

# **Intel® Omni-Path Architecture Management API**

**Programmer's Guide**

---

***Rev. 8.0***

***January 2020***



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.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at [intel.com](https://www.intel.com).

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 © 2017–2020, Intel Corporation. All rights reserved.



## Revision History

Date	Revision	Description
January 2020	8.0	Updated links to Intel® Omni-Path documentation set in <a href="#">Intel® Omni-Path Documentation Library</a> .
October 2019	7.0	Updates to this document include: <ul style="list-style-type: none"> <li>Updated Preface to include new <a href="#">Best Practices</a>.</li> <li>Added Performance information to all PA Functions.</li> <li>Added new PA functions: <ul style="list-style-type: none"> <li><a href="#">omgt_pa_get_group_list2</a></li> <li><a href="#">omgt_pa_get_vf_list2</a></li> <li><a href="#">omgt_pa_release_group_list2</a></li> <li><a href="#">omgt_pa_release_vf_list2</a></li> </ul> </li> </ul>
June 2019	6.0	Updates to this document include: <ul style="list-style-type: none"> <li>Added new section: Optimized Accessing of Historical PA Image Data</li> <li>Updated Protocol Attribute Definitions to include a link to the specification.</li> </ul>
April 2019	5.0	Updates to this document include: <ul style="list-style-type: none"> <li>Updated Building the Programs to correct the CFLAGS line in the sample code.</li> </ul>
September 2018	4.0	Updates to this document include: <ul style="list-style-type: none"> <li>Updated SA Interface functions to change "int * num_records" to "int32_t * num_records".</li> <li>Added new functions: <ul style="list-style-type: none"> <li><a href="#">omgt_sa_get_devicegroupname_records</a></li> <li><a href="#">omgt_sa_get_devicegroupmember_records</a></li> <li><a href="#">omgt_sa_get_devicetreemember_records</a></li> <li><a href="#">omgt_sa_get_scvlr_table_records</a></li> </ul> </li> <li>Added new QUERY_INPUT_TYPE in <a href="#">omgt_sa_selector_t</a>.</li> </ul>
April 2018	3.0	Updates to this document include: <ul style="list-style-type: none"> <li>Added a new Preface section.</li> <li>Updated the Introduction section.</li> <li>Updated the Compatibility section.</li> <li>Updated the Client Communication Model section.</li> <li>Updated the Logging section.</li> <li>Updated the Asynchronous Trap Notifications section.</li> <li>Updated <a href="#">omgt_port_get_ipv4_addr</a>; updated the description for OMGT_STATUS_INVALID_STATE.</li> <li>Updated <a href="#">omgt_port_get_ipv6_addr</a>; updated the description for OMGT_STATUS_INVALID_STATE.</li> <li>Updated <a href="#">omgt_sa_get_notice_report</a>; updated the description for the port and context parameters; deleted OMGT_STATUS_INVALID_STATE from the Returns section.</li> <li>Added a note to <a href="#">omgt_pa_get_port_stats</a> to indicate that this function has been deprecated.</li> <li>Added a note to <a href="#">omgt_pa_get_vf_port_stats</a> to indicate that this function has been deprecated.</li> </ul>

continued...



Date	Revision	Description
		<ul style="list-style-type: none"><li>Added these PA Interface functions:<ul style="list-style-type: none"><li>omgt_pa_get_group_nodeinfo</li><li>omgt_pa_get_group_linkinfo</li><li>omgt_pa_release_group_nodeinfo</li><li>omgt_pa_release_group_linkinfo</li></ul></li><li>Updated code example in saquery.</li></ul>
October 2017	2.0	<p>Updates to this document include:</p> <ul style="list-style-type: none"><li>Added cost matrix query, asynchronous notifications, and configurable timeouts.</li><li>Added the following new features:<ul style="list-style-type: none"><li>SwitchCost Record query: An SA query that returns data an OPA FM uses when making routing decisions in a fabric. See omgt_sa_get_switchinfo_records.</li><li>Asynchronous Trap Subscriptions: Users can now subscribe to certain events that happen in a fabric, such as the appearance or disappearance of nodes. See Asynchronous Trap Notifications.</li><li>Methods to check/refresh states of SA/PA:<ul style="list-style-type: none"><li>omgt_port_get_pa_service_state</li><li>omgt_port_get_sa_service_state</li></ul></li></ul></li><li>Updated saquery sample to set errorcode to 2 if unsuccessful in query.</li><li>Added new code samples to demonstrate the new Switch Cost Record and Asynchronous Trap additions. See Sample Programs.</li><li>Added missing omgt_sa_get_scvInt_table_records declaration. Version 0.1 of OPAMGT omits this declaration, which yields a warning with this function.</li></ul>
August 2017	1.0	Initial release of document.



## Contents

---

<b>Revision History.....</b>	<b>3</b>
<b>Preface.....</b>	<b>9</b>
Intended Audience.....	9
Intel® Omni-Path Documentation Library.....	9
How to Search the Intel® Omni-Path Documentation Set.....	11
Cluster Configurator for Intel® Omni-Path Fabric.....	12
Documentation Conventions.....	12
Best Practices.....	13
License Agreements.....	13
Technical Support.....	13
<b>1.0 Introduction.....</b>	<b>14</b>
<b>2.0 Intel® Omni-Path Architecture Management API.....</b>	<b>15</b>
2.1 Compatibility.....	15
2.2 Dependency.....	15
2.3 Client Communication Model.....	15
2.3.1 In-Band Communication.....	15
2.3.2 Out-of-Band Communication.....	16
2.3.3 Client Timeouts and Retries.....	17
2.4 Logging.....	17
2.5 Asynchronous Trap Notifications.....	17
2.6 Optimized Accessing of Historical PA Image Data.....	17
2.6.1 Querying Using the STL_PA_IMAGE_ID_DATA Structure.....	17
2.6.2 Querying Multiple Attributes from a Single Image.....	18
2.6.3 Querying Multiple Attributes from Multiple Images.....	19
<b>3.0 Functional Documentation.....</b>	<b>20</b>
3.1 Initialization and Maintenance.....	20
3.1.1 Data Structures.....	20
3.1.2 Defines.....	22
3.1.3 Initialization and Maintenance Functions.....	24
3.1.4 General Port Accessor Functions.....	27
3.1.5 In-Band Port Accessor Functions.....	29
3.1.6 Out-of-Band Port Accessor Functions.....	34
3.1.7 General Functions.....	36
3.2 SA Interface.....	37
3.2.1 Data Structures.....	37
3.2.2 Functions.....	38
3.3 Async Notification SA Interface.....	79
3.3.1 Functions.....	79
3.4 PA Interface.....	81
3.4.1 Defines.....	81
3.4.2 Functions.....	81
<b>4.0 Protocol Attribute Definitions.....</b>	<b>100</b>



<b>5.0 Sample Programs.....</b>	<b>101</b>
5.1 Prerequisites.....	101
5.2 Building the Programs.....	101
5.3 saquery.....	102
5.4 paquery.....	103



## Figures

1	In-Band Communication Model.....	16
2	Out-of-Band Communication Model.....	16



## **Tables**

1	API Header File Summary.....	20
---	------------------------------	----





## 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 the next generation, High-Performance Computing (HPC) network solution that is designed to cost-effectively meet the growth, density, and reliability requirements of large-scale HPC clusters.

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

## Intended Audience

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

## Intel® Omni-Path Documentation Library

Intel® Omni-Path publications are available at the following URL, under *Latest Release Library*:

<https://www.intel.com/content/www/us/en/design/products-and-solutions/networking-and-io/fabric-products/omni-path/downloads.html>

Use the tasks listed in this table to find the corresponding Intel® Omni-Path document.

Task	Document Title	Description
Using the Intel® OPA documentation set	<i>Intel® Omni-Path Fabric Quick Start Guide</i>	A roadmap to Intel's comprehensive library of publications describing all aspects of the product family. This document outlines the most basic steps for getting your Intel® Omni-Path Architecture (Intel® OPA) cluster installed and operational.
Setting up an Intel® OPA cluster	<i>Intel® Omni-Path Fabric Setup Guide</i>	Provides a high level overview of the steps required to stage a customer-based installation of the Intel® Omni-Path Fabric. Procedures and key reference documents, such as Intel® Omni-Path user guides and installation guides, are provided to clarify the process. Additional commands and best known methods are defined to facilitate the installation process and troubleshooting.
<i>continued...</i>		



Task	Document Title	Description
Installing hardware	<i>Intel® Omni-Path Fabric Switches Hardware Installation Guide</i>	Describes the hardware installation and initial configuration tasks for the Intel® Omni-Path Switches 100 Series. This includes: Intel® Omni-Path Edge Switches 100 Series, 24 and 48-port configurable Edge switches, and Intel® Omni-Path Director Class Switches 100 Series.
	<i>Intel® Omni-Path Host Fabric Interface Installation Guide</i>	Contains instructions for installing the HFI in an Intel® OPA cluster.
Installing host software Installing HFI firmware Installing switch firmware (externally-managed switches)	<i>Intel® Omni-Path Fabric Software Installation Guide</i>	Describes using a Text-based User Interface (TUI) to guide you through the installation process. You have the option of using command line interface (CLI) commands to perform the installation or install using the Linux* distribution software.
Managing a switch using Chassis Viewer GUI Installing switch firmware (managed switches)	<i>Intel® Omni-Path Fabric Switches GUI User Guide</i>	Describes the graphical user interface (GUI) of the Intel® Omni-Path Fabric Chassis Viewer GUI. This document provides task-oriented procedures for configuring and managing the Intel® Omni-Path Switch family. Help: GUI embedded help files
Managing a switch using the CLI Installing switch firmware (managed switches)	<i>Intel® Omni-Path Fabric Switches Command Line Interface Reference Guide</i>	Describes the command line interface (CLI) task information for the Intel® Omni-Path Switch family. Help: -help for each CLI
Managing a fabric using FastFabric	<i>Intel® Omni-Path Fabric Suite FastFabric User Guide</i>	Provides instructions for using the set of fabric management tools designed to simplify and optimize common fabric management tasks. The management tools consist of Text-based User Interface (TUI) menus and command line interface (CLI) commands. Help: -help and man pages for each CLI. Also, all host CLI commands can be accessed as console help in the Fabric Manager GUI.
Managing a fabric using Fabric Manager	<i>Intel® Omni-Path Fabric Suite Fabric Manager User Guide</i>	The Fabric Manager uses a well defined management protocol to communicate with management agents in every Intel® Omni-Path Host Fabric Interface (HFI) and switch. Through these interfaces the Fabric Manager is able to discover, configure, and monitor the fabric.
	<i>Intel® Omni-Path Fabric Suite Fabric Manager GUI User Guide</i>	Provides an intuitive, scalable dashboard and set of analysis tools for graphically monitoring fabric status and configuration. This document is a user-friendly alternative to traditional command-line tools for day-to-day monitoring of fabric health. Help: Fabric Manager GUI embedded help files
Configuring and administering Intel® HFI and IPoIB driver Running MPI applications on Intel® OPA	<i>Intel® Omni-Path Fabric Host Software User Guide</i>	Describes how to set up and administer the Host Fabric Interface (HFI) after the software has been installed. The audience for this document includes cluster administrators and Message-Passing Interface (MPI) application programmers.
Writing and running middleware that uses Intel® OPA	<i>Intel® Performance Scaled Messaging 2 (PSM2) Programmer's Guide</i>	Provides a reference for programmers working with the Intel® PSM2 Application Programming Interface (API). The Performance Scaled Messaging 2 API (PSM2 API) is a low-level user-level communications interface.
continued...		



Task	Document Title	Description
Optimizing system performance	<i>Intel® Omni-Path Fabric Performance Tuning User Guide</i>	Describes BIOS settings and parameters that have been shown to ensure best performance, or make performance more consistent, on Intel® Omni-Path Architecture. If you are interested in benchmarking the performance of your system, these tips may help you obtain better performance.
Designing an IP or LNet router on Intel® OPA	<i>Intel® Omni-Path IP and LNet Router Design Guide</i>	Describes how to install, configure, and administer an IPoIB router solution (Linux* IP or LNet) for inter-operating between Intel® Omni-Path and a legacy InfiniBand* fabric.
Building Containers for Intel® OPA fabrics	<i>Building Containers for Intel® Omni-Path Fabrics using Docker* and Singularity* Application Note</i>	Provides basic information for building and running Docker* and Singularity* containers on Linux*-based computer platforms that incorporate Intel® Omni-Path networking technology.
Writing management applications that interface with Intel® OPA	<i>Intel® Omni-Path Management API Programmer's Guide</i>	Contains a reference for programmers working with the Intel® Omni-Path Architecture Management (Intel OPAMGT) Application Programming Interface (API). The Intel OPAMGT API is a C-API permitting in-band and out-of-band queries of the FM's Subnet Administrator and Performance Administrator.
Using NVMe* over Fabrics on Intel® OPA	<i>Configuring Non-Volatile Memory Express* (NVMe*) over Fabrics on Intel® Omni-Path Architecture Application Note</i>	Describes how to implement a simple Intel® Omni-Path Architecture-based point-to-point configuration with one target and one host server.
Learning about new release features, open issues, and resolved issues for a particular release	<i>Intel® Omni-Path Fabric Software Release Notes</i>	
	<i>Intel® Omni-Path Fabric Manager GUI Release Notes</i>	
	<i>Intel® Omni-Path Fabric Switches Release Notes</i> (includes managed and externally-managed switches)	
	<i>Intel® Omni-Path Fabric Unified Extensible Firmware Interface (UEFI) Release Notes</i>	
	<i>Intel® Omni-Path Fabric Thermal Management Microchip (TMM) Release Notes</i>	
	<i>Intel® Omni-Path Fabric Firmware Tools Release Notes</i>	

## How to Search the Intel® Omni-Path Documentation Set

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

Follow these steps:

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

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



## 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 13 for more information.

For more information, visit [www.intel.com](http://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.



## **Best Practices**

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

## **License Agreements**

This software is provided under one or more license agreements. 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 <http://www.intel.com/omnipath/support> for additional detail.



## **1.0 Introduction**

---

This manual is a reference for programmers working with the Intel® Omni-Path Architecture Management (Intel® OPAMGT) Application Programming Interface (API). The Intel OPAMGT API is a C-API permitting in-band and out-of-band queries of the FM's Subnet Administrator (SA) and Performance Administrator (PA).

This document assumes you are familiar with the Intel® Omni-Path Fabric Suite including the Fabric Manager (SA/PA) and FastFabric. See the *Intel® Omni-Path Fabric Suite Fabric Manager User Guide* and the *Intel® Omni-Path Fabric Suite FastFabric User Guide* for a complete description.

For other documents in the Intel® Omni-Path Architecture product line, refer to [Intel® Omni-Path Documentation Library](#) on page 9.



## 2.0 Intel® Omni-Path Architecture Management API

---

### 2.1 Compatibility

Intel® Omni-Path Architecture Management (Intel OPAMGT) follows the Linux\* versioning convention of incrementing the major version number to indicate changes affecting API or ABI compatibility, and incrementing the minor version number to indicate other types of changes including bug fixes and additions of new APIs.

Intel OPAMGT operates with the 10.x series of Intel® Omni-Path fabric management software. Most, but not all, Subnet Administrator (SA) and Performance Administrator (PA) query types exposed in the API will be supported by all versions of the Fabric Manager. For the queries that may not be supported, the client can determine support by querying for ClassPortInfo and checking the capability masks.

### 2.2 Dependency

The following libraries are required by Intel OPAMGT:

- libibumad
- libssl
- libcrypto

### 2.3 Client Communication Model

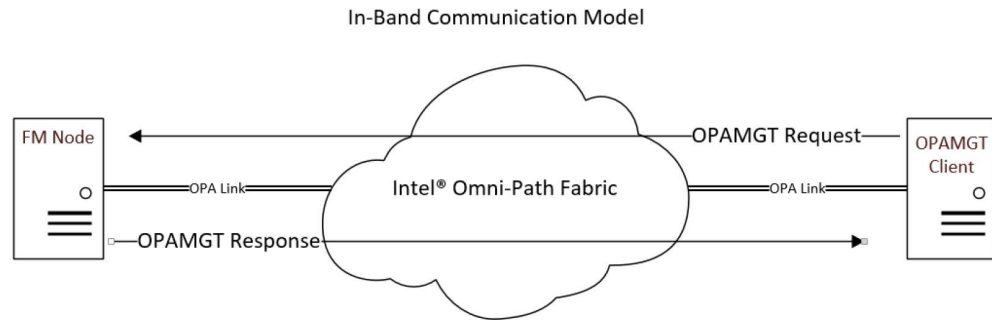
The Intel OPAMGT library can gather data using either In-Band or Out-of-Band mode. In-Band communication requires the OPAMGT user to be operating on a node connected to the Omni-Path fabric. Out-of-Band communication operates through a TCP/IP connection to the Fabric Executive (FE). This will allow the OPAMGT Out-of-Band user to connect outside the fabric. By choosing the appropriate initialization function, a connection can be set up in either mode.

See [Data Structures](#) on page 20 for all the connection initialization functions.

#### 2.3.1 In-Band Communication

In-Band communication is defined as when an OPAMGT client attached to the Intel® Omni-Path Fabric can issue requests through its HFI/port directly to the SA or PA by routing through the Fabric.

**Figure 1. In-Band Communication Model**

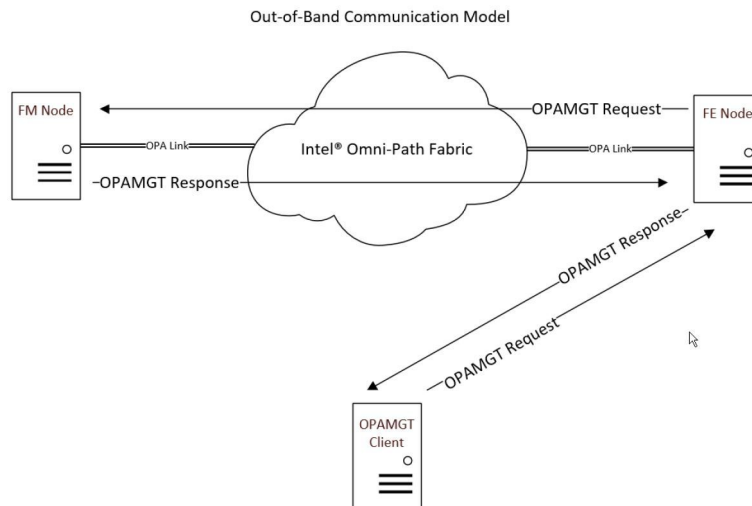


### 2.3.2 Out-of-Band Communication

Out-of-Band communication is defined as when an Intel OPAMGT client issues a request using TCP/IP protocols to the Fabric Executive (FE) process running on a node attached to the Fabric. The FE translates between In-Band and Out-of-Band communication methods to allow remote access of the fabric for the Intel OPAMGT user.

The Intel OPAMGT library supports several forms of SSL encryption to secure Out-of-Band communication. Both the FE and the client communicating through it must agree on the encryption level. See the *Intel® Omni-Path Fabric Suite Fabric Manager User Guide* for more information on how to set up and use SSL encryption.

**Figure 2. Out-of-Band Communication Model**







### 2.3.3 Client Timeouts and Retries

The expected time a query will take is dependent on several factors: fabric size, query type, and query complexity. Querying for the Subnet Administrator (SA) or Performance Administrator's (PA) ClassPortInfo should take less than a second, but querying for all the path records in the fabric takes more time, because there are more records and the SA has to gather and assemble all the data.

Timeout and retry values can be adjusted at any time after port initialization and before a query is executed. See [Initialization and Maintenance](#) on page 20 for defines and functions to change values.

## 2.4 Logging

Intel OPAMGT outputs debugging and error log messages; these can be turned off and the output destination for the messages can also be set. These options can be selected prior to the initialization of a session or they can be changed at any time while a session is active. See [omgt\\_params](#) on page 21, [omgt\\_set\\_dbg](#) on page 27, and [omgt\\_set\\_err](#) on page 28. Logging occurs on a per session basis; debug information for different sessions can be sent to different locations.

## 2.5 Asynchronous Trap Notifications

Intel OPAMGT supports in-band and limited out-of-band asynchronous trap notifications. In-band support includes registering, listening, and unregistering for traps per open port. When registering in-band, only traps that are registered to the open connection will be returned by the listener. Out of band support includes support only for listening for traps through the FE. The FE can be configured to register for traps and all open connections will receive all traps the FE is registered to listen for. After an event is registered, whether through the Intel OPAMGT library or the FE, the FM forwards that trap in an FM Notice.

A unique number, called a Trap ID, identifies an event. Some events contain additional information in common fields. Common events that can be registered include the disappearance or reappearance of a node between SM sweeps and a change in the switch cost matrix. Refer to [Async Notification SA Interface](#) on page 79 for the list of trap functions. See the *Intel® Omni-Path Fabric Suite Fabric Manager User Guide* for more information on Fabric Sweeping and Fabric Change Detection.

## 2.6 Optimized Accessing of Historical PA Image Data

Intel® OPAMGT provides several methods for accessing historical performance data through the PA. The following sections provide recommendations to follow when accessing Image Data. See [Functions](#) on page 81 for the function definitions referenced in this section.

### 2.6.1 Querying Using the STL\_PA\_IMAGE\_ID\_DATA Structure

Most PA queries will return an image ID structure that should be used for further queries when accessing the same image.



There are two main methods of using this structure to query image data. The first is to query a specific image using an ImageNumber value and an optional ImageOffset. The second is to query using an ImageTime. [Defines](#) on page 81 contains the defines to switch between these methods.

### 2.6.1.1 Querying Image Data using an ImageNumber

The ImageNumber is an opaque unsigned 64-bit identifier. ImageNumber of PACIENT\_IMAGE\_CURRENT is a special value to indicate the "current" or last successful image. The use of an ImageOffset will access the image that is ImageOffset images next to the base image at ImageNumber.

On response to most PA queries, this value will be resolved to the most direct ImageNumber that indicates the returned image. This new image ID should be used to be sure you are accessing the correct image's data.

### 2.6.1.2 Querying Image Data using an ImageTime

This feature is only available if the IsAbsTimeQuerySupported PA capability is set. If this capability is set to one, the PA will accept querying images by an Absolute Time. Absolute Time is defined as Unix Time (a.k.a. Epoch Time; the seconds since January 1, 1970). PACIENT\_IMAGE\_TIMED is a special value for ImageNumber that indicates that ImageTime will indicate a specific absolute time. When using PACIENT\_IMAGE\_TIMED the ImageOffset is ignored.

An image will be returned if the requested time falls between ImageTime and ImageTime+ImageInterval. ImageInterval is a value found in the Image Info data that represents the length of time this image's data covers. For recent images, the ImageInterval value is equal to the Pm.SweepInterval from the FM configuration file. For Short Term History (disk) Images, ImageInterval is Pm.ShortTermHistory.ImagesPerComposite multiplied by Pm.SweepInterval.

## 2.6.2 Querying Multiple Attributes from a Single Image

Through the PA interface, a user can access performance data at slices in time called Images. Often a user will need to access different data from the Image and will want to make sure each time they are accessing the same Image. There are several ways to access the data. Use the process below to gather the data from a single Image repeatedly in the most efficient and reproducible manner.

1. Freeze the requested image. This will cache the image for the amount of time specified in the Pm.FreezeFrameLease in the FM configuration file (default of 60 seconds) and will not allow it to age out. If the image is on disk, this will cache the image in memory for quicker access.

```
status = omgt_pa_freeze_image(port, request_image_id, &frozen_image_id);
```

2. Access image data. Generally, applications will want to first access the Image Info data to obtain basic data about the image, such as number of ports and nodes and the ImageInterval.

```
status = omgt_pa_get_image_info(port, frozen_image_id, &response_image_info);
```



3. Access Image Counter Data. You can now gather information, such as what PmPortGroups were active during the Image's Interval, ports under congestion, or Port Counters.

```
// Get Group List
status = omgt_pa_get_group_list2(port, frozen_image_id, &num_groups,
&response_group_list);

// Get top 10 congested links/ports
status = omgt_pa_get_group_focus(port, frozen_image_id, "All",
STL_PA_SELECT_CATEGORY_CONG, 0, 10,
&response_image_id, &num_ports, &response_focus_list);

// Get Raw PM Image Port Counters for Port 5 on Lid 3
status = omgt_pa_get_port_stats2(port, frozen_image_id, 3, 5,
&response_image_id, &response_port_counters, &response_flags,
0, 0);
```

It is worth noting that querying information from an Image may take varying amounts of time. For example, `omgt_pa_get_group_focus()` will usually take a longer time to respond than `omgt_pa_get_port_stats2()`. Higher-level (group-related) queries will have a response that will often scale with the size of the cluster and take a longer time to respond. For instance, the performance of `omgt_pa_get_group_list2()` can be expressed as  $O(G)$  where  $G$  is number of PmPortGroups, while `omgt_pa_get_group_focus()`, the most expensive query, can be expressed as  $O(p+n*\log(n))$ , where  $p$  is the number of ports in the fabric and  $n$  is the number of ports in the group. Also, `omgt_pa_get_port_stats2()` is a low-level query and can be expressed as  $O(1)$ , since it only returns a single port's counter data.

4. Renew or Release Frozen Image. A frozen image is cached for a limited time (Pm.FreezeFrameLease in the FM config). To keep an image frozen for longer, you can renew the frozen image. When the image is no longer needed, you can release it.

```
// Renew Image
status = omgt_pa_renew_image(port, &frozen_image_id);

// Release Image
status = omgt_pa_release_image(port, &frozen_image_id);
```

### 2.6.3 Querying Multiple Attributes from Multiple Images

For a large cluster, loading an image from disk can take some time. We recommend that you structure your queries to only access one image at a time, since only one image can be loaded from disk into memory at a time.

To release a completed image and to freeze a new image, the atomic move frozen image operation performs a release on the old image then freezes the new image in one query. This takes one query and is done under lock, so the new image will not age out between the release of the old image and the freeze of the new image.

```
status = omgt_pa_move_image_freeze(port, old_frozen_image_id,
&new_frozen_image_id);
```



## 3.0 Functional Documentation

**Table 1. API Header File Summary**

Category	File	Description
Primary header files exposing all supported function calls	opamgt.h	API calls to open, configure, close, and read properties of an opamgt port. Supports both in-band and out-of-band connections.
	opamgt_pa.h	Functions to query fabric performance data from the Performance Administrator (PA)
	opamgt_sa.h	Functions to query fabric configuration data from the Subnet Administrator (SA)
	opamgt_sa_notice.h	Functions to register for and receive notifications from the Subnet Administrator.
These header files expose the structure definitions for the data types used in the SA and PA Query functions. Refer to the protocol attribute definitions in <a href="#">Protocol Attribute Definitions</a> on page 100 for more information.	iba/stl_sa_types.h	Defines all structures and some helper macros for working with the SA API calls.
	iba/stl_pa_types.h	Defines all structures and some helper macros for working with the PA API calls.
	iba/stl_sd_types.h	Defines input types and values used for SA queries.
	iba/stl_mad_types.h	Defines data structures for a couple of common query types such as ClassPortInfo.
	iba/stl_types.h	Defines helper macros used internally by other opamgt headers.
Files in this directory are for internal opamgt use only and should not need to be directly included or used by the API client.	iba/public/*	Support files for internal use

### 3.1 Initialization and Maintenance

#### 3.1.1 Data Structures

##### 3.1.1.1 omgt\_port

Opaque structure defined internally to hold port connection and session information.



### Syntax

```
struct omgt_port
```

#### 3.1.1.2 omgt\_params

Configuration settings used when opening an omgt\_port. Current configuration capability is limited to log settings.

### Syntax

```
struct omgt_params {
    FILE *error_file;
    FILE *debug_file;
};
```

### Data Fields

Data Field	Description
error_file	File stream to send ERROR messages.
debug_file	File stream to send DEBUG messages.

Each FILE parameter can be either an open Linux FILE, NULL to disable, or OMGT\_DBG\_FILE\_SYSLOG to send messages to syslog.

#### 3.1.1.3 omgt\_ssl\_params

SSL input configuration options to use during initialization of an out-of-band connection.

### Syntax

```
struct omgt_ssl_params{
    uint32_t enable;
    char directory[OMGT_OOB_SSL_DIR_SIZE];
    char certificate[OMGT_OOB_SSL_FILENAME_SIZE];
    char private_key[OMGT_OOB_SSL_FILENAME_SIZE];
    char ca_certificate[OMGT_OOB_SSL_FILENAME_SIZE];
    uint32_t cert_chain_depth;
    char dh_params[OMGT_OOB_SSL_FILENAME_SIZE];
    uint32_t ca_crl_enable;
    char ca_crl[OMGT_OOB_SSL_FILENAME_SIZE];
};
```

### Data Fields

Data Field	Description
enable	Enable SSL on out-of-band connection
directory	Directory location of OpenSSL-related files
certificate	Certificate PEM file
private_key	Private key PEM file
ca_certificate	Certificate Authority (CA) certificate PEM file
<i>continued...</i>	



Data Field	Description
cert_chain_depth	Limit up to which depth certificates in a chain are used during the verification procedure. If the certificate chain is longer than allowed, the certificates above the limit are ignored.
dh_params	Diffie-Hellman parameters PEM file
ca_crl_enable	To enable/disable the usage of the CRL PEM file
ca_crl	CA CRL PEM file

#### 3.1.1.4 omgt\_oob\_input

Out-of-band input configuration options to setup connection to the FE.

##### Syntax

```
struct omgt_oob_input{
    char *host;
    uint16_t port;
    struct omgt_ssl_params ssl_params;
    int is_esm_fe;
};
```

##### Data Fields

Data Field	Description
host	FE's ipv4, ipv6, or hostname
port	TCP port the FE is listening for requests on
ssl_params	SSL parameters
is_esm_fe	Set to 1 when FE is on the ESM.

#### 3.1.2 Defines

Define	Description
OMGT_DBG_FILE_SYSLOG	Reserved FILE pointer to indicate to the logger to print to the syslog.
OMGT_OOB_SSL_DIR_SIZE	Length of full path to the directory where the SSL files are stored.
OMGT_OOB_SSL_FILENAME_SIZE	Length of the filename of any file parameters to the SSL input.
OMGT_OOB_SSL_PATH_SIZE	Length of an absolute filename including path.

#### 3.1.2.1 OMGT\_STATUS\_T

```
typedef uint32_t OMGT_STATUS_T
```

Common return value type to indicate function exit status. Each value may have a slightly different meaning depending on the function from which it returns. See function return values for a relevant description.



Define	Value
#define OMGT_STATUS_SUCCESS	0x00
#define OMGT_STATUS_ERROR	0x01
#define OMGT_STATUS_INVALID_STATE	0x02
#define OMGT_STATUS_INVALID_OPERATION	0x03
#define OMGT_STATUS_INVALID_SETTING	0x04
#define OMGT_STATUS_INVALID_PARAMETER	0x05
#define OMGT_STATUS_INSUFFICIENT_RESOURCES	0x06
#define OMGT_STATUS_INSUFFICIENT_MEMORY	0x07
#define OMGT_STATUS_COMPLETED	0x08
#define OMGT_STATUS_NOT_DONE	0x09
#define OMGT_STATUS_PENDING	0x0A
#define OMGT_STATUS_TIMEOUT	0x0B
#define OMGT_STATUS_CANCELED	0x0C
#define OMGT_STATUS_REJECT	0x0D
#define OMGT_STATUS_OVERRUN	0x0E
#define OMGT_STATUS_PROTECTION	0x0F
#define OMGT_STATUS_NOT_FOUND	0x10
#define OMGT_STATUS_UNAVAILABLE	0x11
#define OMGT_STATUS_BUSY	0x12
#define OMGT_STATUS_DISCONNECT	0x13
#define OMGT_STATUS_DUPLICATE	0x14
#define OMGT_STATUS_POLL_NEEDED	0x15

Not all status values are currently in use.

### 3.1.2.2 Service State Values

Define	Description
OMGT_SERVICE_STATE_UNKNOWN	Service is in an unknown state
OMGT_SERVICE_STATE_OPERATIONAL	Service is operational
OMGT_SERVICE_STATE_DOWN	Service cannot be contacted and is marked down
OMGT_SERVICE_STATE_UNAVAILABLE	Service is not registered and is unavailable

### 3.1.2.3 Service State Refresh Values

Define	Description
OMGT_REFRESH_SERVICE_NOP	Do not refresh service state
OMGT_REFRESH_SERVICE_BAD_STATE	Only refresh if state is not operational
OMGT_REFRESH_SERVICE_ANY_STATE	Refresh on any state



### 3.1.3 Initialization and Maintenance Functions

#### 3.1.3.1 omgt\_open\_port

Opens an in-band opamgt port using the name of the HFI device.

This function allocates and initializes a connection to the local HFI using the HFI device's name and port number. Additionally, per port object logging can be set up using session\_params.

##### Syntax

```
OMGT_STATUS_T omgt_open_port(struct omgt_port **port, char *hfi_name, uint8_t port_num, struct omgt_params *session_params)
```

##### Parameters

Parameter	Description
port	Port object is allocated and returned
hfi_name	HFI device name (e.g., "hfi1_0")
port_num	Port number of the HFI starting at 1 (0 is a wildcard meaning first active)
session_params	Parameters to open port with (e.g., logging streams)

##### Returns

OMGT_STATUS_SUCCESS	Success, port is a fully allocated initialized omgt_port.
OMGT_STATUS_INSUFFICIENT_MEMORY	Allocation of omgt_port instance failed.
OMGT_STATUS_UNAVAILABLE	Failed to initialize sub libraries (e.g., umad).
OMGT_STATUS_INVALID_PARAMETER	Failed to open physical port because either the HFI does not exist or it is not an Intel® Omni-Path-supported HFI.
OMGT_STATUS_INVALID_STATE	Failed to initialize the port's information cache lock.
OMGT_STATUS_ERROR	Failed to initialize the port's info cache data or other port data.

#### 3.1.3.2 omgt\_open\_port\_by\_num

Opens an in-band opamgt port by the enumerated HFI and port numbers.

This function allocates and initializes a connection to the local HFI using the HFI device's number and port's number. Additionally, per port object logging can be set up using session\_params.





## Syntax

```
OMGT_STATUS_T omgt_open_port_by_num(struct omgt_port **port, int32_t *hfi_num,
uint8_t port_num, struct omgt_params *session_params)
```

## Parameters

Parameter	Description
port	Port object is allocated and returned
hfi_num	HFI number based on the order of the cards starting at 1 (0 is a wildcard meaning first active)
port_num	Port number of the HFI starting at 1 (0 is a wildcard meaning first active)
session_params	Parameters to open port with (e.g., Logging streams)

## Returns

OMGT_STATUS_SUCCESS	Success, port is a fully allocated initialized omgt_port.
OMGT_STATUS_INSUFFICIENT_MEMORY	Allocation of omgt_port instance failed.
OMGT_STATUS_NOT_FOUND	No HFIs found at all or that matched input arguments.
OMGT_STATUS_UNAVAILABLE	Failed to initialize sub libraries (e.g., umad).
OMGT_STATUS_NOT_DONE	No Active HFIs when wildcard ("0") was passed into either of the parameters.
OMGT_STATUS_INVALID_PARAMETER	Failed to open physical port because either the HFI does not exist or it is not an Intel® Omni-Path supported HFI.
OMGT_STATUS_INVALID_STATE	Failed to initialize the port's information cache lock.
OMGT_STATUS_ERROR	Failed to initialize the port's info cache data or other port data.

### 3.1.3.3 omgt\_open\_port\_by\_guid

Opens an in-band opamgt port by port GUID.

This function allocates and initializes a connection to the local HFI using the HFI device's Port GUID. Additionally, per port object logging can be set up using session\_params.



## Syntax

```
OMGT_STATUS_T omgt_open_port_by_guid(struct omgt_port **port, uint64_t port_guid,
struct omgt_params *session_params)
```

## Parameters

Parameter	Description
port	Port object is allocated and returned
port_guid	Port GUID of the port
session_params	Parameters to open port with (e.g., logging streams)

## Returns

OMGT_STATUS_SUCCESS	Success, port is a fully allocated initialized omgt_port.
OMGT_STATUS_INSUFFICIENT_MEMORY	Allocation of omgt_port instance failed.
OMGT_STATUS_NOT_FOUND	No HFIs found at all or that matched input arguments.
OMGT_STATUS_UNAVAILABLE	Failed to initialize sub libraries (e.g., umad).
OMGT_STATUS_INVALID_PARAMETER	Failed to open physical port because either the HFI does not exist or it is not an Intel® Omni-Path supported HFI.
OMGT_STATUS_INVALID_STATE	Failed to initialize the port's information cache lock.
OMGT_STATUS_ERROR	Failed to initialize the port's info cache data or other port data.

### 3.1.3.4 omgt\_oob\_connect

Open an out-of-band opamgt port.

This function allocates and initializes a connection to the FE through an out-of-band interface. Additionally, per port object logging can be set up using session\_params.

## Syntax

```
OMGT_STATUS_T omgt_oob_connect(struct omgt_port **port, struct omgt_oob_input
*oob_input, struct omgt_params *session_params)
```



### Parameters

Parameter	Description
port	Port object is allocated and returned
oob_input	Out-of-band connection info, such as IP address and port of FE as well as SSL options
session_params	Parameters to open port with (e.g., logging streams)

### Returns

OMGT_STATUS_SUCCESS	Success, port is a fully allocated initialized omgt_port.
OMGT_STATUS_INSUFFICIENT_MEMORY	Allocation of omgt_port instance failed.
OMGT_STATUS_UNAVAILABLE	Failed to establish a connection to the FE host.

#### 3.1.3.5 omgt\_close\_port

Closes and frees port object.

This function closes, disconnects, and frees any previously allocated and opened connections for both in-band and out-of-band port objects.

### Syntax

```
void omgt_close_port(struct omgt_port *port)
```

### Parameters

Parameter	Description
port	Port object is cleaned up and freed

### Returns

None.

## 3.1.4 General Port Accessor Functions

#### 3.1.4.1 omgt\_set\_dbg

Sets debug logging output for an opamgt port.

Allows dynamic modification of the debug log target file. Log Settings are initially configured during port open and can be changed at any time with this function. Target file can either be a standard Linux flat FILE, NULL to disable, or OMGT\_DBG\_FILE\_SYSLOG to send debug logging to syslog.



### Syntax

```
void omgt_set_dbg(struct omgt_port *port, FILE *file)
```

### Parameters

Parameter	Description
port	Port instance to modify logging configuration
file	Target file for debug logging output

### Returns

None.

#### 3.1.4.2 omgt\_set\_err

Sets error logging output for an opamgt port.

Allows Dynamic modification of the error log target file. Log settings are initially configured during port open and can be changed at any time with this function. Target file can either be a standard Linux flat FILE, NULL to disable, or OMGT\_DBG\_FILE\_SYSLOG to send error logging to syslog.

### Syntax

```
void omgt_set_err(struct omgt_port *port, FILE *file)
```

### Parameters

Parameter	Description
port	Port instance to modify logging configuration
file	Target file for error logging output

### Returns

None.

#### 3.1.4.3 omgt\_set\_timeout

Set query timeout for an opamgt port.

Allows Dynamic modification of the query timeout value. Timeout Settings are initially configured during port open and can be changed at any time with this function.

### Syntax

```
void omgt_set_timeout(struct omgt_port *port, int ms_timeout)
```



### Parameters

Parameter	Description
port	Previously initialized port object.
ms_timeout	Timeout value in milliseconds (ms). An invalid timeout value will reset timeout to default.

### NOTE

Default timeout is defined as OMGT\_DEF\_TIMEOUT\_MS.

### Returns

None.

#### 3.1.4.4 omgt\_set\_retry\_count

Set query retry count for an opamgt port.

Allows dynamic modification of the query retry value. Retry settings are initially configured during port open and can be changed at any time with this function.

### Syntax

```
void omgt_set_retry_count(struct omgt_port *port, int retry_count)
```

### Parameters

Parameter	Description
port	Previously initialized port object
retry_count	Number of times to retry query. An invalid retry count will reset to default.

### NOTE

Default retry count is defined as OMGT\_DEF\_RETRY\_CNT.

### Returns

None.

### 3.1.5 In-Band Port Accessor Functions

#### 3.1.5.1 omgt\_port\_get\_port\_prefix

Retrieves the port's prefix for an open in-band port.

### Syntax

```
OMGT_STATUS_T omgt_port_get_port_prefix (struct omgt_port *port, uint64_t *prefix)
```



### Parameters

Parameter	Description
port	Previously initialized port object for an in-band connection
prefix	Port prefix to be returned

### Returns

OMGT_STATUS_SUCCESS	Success, value set from port data.
OMGT_STATUS_INVALID_STATE	Port mode error, port is in out-of-band mode.
OMGT_STATUS_PROTECTION	Port failed to access cache data.

#### 3.1.5.2 omgt\_port\_get\_port\_guid

Retrieves the port's GUID for an open in-band port.

### Syntax

```
OMGT_STATUS_T omgt_port_get_port_guid(struct omgt_port *port, uint64_t *guid)
```

### Parameters

Parameter	Description
port	Previously initialized port object for an in-band connection
guid	Port GUID to be returned

### Returns

OMGT_STATUS_SUCCESS	Success, value set from port data.
OMGT_STATUS_INVALID_STATE	Port mode error, port is in out-of-band mode.
OMGT_STATUS_PROTECTION	Port failed to access cache data.

#### 3.1.5.3 omgt\_port\_get\_port\_lid

Retrieves the port's LID for an open in-band port.

### Syntax

```
OMGT_STATUS_T omgt_port_get_port_lid(struct omgt_port *port, uint32_t *lid)
```



### Parameters

Parameter	Description
port	Previously initialized port object for an in-band connection
lid	Port LID to be returned

### Returns

OMGT_STATUS_SUCCESS	Success, value set from port data.
OMGT_STATUS_INVALID_STATE	Port mode error, port is in out-of-band mode.
OMGT_STATUS_PROTECTION	Port failed to access cache data.

#### 3.1.5.4 omgt\_port\_get\_hfi\_port\_num

Retrieves the HFI's port number for an open in-band port.

### Syntax

```
OMGT_STATUS_T omgt_port_get_hfi_port_num(struct omgt_port *port, uint8_t *port_num)
```

### Parameters

Parameter	Description
port	Previously initialized port object for an in-band connection
port_num	Port number to be returned (indexed starting at 1)

### Returns

OMGT_STATUS_SUCCESS	Success, value set from port data.
OMGT_STATUS_INVALID_STATE	Port mode error, port is in out-of-band mode.
OMGT_STATUS_PROTECTION	Port failed to access cache data.

#### 3.1.5.5 omgt\_port\_get\_hfi\_num

Retrieves the HFI's number for an open in-band port.

### Syntax

```
OMGT_STATUS_T omgt_port_get_hfi_num(struct omgt_port *port, int32_t *hfi_num)
```

### Parameters

Parameter	Description
port	Previously initialized port object for an in-band connection
hfi_num	HFI number to be returned (indexed starting at 1)



### Returns

OMGT\_STATUS\_SUCCESS                      Success, value set from port data.

OMGT\_STATUS\_INVALID\_STATE      Port mode error, port is in out-of-band mode.

OMGT\_STATUS\_PROTECTION              Port failed to access cache data.

#### 3.1.5.6 omgt\_port\_get\_port\_state

Retrieves the port's state for an open in-band port.

### Syntax

```
OMGT_STATUS_T omgt_port_get_port_state(struct omgt_port *port, uint8_t
*port_state)
```

### Parameters

Parameter	Description
port	Previously initialized port object for an in-band connection
port_state	Port state to be returned: <ul style="list-style-type: none"><li>• IB_PORT_DOWN: The port has not achieved LinkUp in the physical layer.</li><li>• IB_PORT_INIT: The port is in LinkUp and ready to be programmed by the FM. It cannot yet pass non-management traffic.</li><li>• IB_PORT_ARMED: The port is programmed and ready to become active.</li><li>• IB_PORT_ACTIVE: The port is fully active and capable of passing all traffic.</li></ul>

### Returns

OMGT\_STATUS\_SUCCESS                      Success, value set from port data.

OMGT\_STATUS\_INVALID\_STATE      Port mode error, port is in out-of-band mode.

OMGT\_STATUS\_PROTECTION              Port failed to access cache data.

#### 3.1.5.7 omgt\_port\_get\_hfi\_name

Retrieves the HFI's name for an open in-band port.

### Syntax

```
OMGT_STATUS_T omgt_port_get_hfi_name(struct omgt_port *port, char
hfi_name[IBV_SYSFS_NAME_MAX])
```





### Parameters

Parameter	Description
port	Previously initialized port object for an in-band connection
hfi_name	Buffer to be filled with HFI's name

### Returns

OMGT_STATUS_SUCCESS	Success, value set from port data.
OMGT_STATUS_INVALID_STATE	Port mode error, port is in out-of-band mode.
OMGT_STATUS_PROTECTION	Port failed to access cache data.

#### 3.1.5.8 omgt\_port\_get\_sa\_service\_state

Get the Port's SA service state. If refresh is triggered, this function will send an SA ClassPortInfo query with a small timeout value to quickly check the responsiveness of the SA. This get function is an alternative way to initialize the port's SA service state. The SA service state is also updated or initialized before an SA query when the state is not operational.

### Syntax

```
OMGT_STATUS_T omgt_port_get_sa_service_state(struct omgt_port *port, int
*sa_service_state, uint32_t refresh)
```

### Parameters

Parameter	Description
port	Previously initialized port object for an in-band connection
pa_service_state	PA service state to be returned
refresh	Value to possibly trigger a refresh

### Returns

OMGT_STATUS_SUCCESS	Success, value set from port data and refresh was successful.
OMGT_STATUS_INVALID_STATE	Port mode error; port is in out-of-band mode.
OMGT_STATUS_INVALID_PARAMETER	Value supplied for refresh is invalid.

#### 3.1.5.9 omgt\_port\_get\_pa\_service\_state

Get the Port's PA service state. If refresh is triggered, this function will attempt to find the PA's service record using an SA query to quickly check the responsiveness and presence of the PA. An additional SA path record query will be issued to determine how to route to the PA. This get function is an alternative way to initialize the port's PA service state. The PA service state is also updated or initialized before a PA query when the state is not operational.



## Syntax

```
OMGT_STATUS_T omgt_port_get_pa_service_state(struct omgt_port *port, int
*pa_service_state, uint32_t refresh)
```

## Parameters

Parameter	Description
port	Previously initialized port object for an in-band connection
pa_service_state	PA service state to be returned
refresh	Value to possibly trigger a refresh

## Returns

OMGT\_STATUS\_SUCCESS Success, value set from port data and refresh was successful.

OMGT\_STATUS\_INVALID\_STATE Port mode error; port is in out-of-band mode.

OMGT\_STATUS\_INVALID\_PARAMETER Value supplied for refresh is invalid.

## 3.1.6 Out-of-Band Port Accessor Functions

### 3.1.6.1 omgt\_port\_get\_ip\_version

Retrieves the IP version currently in use for the open out-of-band port.

## Syntax

```
OMGT_STATUS_T omgt_port_get_ip_version(struct omgt_port *port, uint8_t
*ip_version);
```

## Parameters

Parameter	Description
port	Previously initialized port object for an out-of-band connection
ip_version	Version of the IP protocol currently in use. Possible output is either 4 or 6.

## Returns

OMGT\_STATUS\_SUCCESS Success, value set from port data.

OMGT\_STATUS\_INVALID\_STATE Port mode error, port is in in-band mode or socket connection is not set up.

### 3.1.6.2 omgt\_port\_get\_ipv4\_addr

Retrieves the IPv4 address currently in use for the open out-of-band port.



## Syntax

```
OMGT_STATUS_T omgt_port_get_ipv4_addr(struct omgt_port *port, struct in_addr
*ipv4_addr)
```

## Parameters

Parameter	Description
port	Previously initialized port object for an out-of-band connection
ipv4_addr	IPv4 address of the connection currently in use

## Returns

OMGT\_STATUS\_SUCCESS Success, value set from port data.

OMGT\_STATUS\_INVALID\_STATE Port mode error, port is in in-band mode, socket connection is not set up, or connection is using ipv6.

### 3.1.6.3 omgt\_port\_get\_ipv6\_addr

Retrieves the IPv6 address currently in use for the open out-of-band port.

## Syntax

```
OMGT_STATUS_T omgt_port_get_ipv6_addr(struct omgt_port *port, struct in6_addr
*ipv6_addr)
```

## Parameters

Parameter	Description
port	Previously initialized port object for an out-of-band connection
ipv6_addr	IPv6 address of the connection currently in use

## Returns

OMGT\_STATUS\_SUCCESS Success, value set from port data.

OMGT\_STATUS\_INVALID\_STATE Port mode error, port is in in-band mode, socket connection is not set up, or connection is using ipv4.

### 3.1.6.4 omgt\_port\_get\_ip\_addr\_text

Retrieves the IP address currently in use for the open out-of-band port and returns it in text format in a text buffer.

## Syntax

```
OMGT_STATUS_T omgt_port_get_ip_addr_text(struct omgt_port *port, char buf[],
size_t buf_len)
```



### Parameters

Parameter	Description
port	Previously initialized port object for an out-of-band connection
buf	Buffer to store the IP address of the connection currently in use
buf_len	Size of the buffer

### Returns

OMGT_STATUS_SUCCESS	Success, value set from port data.
OMGT_STATUS_INVALID_STATE	Port mode error, port is in in-band mode or socket connection is not set up.

## 3.1.7 General Functions

These are functions where one does not use an omgt\_port object.

### 3.1.7.1 omgt\_get\_hfi\_names

Gets an array of all HFI names available on the system. HFIs will be in HFI numeric order.

#### Syntax

```
OMGT_STATUS_T omgt_get_hfi_names(char hfis[][UMAD_CA_NAME_LEN], int32_t max,
int32_t *hfi_count)
```

### Parameters

Parameter	Description
hfis	Pointer to array of size char [max][UMAD_CA_NAME_LEN]
max	Maximum number of names to return
hfi_count	The number of valid entries in hfis

### Returns

OMGT_STATUS_SUCCESS	Success, HFIs found.
OMGT_STATUS_INSUFFICIENT_MEMORY	Failure to allocate HFIs.
OMGT_STATUS_NOT_FOUND	Was not able to find any HFIs.

### 3.1.7.2 omgt\_service\_state\_totext

Converts the service state to text.

#### Syntax

```
const char* omgt_service_state_totext(int service_state)
```



### Parameters

Parameter	Description
service_state	value to convert to text

### Returns

Service state in text.

## 3.1.7.3 omgt\_status\_totext

Converts the status value to text.

### Syntax

```
const char* omgt_status_totext(OMGT_STATUS_T status);
```

### Parameters

Parameter	Description
status	OMGT_STATUS_T value to convert to text

### Returns

Status in text.

## 3.2 SA Interface

### 3.2.1 Data Structures

#### 3.2.1.1 omgt\_sa\_selector\_t

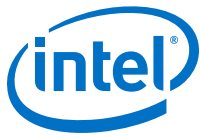
Selects input type and values to be used in SA queries. Inputs can be used to filter responses by various criteria such as LID or GUID. The desired criteria type should be specified in QUERY\_INPUT\_TYPE as shown below. The corresponding values in the OMGT\_QUERY\_INPUT\_VALUE union must be set to match the QUERY\_INPUT\_TYPE selected.

### Syntax:

```
typedef struct omgt_sa_selector {
    QUERY_INPUT_TYPE Input Value;
    OMGT_QUERY_INPUT_VALUE InputValue;
} omgt_sa_selector_t;
```

### Data Fields

Data Field	Description
QUERY_INPUT_TYPE	Type of input the query should be based on
OMGT_QUERY_INPUT_VALUE	Value for input type



## Input Types

QUERY_INPUT_TYPE	Description
InputTypeNoInput	No input - return all records
InputTypeNodeType	Match records by node type
InputTypeSystemImageGuid	Match records by system image GUID
InputTypeNodeGuid	Match records by node GUID
InputTypePortGuid	Match records by port GUID
InputTypePortGid	Match records by port GID
InputTypeMcGid	Match records by McGID
InputTypePortGuidPair	Match records by port GUID pair
InputTypeGidPair	Match records by GID pair
InputTypePathRecord	Match records by path record
InputTypeLid	Match records by LID
InputTypePKey	Match records by Pkey
InputTypeSL	Match records by SL
InputTypeIndex	Match records by a virtual fabric index
InputTypeServiceId	Match records by service ID
InputTypeNodeDesc	Match records by node description
InputTypeServiceRecord	Match records by service record
InputTypeMcMemberRecord	Match records by multicast member records
InputTypePortGuidList	Match records by port GUID list
InputTypeGidList	Match records by GID list
InputTypeMultiPathRecord	Match records by multipath record
InputTypeGeneralPair	Match records by general pair
InputTypeDeviceGroup	Match records by device group name

## 3.2.2 Functions

### 3.2.2.1 omgt\_sa\_free\_records

Free memory associated with an SA query result.

#### Syntax

```
void omgt_sa_free_records (void * record)
```

#### Parameters

Parameter	Description
record	Pointer to records returned from omgt_sa_get_* call



## Returns

None.

### 3.2.2.2 omgt\_get\_sa\_mad\_status

#### Syntax

```
uint16_t omgt_get_sa_mad_status (struct omgt_port * port)
```

#### Parameters

Parameter	Description
port	Port object to check status

## Returns

MAD Status Response code. Returns 0 if success; otherwise returns an error code.

### 3.2.2.3 omgt\_sa\_get\_buffctrl\_records

Query SA for buffer control table records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_buffctrl_records (struct omgt_port * port,
omgt_sa_selector_t * selector, int32_t * num_records,
STL_BUFFER_CONTROL_TABLE_RECORD ** records)
```

#### Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling omgt_free_query_result_buffer

## Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet



OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

### 3.2.2.4 omgt\_sa\_get\_cableinfo\_records

Query SA for cable info records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_cableinfo_records (struct omgt_port * port,  
omgt_sa_selector_t * selector, int32_t * num_records, STL_CABLE_INFO_RECORD **  
records)
```

#### Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling omgt_free_query_result_buffer

#### Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause





OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

### 3.2.2.5 omgt\_sa\_get\_classportinfo\_records

Query SA for ClassPortInfo records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_classportinfo_records (struct omgt_port * port,
omgt_sa_selector_t * selector, int32_t num_records, STL_CLASS_PORT_INFO **
records)
```

#### Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling omgt_free_query_result_buffer

#### Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error



OMGT\_STATUS\_PROTECTION Client lacks privileges necessary to perform query

OMGT\_STATUS\_ERROR Error

### 3.2.2.6 omgt\_sa\_get\_conginfo\_records

Query SA for congestion info records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_conginfo_records (struct omgt_port * port,  
omgt_sa_selector_t * selector, int32_t * num_records, STL_CONGESTION_INFO_RECORD  
** records)
```

#### Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling omgt_free_query_result_buffer

#### Returns

OMGT\_STATUS\_SUCCESS Query successful

OMGT\_STATUS\_INSUFFICIENT\_MEMORY Failed to allocate memory

OMGT\_STATUS\_INVALID\_PARAMETER Query with supplied parameters not supported

OMGT\_STATUS\_INVALID\_STATE Local port not active

OMGT\_STATUS\_NOT\_FOUND Failed to receive packet

OMGT\_STATUS\_TIMEOUT Request timed out

OMGT\_STATUS\_NOT\_DONE Send or receive failed due to timeout or some other cause

OMGT\_STATUS\_OVERRUN Receive error

OMGT\_STATUS\_REJECT umad error

OMGT\_STATUS\_PROTECTION Client lacks privileges necessary to perform query

OMGT\_STATUS\_ERROR Error



### 3.2.2.7 omgt\_sa\_get\_fabric\_info\_records

Query SA for FabricInfo records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_fabric_info_records (struct omgt_port * port,
omgt_sa_selector_t * selector, int32_t * num_records, STL_FABRICINFO_RECORD **
records)
```

#### Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling omgt_free_query_result_buffer

#### Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

### 3.2.2.8 omgt\_sa\_get\_hficong\_records

Query SA for HFI congestion records.



## Syntax

```
OMGT_STATUS_T omgt_sa_get_hficong_records (struct omgt_port * port,  
omgt_sa_selector_t * selector, int32_t * num_records,  
STL_HFI_CONGESTION_SETTING_RECORD ** records)
```

## Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling omgt_free_query_result_buffer

## Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

### 3.2.2.9 omgt\_sa\_get\_hficongctrl\_records

Query SA for HFI congestion control records.

## Syntax

```
OMGT_STATUS_T omgt_sa_get_hficongctrl_records (struct omgt_port * port,  
omgt_sa_selector_t * selector, int32_t * num_records,  
STL_HFI_CONGESTION_CONTROL_TABLE_RECORD ** records)
```



## Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records.
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling <code>omgt_free_query_result_buffer</code>

## Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

### 3.2.2.10 omgt\_sa\_get\_ib\_path\_records

Queries SA for IB path records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_ib_path_records (struct omgt_port * port,
omgt_sa_selector_t * selector, int32_t * num_records, IB_PATH_RECORD ** records)
```



### Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling omgt_free_query_result_buffer

### Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

#### 3.2.2.11 omgt\_sa\_get\_informinfo\_records

Queries SA for informinfo records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_informinfo_records (struct omgt_port * port,  
omgt_sa_selector_t * selector, int32_t * num_records, STL_INFORM_INFO_RECORD **  
records)
```



## Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling <code>omgt_free_query_result_buffer</code>

## Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

### 3.2.2.12 omgt\_sa\_get\_lfdb\_records

Queries SA for linear forwarding database records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_lfdb_records (struct omgt_port * port,
omgt_sa_selector_t * selector, int32_t * num_records,
STL_LINEAR_FORWARDING_TABLE_RECORD ** records)
```



### Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling omgt_free_query_result_buffer

### Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

#### 3.2.2.13 omgt\_sa\_get\_lid\_records

Queries SA for LID records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_lid_records (struct omgt_port * port,  
omgt_sa_selector_t * selector, int32_t * num_records, uint32_t ** lids)
```





## Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling <code>omgt_free_query_result_buffer</code>

## Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

### 3.2.2.14 omgt\_sa\_get\_link\_records

Query SA for link records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_link_records (struct omgt_port * port,
omgt_sa_selector_t * selector, int32_t * num_records, STL_LINK_RECORD ** records)
```



### Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records.
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling omgt_free_query_result_buffer

### Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

#### 3.2.2.15 omgt\_sa\_get\_mcfdb\_records

Query SA for multicast forwarding database records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_mcfdb_records (struct omgt_port * port,  
omgt_sa_selector_t * selector, int32_t * num_records,  
STL_MULTICAST_FORWARDING_TABLE_RECORD ** records)
```



## Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling <code>omgt_free_query_result_buffer</code>

## Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

### 3.2.2.16 omgt\_sa\_get\_ib\_mcmember\_records

Query SA for multicast member records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_ib_mcmember_records (struct omgt_port * port,
omgt_sa_selector_t * selector, int32_t * num_records, IB_MCMEMBER_RECORD **
records)
```



### Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling omgt_free_query_result_buffer

### Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

#### 3.2.2.17 omgt\_sa\_get\_nodeguid\_records

Query SA for node GUIDs.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_nodeguid_records (struct omgt_port * port,  
omgt_sa_selector_t * selector, int32_t * num_records, uint64_t ** guides)
```



## Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select GUIDs
num_records	Output: The number of GUIDs returned in query
guids	Output: Pointer to array of GUIDs. Must be freed by calling <code>omgt_free_query_result_buffer</code>

## Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

### 3.2.2.18 omgt\_sa\_get\_node\_records

Query SA for node records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_node_records (struct omgt_port * port,
omgt_sa_selector_t * selector, int32_t * num_records, STL_NODE_RECORD ** records)
```



### Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
ni_records	Output: Pointer to records. Must be freed by calling omgt_free_query_result_buffer

### Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

#### 3.2.2.19 omgt\_sa\_get\_nodedesc\_records

Query SA for node description records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_nodedesc_records (struct omgt_port * port,  
omgt_sa_selector_t * selector, int32_t * num_records, STL_NODE_DESCRIPTION **  
records)
```



## Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling <code>omgt_free_query_result_buffer</code>

## Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

### 3.2.2.20 omgt\_sa\_get\_pkey\_table\_records

Query SA for PKEY table records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_pkey_table_records (struct omgt_port * port,
omgt_sa_selector_t * selector, int32_t * num_records, STL_P_KEY_TABLE_RECORD **
records)
```



### Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling <code>omgt_free_query_result_buffer</code>

### Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

#### 3.2.2.21 `omgt_sa_get_portguid_records`

Query SA for port GUIDs.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_portguid_records (struct omgt_port * port,  
omgt_sa_selector_t * selector, int32_t * num_records, uint64_t ** guids)
```





## Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select GUIDs
num_records	Output: The number of GUIDs returned in query
guids	Output: Pointer to array of GUIDs. Must be freed by calling <code>omgt_free_query_result_buffer</code>

## Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

### 3.2.2.22 omgt\_sa\_get\_portgroup\_records

Query SA for PortGroup records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_portgroup_records (struct omgt_port * port,
omgt_sa_selector_t * selector, int32_t * num_records, STL_PORT_GROUP_TABLE_RECORD
** records)
```



### Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling omgt_free_query_result_buffer

### Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

#### 3.2.2.23 omgt\_sa\_get\_portgroupfwd\_records

Query SA for PortGroup forwarding table records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_portgroupfwd_records (struct omgt_port * port,
omgt_sa_selector_t * selector, int32_t * num_records,
STL_PORT_GROUP_FORWARDING_TABLE_RECORD ** records)
```



## Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling <code>omgt_free_query_result_buffer</code>

## Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

### 3.2.2.24 omgt\_sa\_get\_portinfo\_records

Query SA for portinfo records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_portinfo_records (struct omgt_port * port,
omgt_sa_selector_t * selector, int32_t * num_records, STL_PORTINFO_RECORD **
records)
```



### Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling omgt_free_query_result_buffer

### Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

#### 3.2.2.25 omgt\_sa\_get\_quarantinenode\_records

Query SA for quarantine node records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_quarantinenode_records (struct omgt_port * port,  
omgt_sa_selector_t * selector, int32_t * num_records, STL_QUARANTINED_NODE_RECORD  
** records)
```



## Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling <code>omgt_free_query_result_buffer</code>

## Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

### 3.2.2.26 omgt\_sa\_get\_scsc\_table\_records

Query SA for SCSC table records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_scsc_table_records (struct omgt_port * port,
omgt_sa_selector_t * selector, int32_t * num_records, STL_SC_MAPPING_TABLE_RECORD
** records)
```



### Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling omgt_free_query_result_buffer

### Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

#### 3.2.2.27 omgt\_sa\_get\_scsl\_table\_records

Query SA for SCSL table records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_scsl_table_records (struct omgt_port * port,
omgt_sa_selector_t * selector, int32_t * num_records,
STL_SC2SL_MAPPING_TABLE_RECORD ** records)
```



## Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records.
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling <code>omgt_free_query_result_buffer</code>

## Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

### 3.2.2.28 omgt\_sa\_get\_scvlt\_table\_records

Query SA for SCVLT table records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_scvlt_table_records (struct omgt_port * port,
omgt_sa_selector_t * selector, int32_t * num_records,
STL_SC2PVL_T_MAPPING_TABLE_RECORD ** records)
```



### Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling omgt_free_query_result_buffer

### Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

#### 3.2.2.29 omgt\_sa\_get\_service\_records

Query SA for service records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_service_records (struct omgt_port * port,  
omgt_sa_selector_t * selector, int32_t * num_records, IB_SERVICE_RECORD **  
records)
```





## Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records.
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling <code>omgt_free_query_result_buffer</code>

## Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

### 3.2.2.30 omgt\_sa\_get\_slsc\_table\_records

Query SA for SLSC table records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_slsc_table_records (struct omgt_port * port,
omgt_sa_selector_t * selector, int32_t * num_records,
STL_SL2SC_MAPPING_TABLE_RECORD ** records)
```



### Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling omgt_free_query_result_buffer

### Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

#### 3.2.2.31 omgt\_sa\_get\_sminfo\_records

Query SA for SM info records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_sminfo_records (struct omgt_port * port,  
omgt_sa_selector_t * selector, int32_t * num_records, STL_SMINFO_RECORD **  
records)
```



## Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling <code>omgt_free_query_result_buffer</code>

## Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

### 3.2.2.32 omgt\_sa\_get\_swcong\_records

Query SA for switch congestion records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_swcong_records (struct omgt_port * port,
omgt_sa_selector_t * selector, int32_t * num_records,
STL_SWITCH_CONGESTION_SETTING_RECORD ** records)
```



### Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling omgt_free_query_result_buffer

### Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

#### 3.2.2.33 omgt\_sa\_get\_switchcost\_records

Query SA for SwitchCost records.

The OPA SM keeps a record of the cost to transmit packets between each switch within the fabric. This data is used when making routing decisions. This query allows the user to request those switch to switch cost records.

See [Async Notification SA Interface](#) on page 79 for registering for notifications when this data changes.

### Syntax

```
OMGT_STATUS_T omgt_sa_get_switchcost_records (struct omgt_port * port,  
omgt_sa_selector_t * selector, int32_t * num_records, STL_SWITCH_COST_RECORD  
**records)
```



## Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling <code>omgt_free_query_result_buffer</code>

## Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

### 3.2.2.34 omgt\_sa\_get\_switchinfo\_records

Query SA for SwitchInfo records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_switchinfo_records (struct omgt_port * port,
omgt_sa_selector_t * selector, int32_t * num_records, STL_SWITCHINFO_RECORD **
records)
```



### Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling <code>omgt_free_query_result_buffer</code>

### Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

#### 3.2.2.35 `omgt_sa_get_sysimageguid_records`

Query SA for system image GUIDs.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_sysimageguid_records (structomgt_port * port,  
omgt_sa_selector_t * selector, int32_t * num_records, uint64_t ** guides)
```



## Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select GUIDs
num_records	Output: The number of GUIDs returned in query
records	Output: Pointer to GUIDs. Must be freed by calling omgt_free_query_result_buffer

## Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

### 3.2.2.36 omgt\_sa\_get\_swportcong\_records

Query SA for switch port congestion records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_swportcong_records (struct omgt_port * port,
omgt_sa_selector_t * selector, int32_t * num_records,
STL_SWITCH_PORT_CONGESTION_SETTING_RECORD ** records)
```



### Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling <code>omgt_free_query_result_buffer</code>

### Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

#### 3.2.2.37 omgt\_sa\_get\_trace\_records

Query SA for trace records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_trace_records (struct omgt_port * port,  
omgt_sa_selector_t * selector, int32_t * num_records, STL_TRACE_RECORD ** records)
```





## Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling <code>omgt_free_query_result_buffer</code>

## Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

### 3.2.2.38 omgt\_sa\_get\_vfinfo\_records

Query SA for VFInfo records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_vfinfo_records (struct omgt_port * port,
omgt_sa_selector_t * selector, int32_t * num_records, STL_VFINFO_RECORD **
records)
```



### Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling omgt_free_query_result_buffer

### Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

#### 3.2.2.39 omgt\_sa\_get\_vlarb\_records

Query SA for VL arbitration table records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_vlarb_records (struct omgt_port * port,  
omgt_sa_selector_t * selector, int32_t * num_records, STL_VLARBTABLE_RECORD **  
records)
```



## Parameters

Parameter	Description
port	Port object previously opened and initialized
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling omgt_free_query_result_buffer

## Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

### 3.2.2.40 omgt\_sa\_get\_devicegroupname\_records

Query SA for Device Group Names.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_devicegroupname_records(struct omgt_port *port,
omgt_sa_selector *selector, int32_t *num_records, STL_DEVICE_GROUP_NAME_RECORD
**records)
```



### Parameters

Parameter	Description
port	Previously initialized port object
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling omgt_free_query_result_buffer

### Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

#### 3.2.2.41 omgt\_sa\_get\_devicegroupmember\_records

Query SA for Device Group Members.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_devicegroupmember_records (struct omgt_port *port,
omgt_sa_selector *selector, int32_t *num_records, STL_DEVICE_GROUP_MEMBER_RECORD
**records)
```



## Parameters

Parameter	Description
port	Previously initialized port object
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling omgt_free_query_result_buffer

## Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

### 3.2.2.42 omgt\_sa\_get\_devicetreemember\_records

Query SA for Device Tree Records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_devicetreemember_records (struct omgt_port *port,
omgt_sa_selector *selector, int32_t *num_records, STL_DEVICE_TREE_MEMBER_RECORD
**records)
```



### Parameters

Parameter	Description
port	Previously initialized port object
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling omgt_free_query_result_buffer

### Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

#### 3.2.2.43 omgt\_sa\_get\_scvlr\_table\_records

Query SA for SC to VLr Mapping Table Records.

#### Syntax

```
OMGT_STATUS_T omgt_sa_get_scvlr_table_records (struct omgt_port *port,  
omgt_sa_selector *selector, int32_t *num_records, STL_SC2PVL_R_MAPPING_TABLE  
_RECORD **records)
```



## Parameters

Parameter	Description
port	Previously initialized port object
selector	Criteria to select records
num_records	Output: The number of records returned in query
records	Output: Pointer to records. Must be freed by calling omgt_free_query_result_buffer

## Returns

OMGT_STATUS_SUCCESS	Query successful
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate memory
OMGT_STATUS_INVALID_PARAMETER	Query with supplied parameters not supported
OMGT_STATUS_INVALID_STATE	Local port not active
OMGT_STATUS_NOT_FOUND	Failed to receive packet
OMGT_STATUS_TIMEOUT	Request timed out
OMGT_STATUS_NOT_DONE	Send or receive failed due to timeout or some other cause
OMGT_STATUS_OVERRUN	Receive error
OMGT_STATUS_REJECT	umad error
OMGT_STATUS_PROTECTION	Client lacks privileges necessary to perform query
OMGT_STATUS_ERROR	Error

## 3.3 Async Notification SA Interface

### 3.3.1 Functions

#### 3.3.1.1 omgt\_sa\_register\_trap

Initiates a registration for the specified trap.

#### Syntax

```
OMGT_STATUS_T omgt_sa_register_trap(struct omgt_port *port, uint16_t trap_num, void *context)
```



### Parameters

Parameter	Description
port	Previously initialized port object for an in-band connection.
trap_num	Trap number to register.
context	Optional opaque info to be returned when trap is received.

### Returns

OMGT_STATUS_SUCCESS	Success, trap was registered.
OMGT_STATUS_INVALID_STATE	Port mode error; port is in out-of-band mode.
OMGT_STATUS_ERROR	Failed to allocate, lock, or register trap.

#### 3.3.1.2 omgt\_sa\_unregister\_trap

Unregisters for the specified trap.

### Syntax

```
OMGT_STATUS_T omgt_sa_unregister_trap(struct omgt_port *port, uint16_t trap_num)
```

### Parameters

Parameter	Description
port	Previously initialized port object for an in-band connection.
trap_num	Trap number to unregister.

### Returns

OMGT_STATUS_SUCCESS	Success, trap was unregistered.
OMGT_STATUS_INVALID_STATE	Port mode error; port is in out-of-band mode.
OMGT_STATUS_ERROR	Failed to unregister trap.

#### 3.3.1.3 omgt\_sa\_get\_notice\_report

Gets the notice report forwarded from the SM.

### Syntax

```
OMGT_STATUS_T omgt_sa_get_notice_report(struct omgt_port *port, STL_NOTICE  
**notice, size_t *notice_len, void **context, int poll_timeout_ms)
```





## Parameters

Parameter	Description
port	Previously initialized port object.
notice	Pointer to notice structure that is allocated and returned. Must be freed by user.
notice_len	Pointer to length of notice structure to be returned. Length should always be greater than or equal to the sizeof(STL_NOTICE). All bytes that exist greater than sizeof(STL_NOTICE) are for ClassData (i.e., ClassData Length = notice_len - sizeof(STL_NOTICE)).
context	Pointer to registration context value returned. Not supported for out of band connection.
poll_timeout_ms	Length of time this function will poll (wait) for a notice report to be received in milliseconds (-1 will block indefinitely, 0 will not block, and X > 0 will block for X).

## Returns

OMGT_STATUS_SUCCESS	Success, notice was allocated and returned.
OMGT_STATUS_TIMEOUT	Poll operation timed out.
OMGT_STATUS_INSUFFICIENT_MEMORY	Failed to allocate notice buffer.
OMGT_STATUS_DISCONNECT	Registration for traps failed and will need to be redone.
OMGT_STATUS_ERROR	Poll operation failed, read operation failed, or message received was unknown.

## 3.4 PA Interface

### 3.4.1 Defines

Define	Description
PACIENT_IMAGE_CURRENT	Used to request PA data from the most recent fabric sweep
PACIENT_IMAGE_TIMED	Used to request PA data from a sweep at a given time

### 3.4.2 Functions

#### 3.4.2.1 omgt\_get\_pa\_mad\_status

Get MAD status code from most recent PA operation.

#### Syntax

```
uint16_t omgt_get_pa_mad_status (struct omgt_port * port)
```



### Parameters

Parameter	Description
port	Port object previously opened and initialized

### Returns

The corresponding status code.

### Performance

O(1)

#### 3.4.2.2 omgt\_pa\_get\_classportinfo

Query PA for ClassPortInfo for given port.

### Syntax

```
OMGT_STATUS_T omgt_pa_get_classportinfo (struct omgt_port * port,  
STL_CLASS_PORT_INFO ** pm_cpi)
```

### Parameters

Parameter	Description
port	Port object previously opened and initialized
pm_cpi	A pointer to the ClassPortInfo. The caller must free it after use.

### Returns

OMGT_STATUS_SUCCESS	Renew successful
OMGT_STATUS_ERROR	Error
OMGT_STATUS_INSUFFICIENT_RESOURCES	Memory allocation failure

### Performance

O(1)

#### 3.4.2.3 omgt\_pa\_freeze\_image

Freeze specified image.

### Syntax

```
OMGT_STATUS_T omgt_pa_freeze_image (struct omgt_port * port, STL_PA_IMAGE_ID_DATA  
pm_image_id_query, STL_PA_IMAGE_ID_DATA * pm_image_id_resp)
```



### Parameters

Parameter	Description
port	Port object previously opened and initialized
pm_image_id_query	Image ID of image to freeze
pm_image_id_resp	Pointer to image ID of image frozen.

### Returns

OMGT\_STATUS\_SUCCESS Freeze successful

OMGT\_STATUS\_ERROR Error

### Performance

$O(1)$

#### 3.4.2.4 omgt\_pa\_get\_group\_config

Get group config info.

### Syntax

```
OMGT_STATUS_T omgt_pa_get_group_config (struct omgt_port * port,
    STL_PA_IMAGE_ID_DATA pm_image_id_query, char * group_name, STL_PA_IMAGE_ID_DATA *
    pm_image_id_resp, uint32 * pNum_ports, STL_PA_PM_GROUP_CFG_RSP ** pm_group_config)
```

### Parameters

Parameter	Description
port	Port object previously opened and initialized
pm_image_id_query	Image ID of group config to get
group_name	Pointer to group name
pm_image_id_resp	Pointer to image ID of group info returned
pNum_ports	Pointer to the number of port records returned by query
pm_group_config	Pointer to group config to fill. Upon successful return, a memory to contain the group config is allocated. The caller must call pa_client_release_group_config to free the memory later.

### Returns

OMGT\_STATUS\_SUCCESS Get successful

OMGT\_STATUS\_ERROR Error

### Performance

$O(n)$ , where  $n$  is the number of ports in the fabric



### 3.4.2.5 omgt\_pa\_get\_group\_focus

Get group focus portlist.

#### Syntax

```
OMGT_STATUS_T omgt_pa_get_group_focus (struct omgt_port * port,  
STL_PA_IMAGE_ID_DATA pm_image_id_query, char * group_name, uint32 select, uint32  
start, uint32 range, STL_PA_IMAGE_ID_DATA * pm_image_id_resp, uint32 *  
pNum_ports, STL_FOCUS_PORTS_RSP ** pm_group_focus)
```

#### Parameters

Parameter	Description
port	Port object previously opened and initialized
pm_image_id_query	Image ID of group focus portlist to get
group_name	Pointer to group name
select	Select value for focus portlist
start	Start index value of portlist
range	Index range of portlist
pm_image_id_resp	Pointer to image ID of group focus portlist returned
pNum_ports	Pointer to the number of port records returned by query
pm_group_focus	Pointer to pointer to focus portlist to fill. Upon successful return, a memory to contain the group focus portlist is allocated. The caller must call pa_client_release_group_focus to free the memory later.

#### Returns

OMGT\_STATUS\_SUCCESS    Get successful

OMGT\_STATUS\_ERROR      Error

#### Performance

$O(p \cdot \log(p))$ , where  $p$  is the number of ports in the group

### 3.4.2.6 omgt\_pa\_get\_group\_info

Get group info.

#### Syntax

```
OMGT_STATUS_T omgt_pa_get_group_info (struct omgt_port * port,  
STL_PA_IMAGE_ID_DATA pm_image_id_query, char * group_name, STL_PA_IMAGE_ID_DATA *  
pm_image_id_resp, STL_PA_PM_GROUP_INFO_DATA * pm_group_info)
```



### Parameters

Parameter	Description
port	Port object previously opened and initialized
pm_image_id_query	Image ID of group info to get
group_name	Pointer to group name
pm_image_id_resp	Pointer to image ID of group info returned
pm_group_info	Pointer to group info to fill

### Returns

OMGT\_STATUS\_SUCCESS Get successful

OMGT\_STATUS\_ERROR Error

### Performance

$O(n)$ , where  $n$  is the number of ports in the fabric

#### 3.4.2.7 omgt\_pa\_get\_group\_list

Get list of group names.

### Syntax

```
OMGT_STATUS_T omgt_pa_get_group_list (struct omgt_port * port, uint32 *
pNum_Groups, STL_PA_GROUP_LIST ** pm_group_list)
```

### Parameters

Parameter	Description
port	Port object previously opened and initialized
pm_group_list	Pointer to group list to fill

### Returns

OMGT\_STATUS\_SUCCESS Get successful

OMGT\_STATUS\_ERROR Error

### Performance

$O(g)$ , where  $g$  is the number of groups defined and enabled

### NOTE

See also [omgt\\_pa\\_get\\_group\\_list2](#)



### 3.4.2.8 omgt\_pa\_get\_group\_list2

Get list of group names that are active during an Image.

#### Syntax

```
OMGT_STATUS_T omgt_pa_get_group_list2(struct omgt_port *port,  
STL_PA_IMAGE_ID_DATA pm_image_id_query, uint32_t *pNum_Groups, STL_PA_GROUP_LIST2  
**pm_group_list)
```

#### Parameters

Parameter	Description
port	Port object previously opened and initialized
pm_image_id_query	Image ID of group list to get
pNum_Groups	Pointer to the number of group records returned by query
pm_group_list	Pointer to group list to fill

#### Returns

OMGT\_STATUS\_SUCCESS   Get successful

OMGT\_STATUS\_ERROR     Error

#### Performance

$O(g)$ , where  $g$  is the number of groups defined and enabled

### 3.4.2.9 omgt\_pa\_get\_image\_info

Get image info.

#### Syntax

```
OMGT_STATUS_T omgt_pa_get_image_info (struct omgt_port *port,  
STL_PA_IMAGE_ID_DATA pm_image_id, STL_PA_IMAGE_INFO_DATA * pm_image_info)
```

#### Parameters

Parameter	Description
port	Port object previously opened and initialized
pm_image_id	Image ID of image info to get
pm_image_info	Pointer to image info to fill

#### Returns

OMGT\_STATUS\_SUCCESS   Get successful

OMGT\_STATUS\_ERROR     Error



## Performance

O (1)

### 3.4.2.10 omgt\_pa\_get\_pm\_config

Get PM configuration data.

## Syntax

```
OMGT_STATUS_T omgt_pa_get_pm_config (struct omgt_port * port, STL_PA_PM_CFG_DATA * pm_config)
```

## Parameters

Parameter	Description
port	Port object previously opened and initialized
pm_config	Pointer to PM configuration data to fill

## Returns

OMGT\_STATUS\_SUCCESS Get successful

OMGT\_STATUS\_ERROR Error

## Performance

O (1)

### 3.4.2.11 omgt\_pa\_get\_port\_stats

## NOTE

Deprecated, use omgt\_pa\_get\_port\_stats2.

### 3.4.2.12 omgt\_pa\_get\_port\_stats2

Get port statistics (counters).

## Syntax

```
OMGT_STATUS_T omgt_pa_get_port_stats2 (struct omgt_port * port, STL_PA_IMAGE_ID_DATA pm_image_id_query, STL_LID lid, uint8 port_num, STL_PA_IMAGE_ID_DATA * pm_image_id_resp, STL_PORT_COUNTERS_DATA * port_counters, uint32* flags, uint32 delta, uint32 user_cntrs)
```

## Parameters

Parameter	Description
port	Port object previously opened and initialized
pm_image_id_query	Image ID of port counters to get
<i>continued...</i>	



Parameter	Description
lid	LID of node
port_num	Port number
pm_image_id_resp	Pointer to image ID of port counters returned
port_counters	Pointer to port counters to fill
flags	Pointer to flags
delta	1 for delta counters, 0 for raw image counters
user_cntrs	1 for running counters, 0 for image counters (delta must be 0)

### Returns

OMGT\_STATUS\_SUCCESS    Get successful

OMGT\_STATUS\_ERROR    Error

### Performance

O(1)

#### 3.4.2.13 omgt\_pa\_get\_vf\_config

Get virtual fabric configuration information.

### Syntax

```
OMGT_STATUS_T omgt_pa_get_vf_config (struct omgt_port * port,  
STL_PA_IMAGE_ID_DATA pm_image_id_query, char * vf_name, STL_PA_IMAGE_ID_DATA *  
pm_image_id_resp, uint32 * pNum_ports, STL_PA_VF_CFG_RSP ** pm_vf_config)
```

### Parameters

Parameter	Description
port	Port object previously opened and initialized
pm_image_id_query	Image ID of vf info to get
vf_name	Pointer to vf name
pm_image_id_resp	Pointer to image ID of vf info returned
pNum_ports	Pointer to the number of port records returned by query
pm_vf_config	Pointer to VF configuration to fill

### Returns

OMGT\_STATUS\_SUCCESS    Get successful

OMGT\_STATUS\_ERROR    Error





## Performance

$O(n)$ , where  $n$  is the number of ports in the fabric

### 3.4.2.14 omgt\_pa\_get\_vf\_focus

Get virtual fabric focus portlist.

## Syntax

```
OMGT_STATUS_T omgt_pa_get_vf_focus (struct omgt_port * port, STL_PA_IMAGE_ID_DATA
pm_image_id_query, char* vf_name, uint32 select, uint32 start, uint32 range,
STL_PA_IMAGE_ID_DATA * pm_image_id_resp, uint32 * pNum_ports,
STL_PA_VF_FOCUS_PORTS_RSP ** pm_vf_focus)
```

## Parameters

Parameter	Description
port	Port object previously opened and initialized
pm_image_id_query	Image ID of vf focus portlist to get
vf_name	Pointer to vf name
select	Select value for focus portlist
start	Start index value of portlist
range	Index range of portlist
pm_image_id_resp	Pointer to image ID of vf info returned
pNum_ports	Pointer to the number of port records returned by query
pm_vf_focus	Pointer to vf info to fill

## Returns

OMGT\_STATUS\_SUCCESS    Get successful

OMGT\_STATUS\_ERROR      Error

## Performance

$O(p \log(p))$ , where  $p$  is the number of ports in the group

### 3.4.2.15 omgt\_pa\_get\_vf\_info

Get virtual fabric information.

## Syntax

```
OMGT_STATUS_T omgt_pa_get_vf_info(struct omgt_port *port, STL_PA_IMAGE_ID_DATA
pm_image_id_query, char *vf_name, STL_PA_IMAGE_ID_DATA *pm_image_id_resp,
STL_PA_VF_INFO_DATA *pm_vf_info);
```



### Parameters

Parameter	Description
port	Port object previously opened and initialized
pm_image_id_query	Image ID of vf info to get
vf_name	Pointer to vf name
pm_image_id_resp	Pointer to image ID of vf info returned
pm_vf_info	Pointer to vf info to fill

### Returns

OMGT\_STATUS\_SUCCESS    Get successful

OMGT\_STATUS\_ERROR      Error

### Performance

$O(n)$ , where  $n$  is the number of ports in the fabric

#### 3.4.2.16 omgt\_pa\_get\_vf\_list

Get list of vf names.

### Syntax

```
OMGT_STATUS_T omgt_pa_get_vf_list (struct omgt_port port, uint32_t pNum_VFs,  
STL_PA_VF_LIST **pm_vf_list)
```

### Parameters

Parameter	Description
port	Port object previously opened and initialized
pNum_VFs	Pointer to the number of VF records returned by query
pm_vf_list	Pointer to vf list to fill

### Returns

OMGT\_STATUS\_SUCCESS    Get successful

OMGT\_STATUS\_ERROR      Error

### Performance

$O(g)$ , where  $g$  is the number of vfs defined and enabled

### NOTE

See also [omgt\\_pa\\_get\\_vf\\_list2](#)



### 3.4.2.17 omgt\_pa\_get\_vf\_list2

Get list of vf names that are active during an Image.

#### Syntax

```
OMGT_STATUS_T omgt_pa_get_vf_list2(struct omgt_port *port, STL_PA_IMAGE_ID_DATA
pm_image_id_query, uint32_t *pNum_VFs, STL_PA_VF_LIST2 **pm_vf_list)
```

#### Parameters

Parameter	Description
port	Port object previously opened and initialized
pm_image_id_query	Image ID of vf list to get
pNum_VFs	Pointer to the number of VF records returned by query
pm_vf_list	Pointer to vf list to fill

#### Returns

OMGT\_STATUS\_SUCCESS Get successful

OMGT\_STATUS\_ERROR Error

#### Performance

$O(g)$ , where  $g$  is the number of vfs defined and enabled

### 3.4.2.18 omgt\_pa\_get\_vf\_port\_stats

#### NOTE

Deprecated, use omgt\_pa\_get\_vf\_port\_stats2.

### 3.4.2.19 omgt\_pa\_get\_vf\_port\_stats2

Get vf port statistics (counters).

#### Syntax

```
OMGT_STATUS_T omgt_pa_get_vf_port_stats2 (struct omgt_port * port,
STL_PA_IMAGE_ID_DATA pm_image_id_query, char * vf_name, STL_LID lid, uint8
port_num, STL_PA_IMAGE_ID_DATA * pm_image_id_resp, STL_PA_VF_PORT_COUNTERS_DATA *
vf_port_counters, uint32 * flags, uint32 delta, uint32 user_cntrs)
```

#### Parameters

Parameter	Description
port	Port object previously opened and initialized
pm_image_id_query	Image ID of port counters to get
continued...	



Parameter	Description
vf_name	Pointer to VF name
lid	LID of node
port_num	Port number
pm_image_id_resp	Pointer to image ID of port counters returned
vf_port_counters	Pointer to vf port counters to fill
flags	Pointer to flags
delta	1 for delta counters, 0 for raw image counters
user_cntrs	1 for running counters, 0 for image counters (delta must be 0)

### Returns

OMGT\_STATUS\_SUCCESS    Get successful

OMGT\_STATUS\_ERROR      Error

### Performance

O(1)

## 3.4.2.20 omgt\_pa\_move\_image\_freeze

Move freeze of image 1 to image 2.

### Syntax

```
OMGT_STATUS T omgt_pa_move_image_freeze (struct omgt_port * port,  
STL_PA_IMAGE_ID_DATA pm_image_id1, STL_PA_IMAGE_ID_DATA * pm_image_Id2)
```

### Parameters

Parameter	Description
port	Port object previously opened and initialized
pm_image_id1	Image ID of frozen image 1
pm_image_Id2	Pointer to image ID of image 2

### Returns

OMGT\_STATUS\_SUCCESS      Move image freeze successful

OMGT\_STATUS\_UNAVAILABLE   Image 2 unavailable to freeze

OMGT\_STATUS\_ERROR        Error

### Performance

O(1)



### 3.4.2.21 omgt\_pa\_get\_group\_nodeinfo

Get group node information.

#### Syntax

```
OMGT_STATUS_T omgt_pa_get_group_node_info( struct omgt_port *port,
STL_PA_IMAGE_ID_DATA pm_image_id_query, char *group_name, STL_LID nodeLid, uint64
nodeGuid, char *nodeDesc, STL_PA_IMAGE_ID_DATA *pm_image_id_resp, uint32
*pNum_nodes, STL_PA_GROUP_NODEINFO_RSP **pm_group_nodeinfo)
```

#### Parameters

Parameter	Description
port	Port object previously opened and initialized
pm_image_id_query	Image ID of group node info to get
group_name	Pointer to group name
nodeLid	LID of the node
nodeGuid	GUID of the node
nodeDesc	Description of the node
pm_image_id_resp	Pointer to image ID of group node info returned
pNum_nodes	Pointer to the number of node records returned by query
pm_group_nodeinfo	Pointer to pointer to group node info to fill. Upon successful return, a memory to contain the group node info is allocated.

#### Returns

OMGT\_STATUS\_SUCCESS Get successful

OMGT\_STATUS\_ERROR Error

#### Performance

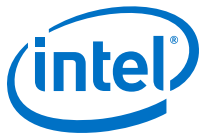
$O(n)$ , where  $n$  is the number of ports in the fabric

### 3.4.2.22 omgt\_pa\_get\_group\_linkinfo

Get group link information.

#### Syntax

```
OMGT_STATUS_T omgt_pa_get_group_linkinfo ( struct omgt_port *port,
STL_PA_IMAGE_ID_DATA pm_image_id_query, char *group_name, STL_LID inputLid, uint8
inputPort, STL_PA_IMAGE_ID_DATA *pm_image_id_resp, uint32 *pNum_links,
STL_PA_GROUP_LINKINFO_RSP **pm_group_linkinfo)
```



### Parameters

Parameter	Description
port	Port object previously opened and initialized
pm_image_id_query	Image ID of group link info to get
group_name	Pointer to group name
inputLid	LID of the input node
inputPort	Port of the input port
pm_image_id_resp	Pointer to image ID of group link info returned
pNum_ports	Pointer to the number of node records returned by query
pm_group_linkinfo	Pointer to pointer to group link info to fill. Upon successful return, a memory to contain the group link info is allocated.

### Returns

OMGT\_STATUS\_SUCCESS    Get successful

OMGT\_STATUS\_ERROR      Error

### Performance

$O(n)$ , where  $n$  is the number of ports in the fabric

#### 3.4.2.23 omgt\_pa\_release\_group\_config

Release group config information.

### Syntax

```
void omgt_pa_release_group_config (STL_PA_PM_GROUP_CFG_RSP ** pm_group_config)
```

### Parameters

Parameter	Description
pm_group_config	Pointer to pointer to the group config to free

### Returns

None

### Performance

$O(1)$

#### 3.4.2.24 omgt\_pa\_release\_group\_focus

Release group focus portlist.



### Syntax

```
void omgt_pa_release_group_focus (STL_FOCUS_PORTS_RSP ** pm_group_focus)
```

### Parameters

Parameter	Description
pm_group_focus	Pointer to the pointer to the group focus portlist to free

### Returns

None

### Performance

$O(1)$

#### 3.4.2.25 omgt\_pa\_release\_group\_list

Release group list.

### Syntax

```
void omgt_pa_release_group_list (STL_PA_GROUP_LIST ** pm_group_list)
```

### Parameters

Parameter	Description
pm_group_list	Pointer to pointer to the group list to free

### Returns

None

### Performance

$O(1)$

#### 3.4.2.26 omgt\_pa\_release\_group\_list2

Release group list.

### Syntax

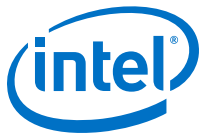
```
void omgt_pa_release_group_list2(STL_PA_GROUP_LIST2 **pm_group_list)
```

### Parameters

Parameter	Description
pm_group_list	Pointer to pointer to the group list to free

### Returns

None



## Performance

O(1)

### 3.4.2.27 omgt\_pa\_release\_image

Release specified image.

## Syntax

```
OMGT_STATUS_T omgt_pa_release_image (struct omgt_port * port,  
STL_PA_IMAGE_ID_DATA pm_image_id_query)
```

## Parameters

Parameter	Description
port	Port object previously opened and initialized
pm_image_id_query	Image ID of image to release

## Returns

OMGT\_STATUS\_SUCCESS Release successful

OMGT\_STATUS\_ERROR Error

## Performance

O(1)

### 3.4.2.28 omgt\_pa\_release\_vf\_config

Releases virtual fabric configuration information. Must be called after omgt\_pa\_get\_vf\_config to free the associated memory.

## Syntax

```
void omgt_pa_release_vf_config (STL_PA_VF_CFG_RSP ** pm_vf_config)
```

## Parameters

Parameter	Description
pm_vf_config	Pointer to the pointer to the vf configuration to free

## Returns

None

## Performance

O(1)





### 3.4.2.29 omgt\_pa\_release\_vf\_focus

Releases vf focus portlist.

#### Syntax

```
void omgt_pa_release_vf_focus (STL_PA_VF_FOCUS_PORTS_RSP ** pm_vf_focus)
```

#### Parameters

Parameter	Description
pm_vf_focus	Pointer to the pointer to vf focus portlist to free

#### Returns

None

#### Performance

$O(1)$

### 3.4.2.30 omgt\_pa\_release\_vf\_list

Release vf list.

#### Syntax

```
void omgt_pa_release_vf_list (STL_PA_VF_LIST ** pm_vf_list)
```

#### Parameters

Parameter	Description
pm_vf_list	Pointer to pointer to the vf list to free

#### Returns

None

#### Performance

$O(1)$

### 3.4.2.31 omgt\_pa\_release\_vf\_list2

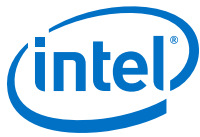
Release vf list.

#### Syntax

```
void omgt_pa_release_vf_list2 (STL_PA_VF_LIST2 **pm_vf_list)
```

#### Parameters

Parameter	Description
pm_vf_list	Pointer to pointer to the vf list to free

**Returns**

None

**Performance** $O(1)$ **3.4.2.32 omgt\_pa\_renew\_image**

Renew lease of specified image.

**Syntax**

```
OMGT_STATUS_T omgt_pa_renew_image (struct omgt_port * port, STL_PA_IMAGE_ID_DATA pm_image_id_query)
```

**Parameters**

Parameter	Description
port	Port object previously opened and initialized
pm_image_id_query	Image ID of image to renew

**Returns**

OMGT\_STATUS\_SUCCESS Renew successful

OMGT\_STATUS\_ERROR Error

**Performance** $O(1)$ **3.4.2.33 omgt\_pa\_release\_group\_nodeinfo**

Release group node information.

**Syntax**

```
void omgt_pa_release_group_nodeinfo (STL_PA_GROUP_NODEINFO_RSP ** pm_group_nodeinfo)
```

**Parameters**

Parameter	Description
pm_group_nodeinfo	Pointer to pointer to the group nodeinfo to free

**Returns**

None



## Performance

$O(1)$

### 3.4.2.34 omgt\_pa\_release\_group\_linkinfo

Release group link information.

## Syntax

```
void omgt_pa_release_group_linkinfo (STL_PA_GROUP_LINKINFO_RSP **  
pm_group_linkinfo )
```

## Parameters

Parameter	Description
pm_group_linkinfo	Pointer to pointer to the group linkinfo to free

## Returns

None

## Performance

$O(1)$



## **4.0 Protocol Attribute Definitions**

---

Refer to the [Intel® Omni-Path Specification Volume 1 Fabric Architecture Overview](https://cdrdv2.intel.com/v1/dl/getContent/611496) (<https://cdrdv2.intel.com/v1/dl/getContent/611496>) for documentation of Intel® Omni-Path management protocol attributes.



## 5.0 Sample Programs

---

This section describes two sample programs that can be used to understand basic Intel OPAMGT functionality, similar to "hello world" code. The libopamgt-devel package contains extended versions of the samples presented here showing all available requests for management data.

The full sample listing includes:

- simple\_sa\_query ([saquery](#) on page 102) - A "hello world" of how to query an OPA SA
- saquery - A sample showing all requests for subnet management information stored in an OPA SA
- simple\_pa\_query ([paquery](#) on page 103) - A "hello world" of how to query an OPA PA
- paquery - A sample showing all requests for performance management information stored in an OPA PA
- simple\_sa\_notice - A "hello world" of how to subscribe to fabric notifications
- show\_switch\_cost\_matrix - A sample showing how to request the cost data an OPA FM uses to make routing decisions
- job\_schedule - A sample showing how to determine fabric topology and schedule a job or request a number of nodes based on that topology.

### 5.1 Prerequisites

To successfully query data an Intel® Omni-Path Fabric must be available and the Intel OPAMGT library must be installed.

### 5.2 Building the Programs

The programs are found in the libopamgt-devel package and are placed in the directory `/usr/src/opamgt` upon installation. A makefile is included that will build each program (along with simpler versions of each) with the necessary flags. The file content is shown here:

```
CC=gcc
CFLAGS=-D_X86_64_ -DCPU_LE -DLINUX -DIB_STACK_OPENIB -I/usr/include/opamgt
-Wall -g

LIBS=-lopamgt \
-libverbs \
-libumad \
-lssl

SOURCES := $(shell ls *.c)

APPS := $(SOURCES:%.c=%)

.PHONY: all
```



```
all: $(APPS)

#default rule for all apps
% : %.c
$(CC) $(CFLAGS) $^ -o $@ $(LIBS)

.PHONY: clean
clean:
rm -rf $(APPS)
```

## 5.3 saquery

```
/* This file shows a simple example of requesting and printing OPA
 * management data, in this case port information.
 *
 */

// core API
#include <opamgt/opamgt.h>
// extensions for SA queries
#include <opamgt/opamgt_sa.h>

int main(int argc, char ** argv)
{
    OMGT_STATUS_T status = OMGT_STATUS_SUCCESS;
    int exitcode = 0;
    int i;
    struct omgt_port * port = NULL;
    int num_records;
    STL_PORTINFO_RECORD *pi_records = NULL;

    // create a session
    status = omgt_open_port_by_num(&port, 1 /* hfi */, 1 /* port */,
                                   NULL);
    if (OMGT_STATUS_SUCCESS != status) {
        fprintf(stderr, "failed to open port\n");
        exitcode=1;
        goto fail1;
    }

    // specify how and what we want to query by
    omgt_sa_selector_t selector;
    selector.InputType = InputTypeLid;
    selector.InputValue.PortInfoRecord.Lid = 1;

    // execute query synchronously
    status = omgt_sa_get_portinfo_records(port, &selector,
                                           &num_records,
                                           &pi_records);

    if (status != OMGT_STATUS_SUCCESS) {
        exitcode=1;
        fprintf(stderr, "failed to execute query. MadStatus=0x%x\n",
                omgt_get_sa_mad_status(port));
        goto fail2;
    }

    if (!num_records) {
        // we can check result count independent of result type
        printf("No records found.\n");
    } else {
        for (i = 0; i < num_records; ++i) {
            // the result is a set of SA records, which often follow
            // a pattern of including a RID section containing
            // top-level identification of the record, and an
            // encapsulated SM payload.
            //
        }
    }
}
```



```

        // in this case:
        //   r->RID: contains the LID and port number as record
        //   identifiers
        //   r->PortInfo: the encapsulated subnet management
        //   structure (STL_PORT_INFO)
        STL_PORTINFO_RECORD * r = &pi_records[i]; // sa
        printf("PortNum: %2u   PortLID: 0x%08x\n",
               r->RID.PortNum, r->RID.EndPortLID);
    }
}

fail2:
    // free our result buffer...
    if (pi_records) omgt_sa_free_records(pi_records);
    // ...and close our session
    omgt_close_port(port);
fail1:
    return exitcode;
}

```

## 5.4 paquery

```

/* This file shows a simple example of requesting and printing OPA
 * port counters data.
 *
 */

//core API
#include <opamgt/opamgt.h>
//extensions for PA queries
#include <opamgt/opamgt_pa.h>
#include <inttypes.h>

int main()
{
    struct omgt_port * port = NULL;
    OMTG_STATUS_T status;
    int exitcode = 0;

    // queries that take an STL_PA_IMADE_ID_DATA argument
    // can be passed this cleared image to request current data
    STL_PA_IMAGE_ID_DATA image_ID = {0};
    STL_PA_IMAGE_INFO_DATA image_info;

    // create a session
    status = omgt_open_port_by_num(&port, 1 /* hfi */, 1 /* port */,
                                   NULL);
    if(OMGT_STATUS_SUCCESS != status) {
        fprintf(stderr, "Failed to open port or initialize PA
            connection\n");
        exitcode=1;
        goto fail1;
    }

    // Request information about the image specified by image_ID
    // This returns meta information about PM sweeps such as start
    // and duration
    if (omgt_pa_get_image_info(port, image_ID, &image_info)){
        fprintf(stderr, "Failed to get PA image\n");
        exitcode=1;
        goto fail2;
    }

    printf("Sweep start: %s",
           ctime((time_t *)&image_info.sweepStart));

    STL_PORT_COUNTERS_DATA port_counters;

```



```
// Request port statistics capture in image specified by
// image_ID and store in port_counters
if (omgt_pa_get_port_stats(port, image_ID, 1 /* node LID*/,
                          1 /* port number*/, &image_ID,
                          &port_counters, NULL /* no flags */,
                          0 /* Totals */,
                          1 /* running counters */)){
    fprintf(stderr, "Failed to get port counters\n");
    exitcode=1;
    goto fail2;
}

// Display some of the data returned by the query
printf("Port Counters Data:\n");
printf("portXmitData:          %"PRIu64"\n",
       port_counters.portXmitData);
printf("portRcvData:          %"PRIu64"\n",
       port_counters.portRcvData);
printf("portXmitPkts:          %"PRIu64"\n",
       port_counters.portXmitPkts);
printf("portRcvPkts:          %"PRIu64"\n",
       port_counters.portRcvPkts);
printf("localLinkIntegrityErrors: %"PRIu64"\n",
       port_counters.localLinkIntegrityErrors);
printf("linkDowned:           %u\n",
       port_counters.linkDowned);

fail2:
    // close our session
    omgt_close_port(port);
fail1:
    return exitcode;
}
```