hardware defect or data corruption inside the device.

6.2.8 FMConfigErrors Counter

This counter indicates inconsistencies of low level SMA configuration on both sides of the link. It indicates possible misconfiguration of a port, either by the SM, or, more likely, by user intervention.

6.3 Congestion

These counters reflect possible errors that indicate traffic congestion in the fabric.

When congestion or a packet that has seen congestion is detected, one of these counters is incremented and then depending on the issue reported, the packet must wait. In an extreme case, the packet may time out and be dropped.

The category is calculated as a weighted sum of the counters in the context of the utilization counters. With the exception of PortRcvFECN, the counters are all reported on the transmit side of the link. In addition, PortRcvBECN is only taken if the local node is an HFI. However, the counter could indicate a problem on either side of the link.

6.3.1 CongDiscards Counter

Note: Formerly known as "SwPortCongestion".

This switch-only counter indicates the number of packets that were discarded as unable to transmit due to timeouts.

6.3.2 PortRcvFECN Counter

When a device receives a packet with the FECN (Forward Explicit Congestion Notification) bit set to one, this counter is incremented.

6.3.3 PortRcvBECN Counter

When a device receives a packet with the BECN (Backward Explicit Congestion Notification) bit set to one, this counter is incremented.

6.3.4 PortMarkFECN Counter

This counter indicates the total number of packets that were marked FECN (Forward Explicit Congestion Notification) by the transmitter due to congestion.

6.3.5 PortXmitTimeCong Counter

This counter indicates the total number of *flit times* that the port was in a congested state for any data VL.

6.3.6 PortXmitWait Counter

This counter indicates the amount of time (in *flit times*) any virtual lane had data but was unable to transmit due to no credits available.

6.4 SMA Congestion

These counters reflect congestion in the fabric specific to communication between the Subnet Manager and Subnet Manager Agents using the management VL (VL 15).

The category is calculated exactly as the Congestion category using the same weights and the correct VL15 utilization counters.

6.4.1 PortVLXmitWait[15] Counter

This counter behaves the same as PortXmitWait, but it is restricted to VL 15, which carries only SM traffic.

6.4.2 SwPortVLCongestion[15] Counter

This counter behaves the same as CongDiscards, but it is restricted to VL 15, which carries only SM traffic.

6.4.3 PortVLRcvFECN[15] Counter

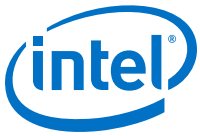
This counter behaves the same as PortRcvFECN, but it is restricted to VL 15, which carries only SM traffic.

6.4.4 PortVLRcvBECN[15] Counter

This counter behaves the same as PortRcvBECN, but it is restricted to VL 15, which carries only SM traffic.

6.4.5 PortVLXmitTimeCong[15] Counter

This counter behaves the same as PortXmitTimeCong, but it is restricted to VL 15, which carries only SM traffic.



6.4.6 PortVLMarkFECN[15] Counter

This counter behaves the same as PortMarkFECN, but it is restricted to VL 15, which carries only SM traffic.

6.5 Bubble

These counters occur when an unexpected idle flit is transmitted or received.

The transmit port sends idle flits until it can continue sending the rest of the packet. The category is calculated as follows:

1. The maximum value between the sum of the XmitWastedBW and XmitWaitData or the neighbor's PortRcvBubble.
2. Then divide the previous value by the port's utilization to provide context.

6.5.1 PortXmitWastedBW Counter

This counter indicates the number of *flit times* where one or more packets have been started but the transmitters are forced to send idles due to bubbles in the ingress stream. Also, the VLs that have data to be sent are not permitted to preempt the currently transmitting VL.

6.5.2 PortXmitWaitData Counter

This counter indicates the number of *flit times* where one or more packets have been started but interrupted due to bubbles in the ingress stream.

6.5.3 PortRcvBubble Counter

This counter indicates the total number of *flit times* where one or more packets have started to be received, but the receiver received idle flits from the wire.

6.6 Security

These counters reflect possible security problems in the fabric.

Security problems can occur if a PKey or SLID violation occurs at the port during the ingress or egress of a packet.

The category is calculated as the sum of the neighbor's PortRcvConstraintErrors and the local port's PortXmitConstraintErrors.

6.6.1 PortRcvConstraintErrors

This counter is incremented when partition key or source LID violations are detected in a received packet, indicating a possible security issue or misconfiguration of device security settings.

6.6.2 PortXmitConstraintErrors

This counter is incremented when partition key violations are detected in a packet attempting to be transmitted, indicating a possible security issue or misconfiguration of device security settings.

6.7 Routing

These counters reflect possible routing issues. When a routing issue occurs, the offending packet is dropped.

A typical cause of this error is the routing to a wrong egress port or an improper Service Channel (SC) mapping. These errors can be a side effect of a port or device going down while traffic was still in flight to or through the given port or device.

6.7.1 PortRcvSwitchRelayErrors

This counter indicates the number of packets that were dropped due to internal routing errors. It indicates possible misconfiguration of a switch by the SM.

6.8 Other

These counters do not fit into any of the previous categories.

6.8.1 PortRcvRemotePhysicalErrors

This counter indicates the number of downstream effects of signal integrity (SI) problems. It indicates an SI issue in the upstream path.

This counter was not included as it does not directly indicate the link that had the issue, so it can be misleading.

6.8.2 PortXmitDiscards

This counter indicates the number of packets dropped due to several reasons including timeouts and improper packet lengths.

Note: This counter is a super set that includes Congestion Discards counter.

Appendix A Map of Intel® Omni-Path Architecture Commands

The following table maps certain InfiniBand* and Intel® True Scale commands to corresponding Intel® Omni-Path Architecture commands. It is not a complete list of commands.

Table 6. Map of InfiniBand*, Intel® True Scale, and Intel® OPA Commands

InfiniBand*	Intel® True Scale	Intel® OPA
ibstat	ibstat	opainfo ibstat *Note:* Use opainfo for more complete information.
ibv_devinfo	ibv_devinfo	ibv_devinfo *Note:* MTU >=4K reports as 4K.
ibstatus	ibstatus	opainfo
ibportstate	ibportstate iba_portconfig	opaportconfig
ibdiagnet	iba_report -o all	opareport -o all
iblinkinfo	iba_report -o links	opareport -o links
ibnetdiscover	iba_report -o links	opareport -o links
ibnodes	iba_report -d 1	opareport -d 1
ibhosts	iba_report -d 1	opareport -d 1
ibswitches	iba_report -d 1	opareport -d 1
sminfo	fabric_info	opafabricinfo opareport -d 1
ibqueryerrors -r	iba_report -o errors	opareport -o errors
ibqueryerrors -k	iba_report -o none -C	opareport -o none -C
ibtracert	iba_report -o route -S node:x -D node:y	opareport -o route -S node:x -D node:y
ibroute	iba_report -o linear [-o mcast]	opareport -o linear [-o mcast]
perfquery ibclearerrors	iba_extract_stat iba_extract_stat2 iba_report -o nodes -s -d 10	opaextractstat opaextractstat2 opareport -o nodes -s -d 10
ibping	ibping	ibping
ibcheckwidth	iba_report -o slowlinks	opareport -o slowlinks