



Cornelis™ Omni-Path Express™ Architecture Management API

Programmer's Guide

November 2022
Doc. No. J68876, Rev. 10.0

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Cornelis Networks products described herein. You agree to grant Cornelis Networks 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 product plans and roadmaps are subject to change without notice.

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.

Cornelis Networks technologies may require enabled hardware, software, or service activation.

No product or component can be absolutely secure.

Your costs and results may vary.

Cornelis, Cornelis Networks, Omni-Path, Omni-Path Express, and the Cornelis Networks logo belong to Cornelis Networks, Inc. Other names and brands may be claimed as the property of others.

Copyright © 2022 Cornelis Networks, Inc. All rights reserved.

Table of Contents

1. Preface	8
1.1. Intended Audience	8
1.2. Documentation Library	8
1.2.1. How to Search the Cornelis Omni-Path Express Documentation Set	11
1.3. Document Conventions	11
1.4. Best Practices	12
1.5. Cornelis Omni-Path Express Fabric Design Generator for Cornelis Omni-Path Express Fabric	12
1.6. License Agreements	13
1.7. Technical Support	13
2. Introduction	14
3. OPX Management API	15
3.1. Compatibility	15
3.2. Dependency	15
3.3. Client Communication Model	15
3.3.1. In-Band Communication	15
3.3.2. Out-of-Band Communication	16
3.3.3. Client Timeouts and Retries	17
3.4. Logging	17
3.5. Asynchronous Trap Notifications	17
3.6. Optimized Accessing of Historical PA Image Data	17
3.6.1. Querying Using the STL_PA_IMAGE_ID_DATA Structure	17
3.6.2. Querying Multiple Attributes from a Single Image	18
3.6.3. Querying Multiple Attributes from Multiple Images	20
4. Functional Documentation	21
4.1. Initialization and Maintenance	21
4.1.1. Data Structures	21
4.1.2. Defines	23
4.1.3. Initialization and Maintenance Functions	25
4.1.4. General Port Accessor Functions	29
4.1.5. In-Band Port Accessor Functions	31
4.1.6. Out-of-Band Port Accessor Functions	36
4.1.7. General Functions	39
4.2. SA Interface	40
4.2.1. Data Structures	40
4.2.2. Functions	42
4.3. Async Notification SA Interface	79
4.3.1. Functions	79
4.4. PA Interface	81
4.4.1. Defines	81
4.4.2. Functions	82
5. Sample Programs	103

5.1. Prerequisites	103
5.2. Building the Programs	103
5.3. saquery	104
5.4. paquery	105
6. Glossary of Acronyms	108

Revision History

Date	Rev	Description
Nov 2022	10.0	Updated to Cornelis branding and naming conventions.
Aug 2020	9.0	Updated "Best Practices" to include pointer to "Updating the Certificate" in the <i>Cornelis Omni-Path Express Fabric Switches GUI User Guide</i>.
Jan 2020	8.0	Updated links to OPX documentation set in "Omni-Path Documentation Library".
Oct 2019	7.0	- Updated Preface to include new "Best Practices". - Added Performance information to all PA "MGT Functions". - Added new PA functions: - omgt_pa_get_group_list2 - omgt_pa_get_vf_list2 - omgt_pa_release_group_list2 - omgt_pa_release_vf_list2
Jun 2019	6.0	Added new section: "Optimized Accessing of Historical PA Image Data"
Apr 2019	5.0	Updated "Building the Programs" to correct the CFLAGS line in the sample code.
Sep 2018	4.0	- Updated "SA Interface" functions to change "int * num_records" to "int32_t * num_records". - Added new functions: - omgt_sa_get_devicegroupname_records - omgt_sa_get_devicegroupmember_records - omgt_sa_get_devicetreemember_records - omgt_sa_get_scv1r_table_records - Added new QUERY_INPUT_TYPE in omgt_sa_selector_t.

Date	Rev	Description
Apr 2018	3.0	• Added a new "Preface" section. • Updated the "Introduction" section. • Updated the "Compatibility" section. • Updated the "Client Communication Model" section. • Updated the "Logging" section. • Updated the "Asynchronous Trap Notifications" section. • Updated <code>omgt_port_get_ipv4_addr</code>; updated the description for <code>OMGT_STATUS_INVALID_STATE</code>. • Updated <code>omgt_port_get_ipv6_addr</code>; updated the description for <code>OMGT_STATUS_INVALID_STATE</code>. • Updated <code>omgt_sa_get_notice_report</code>; updated the description for the port and context parameters; deleted <code>OMGT_STATUS_INVALID_STATE</code> from the Returns section. • Added a note to <code>omgt_pa_get_port_stats</code> to indicate that this function has been deprecated. • Added a note to <code>omgt_pa_get_vf_port_stats</code> to indicate that this function has been deprecated. • Added these "PA Interface" functions: – <code>omgt_pa_get_group_nodeinfo</code> – <code>omgt_pa_get_group_linkinfo</code> – <code>omgt_pa_release_group_nodeinfo</code> – <code>omgt_pa_release_group_linkinfo</code> • Updated code example in <code>saquery</code>.

Date	Rev	Description
Oct 2017	2.0	• Added cost matrix query, asynchronous notifications, and configurable timeouts. • Added the following new features: – SwitchCost Record query: An SA query that returns data an OPXS Fabric Manager uses when making routing decisions in a fabric. See <code>omgt_sa_get_switchinfo_records</code>. – 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". – Methods to check/refresh states of SA/PA: ▪ <code>omgt_port_get_pa_service_state</code> ▪ <code>omgt_port_get_sa_service_state</code> • Updated saquery sample to set errorcode to 2 if unsuccessful in query. • Added new code samples to demonstrate the new Switch Cost Record and Asynchronous Trap additions. See "Sample Programs". • Added missing <code>omgt_sa_get_scvInt_table_records</code> declaration. Version 0.1 of OPAMGT omits this declaration, which yields a warning with this function.
Aug 2017	1.0	Initial release of document.

1. Preface

This guide is part of the documentation set for the Cornelis Omni-Path Express (OPX) Fabric, which is an end-to-end solution consisting of Cornelis Omni-Path Express Host Fabric Interface Adapters (HFIs), OPX Switches, and fabric management and development tools.

The Cornelis Omni-Path Express 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 OPX Fabric and standard InfiniBand (IB) can send Internet Protocol (IP) traffic over the fabric, or *IPoFabric*. In this document 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.

1.1. Intended Audience

The intended audience for the OPX document set is network administrators and other qualified personnel.

1.2. Documentation Library

Go to the [Cornelis Customer Center](#) to download the publications from the Release Library. Use the tasks listed in this table to find the corresponding document.

Task	Document Title	Description
Using the OPX documentation set	<i>Cornelis Omni-Path Express Fabric Quick Start Guide</i>	A roadmap to Cornelis' comprehensive library of publications describing all aspects of the product family. This document outlines the basic steps for installing your OPX cluster and ensuring it is operational.
Setting up an OPX cluster	<i>Cornelis Omni-Path Express Fabric Setup Guide</i>	Provides a high-level overview of the steps required to stage a customer-based installation of the OPX Fabric. Procedures and key reference documents, such as OPX user and installation guides, are provided to clarify the process. Additional commands and best known methods are defined to facilitate the installation process and troubleshooting.
Installing hardware	<i>Cornelis Omni-Path Express Fabric Switches Hardware Installation Guide</i>	Describes the hardware installation and initial configuration tasks for OPX Director Class Switches and OPX Edge Switches.
	<i>Cornelis Omni-Path Express Host Fabric Interface Installation Guide</i>	Contains instructions for installing the OPX HFI in an OPX cluster.

Task	Document Title	Description
	<i>Cornelis Omni-Path Express Gateway Installation and Setup Guide</i>	Describes the hardware installation and set up tasks for the OPX Gateways.
Installing Host Software Installing HFI firmware Installing switch firmware (externally-managed switches)	<i>Cornelis Omni-Path Express 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>Cornelis Omni-Path Express Fabric Switches GUI User Guide</i>	Describes the graphical user interface (GUI) of the OPX Chassis Viewer GUI. This document provides task-oriented procedures for configuring and managing the OPX Switch family. Help: GUI embedded help files
Managing a switch using the CLI Installing switch firmware (managed switches)	<i>Cornelis Omni-Path Express Fabric Switches Command Line Interface Reference Guide</i>	Describes the CLI task information for the OPX Switch family. Help: -help for each CLI
Managing a fabric using FastFabric	<i>Cornelis Omni-Path Express 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 TUI menus and 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>Cornelis Omni-Path Express Fabric Suite Fabric Manager User Guide</i>	The Fabric Manager uses a well-defined management protocol to communicate with management agents in every OPX HFI and switch. Through these interfaces the Fabric Manager can discover, configure, and monitor the fabric.
	<i>Cornelis Omni-Path Express 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 HFI and IPoIB driver Running MPI applications on OPX	<i>Cornelis Omni-Path Express Fabric Host Software User Guide</i>	Describes how to set up and administer the OPX HFI after the software has been installed. This document is for cluster administrators and Message-Passing Interface (MPI) application programmers.

Task	Document Title	Description
Writing and running middleware that uses OPX software	<i>Cornelis Performance Scaled Messaging 2 (PSM2) Programmer's Guide</i>	Provides a reference for programmers working with the PSM2 Application Programming Interface (API). The Performance Scaled Messaging 2 API (PSM2 API) is a low-level user-level communications interface.
	<i>Cornelis Omni-Path Express OPX_Provider Installation and Setup Application Note</i>	Provides a reference for programmers working with the OPX provider.
Optimizing system performance	<i>Cornelis Omni-Path Express 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 the OPX 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 OPX	<i>Cornelis Omni-Path Express 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 an OPX Fabric and a legacy InfiniBand fabric.
Building Containers for OPX Fabrics	<i>Building Containers for Cornelis Omni-Path Express 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 OPX networking technology.
Writing management applications that interface with OPX	<i>Cornelis Omni-Path Express Management API Programmer's Guide</i>	Contains a reference for programmers working with the OPX Architecture Management (OPAMGT) Application Programming Interface (API). The OPAMGT API is a C-API permitting in-band and out-of-band queries of the FM's Subnet Administrator and Performance Administrator.
Using NVM over Fabrics on OPX	<i>Configuring Non-Volatile Memory Express (NVMe) over Fabrics on Cornelis Omni-Path Express Application Note</i>	Describes how to implement a simple OPX 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>Cornelis Omni-Path Express Fabric Software Release Notes</i>	
	<i>Cornelis Omni-Path Express Fabric Manager GUI Software Release Notes</i>	
	<i>Cornelis Omni-Path Express Fabric Switches Release Notes</i> (includes managed and externally-managed switches)	
	<i>Cornelis Omni-Path Express Fabric Unified Extensible Firmware Interface (UEFI) Release Notes</i>	
	<i>Cornelis Omni-Path Express Fabric Thermal Management Microchip (TMM) Release Notes</i>	
	<i>Cornelis Omni-Path Express Fabric Firmware Tools Release Notes</i>	

1.2.1. How to Search the Cornelis Omni-Path Express 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 publications into a single folder.
2. Open Acrobat 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.

1.3. Document Conventions

The following conventions are standard for Cornelis Omni-Path Express 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 [Section 1.6 "License Agreements"](#) for more information.

For more information, visit www.cornelisnetworks.com.

- Text in **bold** font indicates user interface elements such as menu items, buttons, check boxes, key names, keystrokes, 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 *Cornelis Omni-Path Express 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 qualifiers:

- **(Linux)** – Tasks are only applicable when Linux is being used.
- **(Host)** – Tasks are only applicable when OPX Host Software or OPXS is being used on the hosts.
- **(Switch)** – Tasks are applicable only when OPX Switches or Chassis are being used.
- Tasks that are generally applicable to all environments are not marked.

1.4. Best Practices

Note the following Cornelis recommendations:

- Update to the latest versions of OPX firmware and software to obtain the most recent functional and security updates.
- To improve security:
 - administrators should log out users and disable multi-user logins prior to performing provisioning and similar tasks.
 - update the default HTTPS certificate (refer to the *Cornelis Omni-Path Express Fabric Switches GUI User Guide*, “Updating the Certificate” for details).
 - using 48-bit CRC when the peak bandwidth is not critical in the fabric. 10.8.2.0 firmware introduced a feature to improve the detection of failing cables while also providing avoidance of possible downstream issues.

Due to the improved detection of failing cables provided by 10.8.2.0 firmware, it is possible that failing cables that were not previously detected by earlier firmware will now be detected by the 10.8.2.0 firmware. It is expected that the 48-bit CRC and the 10.8.2.0 firmware detection methods will have similar detection robustness. When a failing cable is identified, it is important to replace the cable at the earliest possible convenience.
- Ensure all fabrics contain at least one managed switch to troubleshoot potential issues with externally-managed edge switches.

1.5. Cornelis Omni-Path Express Fabric Design Generator for Cornelis Omni-Path Express Fabric

The Fabric Design Generator 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.

To access the Cluster Configurator Fabric Design Generator for OPX Fabric, go to [Go to Cornelis™ Omni-Path Express™ Fabric Design Generator.](#)

1.6. 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.

1.7. Technical Support

Technical support for Cornelis Omni-Path Express products is available 24 hours a day, 365 days a year. Please contact [Cornelis Networks Customer Support](#) by visiting www.cornelisnetworks.com or send your request directly to support@cornelisnetworks.com.

2. Introduction

This manual is a reference for programmers working with the Cornelis Omni-Path Express Management (OPAMGT) Application Programming Interface (API). The 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 Cornelis Omni-Path Express Fabric Suite (OPXS) including the Fabric Manager (SA/PA) and FastFabric. See the *Cornelis Omni-Path Express Fabric Suite Fabric Manager User Guide* and the *Cornelis Omni-Path Express Fabric Suite FastFabric User Guide* for a complete description.

For other documents in the OPX product line, refer to [Section 1.2 "Documentation Library"](#).

3. OPX Management API

3.1. Compatibility

OPX Management (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.

OPAMGT operates with the 10.x series of OPX 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.

3.2. Dependency

The following libraries are required by OPAMGT:

- libibumad
- libssl
- libcrypto

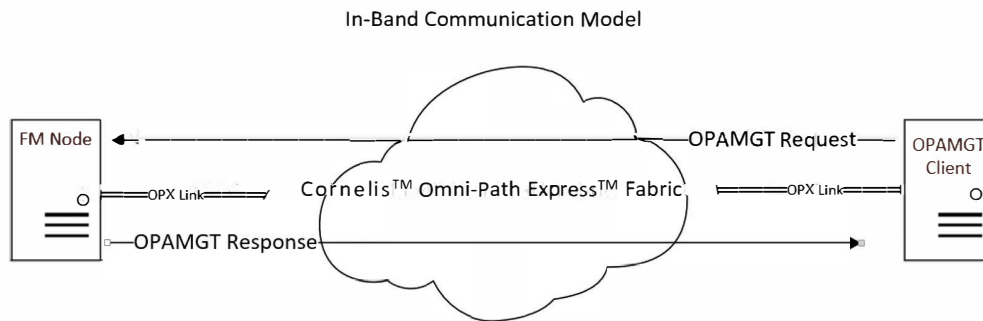
3.3. Client Communication Model

The 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 [Section 4.1.1 "Data Structures"](#) for all the connection initialization functions.

3.3.1. In-Band Communication

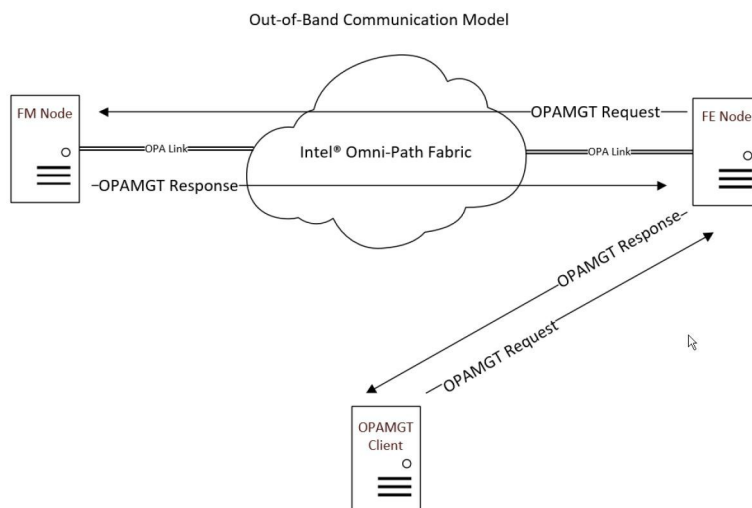
In-Band communication is defined as when an OPAMGT client attached to the OPX Fabric can issue requests through its HFI/port directly to the SA or PA by routing through the Fabric.

Figure 1. In-Band Communication Model


3.3.2. Out-of-Band Communication

Out-of-Band communication is defined as when an OPAMGT client issues a request using TCP/IP protocols to the 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 OPAMGT user.

The 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 *Cornelis Omni-Path Express 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


3.3.3. 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 [Section 4.1 "Initialization and Maintenance"](#) for defines and functions to change values.

3.4. Logging

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 [Section 4.1.1.2 "omgt_params"](#), [Section 4.1.4.1 "omgt_set_dbg"](#), and [Section 4.1.4.2 "omgt_set_err"](#). Logging occurs on a per-session basis; debug information for different sessions can be sent to different locations.

3.5. Asynchronous Trap Notifications

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 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 [Section 4.3 "Async Notification SA Interface"](#) for the list of trap functions. See the *Cornelis Omni-Path Express Fabric Suite Fabric Manager User Guide* for more information on Fabric Sweeping and Fabric Change Detection.

3.6. Optimized Accessing of Historical PA Image Data

OPAMGT provides several methods for accessing historical performance data through the PA. The following sections provide recommendations to follow when accessing Image Data. See [Section 4.2.2 "Functions"](#) for the function definitions referenced in this section.

3.6.1. 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. [Section 4.4.1 "Defines"](#) contains the defines to switch between these methods.

3.6.1.1. 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.

In 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.

3.6.1.2. 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.

3.6.2. 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 the 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 the 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);
```

3.6.3. 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 freeze a new image, the atomic move frozen image operation performs a release on the old image and 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);
```

4. 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 SA.
These header files expose the structure definitions for the data types used in the SA and PA Query functions.	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.

4.1. Initialization and Maintenance

4.1.1. Data Structures

4.1.1.1. omgt_port

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

Syntax

```
struct omgt_port
```

4.1.1.2. 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.

4.1.1.3. 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
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

4.1.1.4. omgt_oob_input

Out-of-band input configuration options to set up 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

4.1.2. Defines

Define	Description
OMGT_DBG_FILE_SYSLOG	Reserved FILE pointer to indicate to the logger to print to the syslog.

Define	Description
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.

4.1.2.1. 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.

4.1.2.2. 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

4.1.2.3. 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

4.1.3. Initialization and Maintenance Functions

4.1.3.1. 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 OPX-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.

4.1.3.2. 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 OPX-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.

4.1.3.3. 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 OPX-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.

4.1.3.4. 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.

4.1.3.5. omgt_close_port

Closes and frees a 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.

4.1.4. General Port Accessor Functions

4.1.4.1. 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.

4.1.4.2. 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.

4.1.4.3. 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.

4.1.4.4. 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.

4.1.5. In-Band Port Accessor Functions

4.1.5.1. 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_STAT E Port mode error, port is in out-of-band mode.
OMGT_STATUS_PROTECTION Port failed to access cache data.

4.1.5.2. 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_STAT E Port mode error, port is in out-of-band mode.
OMGT_STATUS_PROTECTION Port failed to access cache data.

4.1.5.3. 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_STAT E	Port mode error, port is in out-of-band mode.
OMGT_STATUS_PROTECTION	Port failed to access cache data.

4.1.5.4. 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_STAT E	Port mode error, port is in out-of-band mode.
OMGT_STATUS_PROTECTION	Port failed to access cache data.

4.1.5.5. 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

Parameter	Description
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.

4.1.5.6. 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: • IB_PORT_DOWN: The port has not achieved LinkUp in the physical layer. • IB_PORT_INIT: The port is in LinkUp and ready to be programmed by the FM. It cannot yet pass non-management traffic. • IB_PORT_ARMED: The port is programmed and ready to become active. • IB_PORT_ACTIVE: The port is fully active and capable of passing all traffic.

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.

4.1.5.7. 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.
 OMT_STATUS_INVALID_STAT Port mode error, port is in out-of-band mode.
 E
 OMT_STATUS_PROTECTION Port failed to access cache data.

4.1.5.8. 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.

METER

4.1.5.9. 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.

METER

4.1.6. Out-of-Band Port Accessor Functions

4.1.6.1. 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.

4.1.6.2. 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.

4.1.6.3. 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.

4.1.6.4. 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.

4.1.7. General Functions

These are functions where one does not use an `omgt_port` object.

4.1.7.1. `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
<code>hfis</code>	Pointer to array of size <code>char [max][UMAD_CA_NAME_LEN]</code>
<code>max</code>	Maximum number of names to return
<code>hfi_count</code>	The number of valid entries in <code>hfis</code>

Returns

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

4.1.7.2. `omgt_service_state_totext`

Converts the service state to text.

Syntax

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

Parameters

Parameter	Description
<code>service_state</code>	value to convert to text

Returns

Service state in text.

4.1.7.3. 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.

4.2. SA Interface

4.2.1. Data Structures

4.2.1.1. 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

4.2.2. Functions

4.2.2.1. 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.

4.2.2.2. 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.

4.2.2.3. 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

4.2.2.4. 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

Parameter	Description
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

4.2.2.5. 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

Parameter	Description
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

4.2.2.6. 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 <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

4.2.2.7. 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

4.2.2.8. 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

4.2.2.9. 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 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

4.2.2.10. 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

4.2.2.11. 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 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

4.2.2.12. 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

4.2.2.13. 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 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

4.2.2.14. 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

Parameter	Description
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

4.2.2.15. 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

Parameter	Description
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

4.2.2.16. 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

4.2.2.17. 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 ** 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 an array of 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

4.2.2.18. 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

4.2.2.19. 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 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

4.2.2.20. 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 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

4.2.2.21. 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 an array of 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

4.2.2.22. 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

4.2.2.23. 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 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

4.2.2.24. 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

Parameter	Description
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

4.2.2.25. 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

Parameter	Description
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

4.2.2.26. 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

4.2.2.27. 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 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

4.2.2.28. 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

4.2.2.29. 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 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

4.2.2.30. 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

4.2.2.31. 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 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

4.2.2.32. 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

4.2.2.33. omgt_sa_get_switchcost_records

Query SA for SwitchCost records.

Syntax

The OPX 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 [Section 4.3 "Async Notification SA Interface"](#) for registering for notifications when this data changes.

```
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 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

4.2.2.34. 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 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

4.2.2.35. 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 ** guids)
```

Parameters

Parameter	Description
port	Port object previously opened and initialized

Parameter	Description
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

4.2.2.36. 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

Parameter	Description
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

4.2.2.37. 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 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

4.2.2.38. 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

4.2.2.39. 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

4.2.2.40. 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

4.2.2.41. 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

4.2.2.42. 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

4.2.2.43. 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

4.3. Async Notification SA Interface

4.3.1. Functions

4.3.1.1. 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.

4.3.1.2. 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.

4.3.1.3. omgt_sa_get_notice_report

Gets the notice report forwarded from the SM.

4.3.1.3.1. 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)
```

4.3.1.3.2. 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).

4.3.1.3.3. 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.

4.4. PA Interface

4.4.1. Defines

Define	Description
PAClient_IMAGE_CURRENT	Used to request PA data from the most recent fabric sweep

Define	Description
PACIENT_IMAGE_TIMED	Used to request PA data from a sweep at a given time

4.4.2. Functions

4.4.2.1. 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)

4.4.2.2. 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)

4.4.2.3. 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)

4.4.2.4. 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
 OMT_STATUS_ERROR Error

Performance

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

4.4.2.5. 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

4.4.2.6. 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

Parameter	Description
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
 OMT_STATUS_ERROR Error

Performance

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

4.4.2.7. 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
 OMT_STATUS_ERROR Error

Performance

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



NOTE

See also [Section 4.4.2.8 "omgt_pa_get_group_list2"](#)

4.4.2.8. 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

4.4.2.9. 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

Parameter	Description
pm_image_info	Pointer to image info to fill

Returns

OMGT_STATUS_SUCCESS Get successful
 OMT_STATUS_ERROR Error

Performance

O(1)

4.4.2.10. 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
 OMT_STATUS_ERROR Error

Performance

O(1)

4.4.2.11. omgt_pa_get_port_stats



NOTE

Deprecated, use [Section 4.4.2.12 "omgt_pa_get_port_stats2"](#).

4.4.2.12. 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
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
 OMTG_STATUS_ERROR Error

Performance

O(1)

4.4.2.13. omgt_pa_get_vf_config

Get virtual fabric (VF) 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

4.4.2.14. omgt_pa_get_vf_focus

Get virtual fabric (VF) 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

Parameter	Description
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

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

4.4.2.15. omgt_pa_get_vf_info

Get virtual fabric (VF) 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

4.4.2.16. 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 [Section 4.4.2.17 "omgt_pa_get_vf_list2"](#)

4.4.2.17. 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

4.4.2.18. omgt_pa_get_vf_port_stats



NOTE

Deprecated, use [Section 4.4.2.19 "omgt_pa_get_vf_port_stats2"](#).

4.4.2.19. 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
vf_name	Pointer to VF name

Parameter	Description
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
 OMTG_STATUS_ERROR Error

Performance

O(1)

4.4.2.20. 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
 OMTG_STATUS_UNAVAILABLE Image 2 unavailable to freeze
 OMTG_STATUS_ERROR Error

Performance

$O(1)$

4.4.2.21. 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 nodeId, 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
 OMTG_STATUS_ERROR Error

Performance

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

4.4.2.22. 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
 OMTG_STATUS_ERROR Error

Performance

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

4.4.2.23. 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)

4.4.2.24. 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)

4.4.2.25. 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)

4.4.2.26. 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)

4.4.2.27. 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)

4.4.2.28. omgt_pa_release_vf_config

Releases virtual fabric (VF) 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)

4.4.2.29. 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)

4.4.2.30. 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)

4.4.2.31. 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)

4.4.2.32. 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)

4.4.2.33. 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)

4.4.2.34. 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)

5. Sample Programs

This section describes two sample programs that can be used to understand basic 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 ([Section 5.3 "saquery"](#)) - A "hello world" of how to query an OPX SA
- saquery - A sample showing all requests for subnet management information stored in an OPX SA
- simple_pa_query ([Section 5.4 "paquery"](#)) - A "hello world" of how to query an OPX PA
- paquery - A sample showing all requests for performance management information stored in an OPX 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 OPX 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 OPX Fabric must be available and the 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)
{
    OMT_STATUS_T status = OMT_STATUS_SUCCESS;
    int exitcode = 0;
    int i;
    struct omt_port * port = NULL;
    int num_records;
    STL_PORTINFO_RECORD *pi_records = NULL;

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

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

    // execute query synchronously
    status = omt_sa_get_portinfo_records(port, &selector,
                                         &num_records,
                                         &pi_records);
    if (status != OMT_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;
    OMG_T_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;
}
```

6. Glossary of Acronyms

The following acronyms are used throughout the OPX Documentation Set.

ABI	Application Binary Interface - an interface between two binary program modules; often, a library or operating system, and a program being run by a user
API	Application Programming Interface - how an application interacts with another application or a user
CA	Certificate Authority - a trusted third party that stores, signs, and issues digital certificates certifying the ownership of a public key
CRL	Certificate Revocation List - a list of revoked digital certificates by an - issuing certificate authority (CA) before their scheduled expiration and should no longer be trusted
ESM	Embedded Subnet Manager
FE	Fabric Executive
FM	Fabric Manager
GUID	Globally Unique Identifier - a 128-bit text string generated when a unique reference number is needed to identify a component (e.g., hardware, software, account, node, etc.) on a computer or network
HFI	Host Fabric Interface - the hardware and software that allows one module to talk to another in an OPX Fabric.
IB	InfiniBand - a computer networking communications standard used in high-performance computing that features very high throughput and very low latency
IP	Internet Protocol - a set of rules for communicating over the internet
LID	Local Identifier
MAD	Management Datagram
OPAMGT	Omni-Path Express Management
OPX	Omni-Path Express

OPX provider	OPX provider is a new enhanced libfabric provider that takes full advantage of the libfabric acceleration features while running over existing and future Omni-Path Express hardware.
PA	Performance Administrator/Administration
PEM	Privacy Enhanced Mail - a file type (.pem)
PKey	Partition Key
PM	Performance Manager, Performance Management, Program Manager, Program Management
SA	Subnet Administrator/Administration
SC	Service Channel
SCSC	SCSC
SCSL	SCSL
SCVLT	SCVLT
SL	Service Level
SLSC	SLSC
SM	Subnet Manager or Subnet Management
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
VF	Virtual Fabric
VL	Virtual Lane