OpenSM for Windows User's Manual

Rev 1.40

OpenSM for Windows User's Manual

Document Number: 2352

Mellanox Technologies, Inc.
2900 Stender Way
Santa Clara, CA 95054
U.S.A.
www.Mellanox.com

Tel: (408) 970-3400
Fax: (408) 970-3403

Mellanox Technologies Ltd
PO Box 586 Hermon Building
Yokneam 20692
Israel

Tel: +972-4-909-7200
Fax: +972-4-959-3245

Contents

About this Manual

This manual describes the OpenSM for Windows package. OpenSM is a Subnet Manager for the initialization of InfiniBand compliant devices.

This manual is organized in the following manner:

- Chapter 1 provides an overview of the OpenSM package(page 7)
- Chapter 2 provides instructions for the installation of the OpenSM package (page 9)
- Chapter 3 describes how to use the tools included this OpenSM package (page 10)
- Chapter 4 describes the UPDN algorithm, its purpose and usage (page 31)

Intended Audience

The target audience of this User's Manual is System Administrators who have installed InfiniBand hardware and need to run a Subnet Manager (SM) in order to initialize it.

Related Documentation

For InfiniBand related issues, please refer to the following specification:

- InifiniBand Architecture Specification Volume 1, Release 1.2

For implementing user-level applications that interface with the InfiniBand Subnet Administrator, send MADs through the SMI or GSI, please refer to the following manual:

- OpenSM Vendor Layer API and Programmer's Manual, Rev. 0.1

1 Overview

The *InifiniBand Architecture Specification (Volume 1)* defines extensively and explicitly how InfiniBand (IB) compliant devices are to be managed. There it is stated that routing and other management policies, for each compliant device, should be exported to a centralized entity called: Subnet Manager (SM). This standardization of management significantly reduces the costs of hardware when compared to the costs of traditional distributed network management policies.

OpenSM is an InfiniBand compliant Subnet Manager. It is provided as a fixed flow executable called ***opensm***. It is accompanied by a testing application called ***osmtest***.

The target audience of this User's Manual is System Administrators who have installed InfiniBand hardware and need to run a Subnet Manager in order to initialize it. Since OpenSM implements an SM as defined in the IB specification, it will not be defined here again. Instead, to understand what OpenSM implements, the user is kindly referred to the following chapters of that specification: Management Model (13), Subnet Management (14), and Subnet Administration (15).

1.1 Key Concepts and Terms

Throughout this manual, there is frequent reference to various concepts and terms which are common to the general audience of System Administrators. There is also reference to InfiniBand-specific concepts and terms, a part of which are briefly defined in the list to follow. Other terms may be defined where relevant in this document; however, the IB specification remains the main reference for all (missing and existing) IB terms.

- **IB devices:**
  Integrated Circuits implementing InfiniBand compliant communication.
- **IB Fabric/Cluster/Subnet:**
  A set of IB devices connected by IB cables.
- **Subnet Manager (SM):**
  One of several entities involved in the configuration and control of the subnet.
- **Master Subnet Manager:**
  The subnet manager that is authoritative, that has the reference configuration information for the subnet.
- **Standby Subnet Manager:**
  A subnet manager that is currently quiescent, and not in the role of a master SM, by agency of the master SM.
- **Subnet Administrator (SA):**
  An application (normally part of the Subnet Manager) that implements the interface for querying and manipulating subnet management data.
- **LID**:
  An address assigned to a port (data sink or source point) by the Subnet Manager, unique within the subnet, used for directing packets within the subnet.
- **Unicast Linear Forwarding Tables (LFT):**
  A table that exists in every switch providing the port through which packets should be sent to each LID.
- **Multicast Forwarding Tables:**
  A table that exists in every switch providing the list of ports to forward received multicast packet. The table is organized by MLID.

1.2 Contents of OpenSM Package

The OpenSM package contains the following executables and libraries:

- ***opensm*:**
  A Subnet Manager and Administrator. It should be run as a service on one or two machines of the IB cluster.
- ***osmtest*:**
  A simple application to test ***opensm***. It is capable of exercising most of the SA queries and provides clear feedback on their success.

1.3 Dependencies

OpenSM in its current form is available on top of Mellanox Technologies' WinIB stack.

2 Installation of OpenSM Package

OpenSM is bundled into Mellanox Technologies' WinIB software stack (and various other distributions). For WinIB installation instructions, please refer to http://www.mellanox.com. To check whether an OpenSM version is already installed on your machine, check for the existence of the two bin files *opensm.exe* and *osmtest.exe* under \Program files\Mellanox\WinIB\Tools.

Note that OpenSM is installed both as a command line executable and as a Windows service. As a Windows service, OpenSM can be automatically started upon boot.

3 Using *opensm* and *osmtest*

This section of the manual describes the provided executables and their usage. It is recommended to run OpenSM as a service and make it start automatically on the machine to run *opensm*. OpenSM can also be run as a standalone executable from the command window once the service is stopped. The following sections describes the usage of *opensm and osmtest.*

3.1 opensm

There are two ways to start OpenSM: manually and automatically.

3.1.1 Manual Operation

opensm is a simple command line executable that serves as both a Subnet Manager and a Subnet Administrator. It can be run with or without specifying any command line options.

3.1.1.1 Default or Common Case Usage

By entering ***opensm*** on the command line, without any additional options, the default settings will be chosen. These defaults were designed to meet the common case usage on clusters with up to a few hundred nodes. Thus, in this default mode, ***opensm*** will scan the IB fabric, initialize it, and sweep occasionally for changes.

OpenSM attaches to a specific IB port on the local machine and configures only the fabric connected to it. If the local machine has more than one IB port, OpenSM binds to the fist port which is not in the DOWN state -- unless a specific port GUID is specified either on the command line or by the cache options file (see ‘opensm.opts’ below). If all ports are in the DOWN state and no port GUID is specified on the command line or the in the option file, OpenSM binds to Port 1.

The run is logged into the osm.log file under the user temp directory (%temp%\osm.log) and in the Application section of the Windows event viewer. In case of a fatal and non-recoverable error occurs, ***opensm*** aborts.

Both the log file and the event viewer should include the message “SUBNET UP” if ***opensm*** was able to setup the subnet correctly.

3.1.1.2 Non-default Usage

It is possible for the user to run ***opensm*** with settings other than the default ones. Table 1 lists the ***opensm*** command line options in the first column, the effect of each option in the second, and tips on when to use each option in the last.

Table 1 - ***opensm*** Command Line Options.

Option	Effect	When to Use Option
-g <GUID in hex> --guid <GUID in hex>	OpenSM will bind to the port with the provided GUID. Default is to present to user the available GUIDs, and to select one of them.	To avoid the interaction required in order to select the port. It is possible to enter: “echo 1 | opensm” to select the first GUID.
-s <interval> --sweep <interval>	This option specifies the number of seconds between subnet sweeps. Specifying -s 0 disables sweeping. Default: OpenSM sweeps with intervals of 10 seconds.	To minimize unneeded sweeps, it is possible to set this value to 0. This will cause a single sweep and only traps will cause new sweeps. On large clusters, it is recommended to set this to a value higher than 60.

Table 1 - ***opensm*** Command Line Options.

Option	Effect	When to Use Option
-t <milliseconds> --timeout <milliseconds>	This option specifies the time in milliseconds used for transaction timeouts (request to response). Default: Timeout value is 100ms.	This value should be changed only on large subnets. A reasonable value for a >1000nodes cluster is ~1000ms.
-p <PRIORITY> --priority <PRIORITY>	This option specifies the SM's PRIORITY. This will affect the handover cases, where master is chosen by priority and GUID.	Only if there is a need to explicitly control which SM should be the master.
-v --verbose	This option increases the log verbosity level. The -v option may be specified multiple times to further increase the verbosity level.	The first -v will print to the stdout a summary table of the discovered fabric.
-V	This option sets the verbosity to its maximum level and forces log flushing.	Use this option to investigate an error or send a bug report
-f <file_name> --log_file <file_name>	This option defines the log file. By default the log goes to %TEMP%\osm.log. To send it to standard output use "-f stdout".	For your convenience only. Note that if you use -V the log file might be too large for the %TEMP%...
-e --erase_log_file	This option causes deletion of old log file, and the current log file will start from scratch. By default the log file is accumulative.	For your convenience only. Note that since the log is accumulative by default, it might grow to be too large even without running in verbose mode.
-o --once	This option causes OpenSM to configure the subnet once, then exit. Ports remain in the ACTIVE state.	For testing purposes only.
-r --reassign_lids	This option causes OpenSM to reassign LIDs to all endnodes. Specifying -r on a running subnet may disrupt subnet traffic. Default: OpenSM attempts to preserve existing LID assignments resolving multiple use of same LID.	Should rarely be used. Use only if re-numbering of all the LIDs is required. Note that when using multiple SMs (for redundancy), this option should NEVER be used.
-c --cache-options	Create a cache file with the full set of options used by the SM. This includes command line and internal defaults.	If you need to change the internal options for OpenSM (that are not exposed through the command line), you need to create the options cache file (named opensm.opts under the OpenSM cache directory: \Program files\Mellanox\WinIB\etc). You can later modify the file and restart OpenSM for your changes to take effect.
-l <LMC> --lmc <LMC>	This option specifies the subnet LMC value. The number of LIDs assigned to each port is 2^LMC. The LMC value must be in the range 0-7. LMC values > 0 allow multiple paths between ports. Default: OpenSM defaults to LMC = 0, which allows one path between any two ports.	Use when "path migration" option is required. Note that in order to take advantage of path migration, the Connection Manager (or any other method of RC communication setup) should take additional steps. LMC values > 0 should only be used if the subnet topology provides multiple paths between ports, i.e. multiple interconnects exist between switches.
-i <eq-ignore-guids-file> -ignore-guids <eq-ignore-guids-file>	This option provides means to define a set of ports (by guids) that will be ignored by the link load equalization algorithm.	If there are some ports on the fabric that are rarely used (like a dedicated OpenSM node), it is possible to specify their guids. This way, their BW will be ignored by the routing algorithm.

Table 1 - ***opensm*** Command Line Options.

Option	Effect	When to Use Option
-d <number> --debug <number>	This option specifies debug behavior. The number following -d selects the debug option (can be specified multiple times): -d 0 - Ignore other SM nodes. -d 1 - Force single threaded dispatching. -d 2 - Force log flushing after each log message. -d 3 - Disable multicast support. -d 4 - Put OpenSM in memory tracking mode. -d 10. Put OpenSM in testability mode. Default: No debug options are enabled.	These options are not normally needed.
-u	This option activates the UPDN routing algorithm instead of the (default) Min Hop algorithm	When there is a deadlock (e.g., due to high pressure) in a loop of the subnet. See “UPDN Unicast Routing Algorithm” on page 15 for more details.
-a <guid_list_file>	This option is active only when the UPDN algorithm is activate (option -u). It specifies the guid list file in which each guid is specified on a separate line	When the user wishes to manually specify the nodes of the subnet to be used as roots of the UPDN algorithm

3.1.2 Automatic Startup

By default, the OpenSM service (opensm) startup type is “Manual.” To start OpenSM automatically upon boot, set the startup type to “Automatic” in the service control panel.

In addition, the service can be started by typing the following command in the windows command prompt:

```
net start opensm
```

The service can be stopped by the command:

```
net stop opensm
```

When OpenSM is run as a service, the output is redirected to the Application section of Windows even (TBD: ??) viewer and the osm.log file as mentioned above.

To pass non-default parameters, use the cache options file as mentioned above.

3.2 OsmTest

OsmTest provides a test suite for OpenSM. Its executable is invoked by typing ***osmtest***.

OsmTest has the following capabilities and testing flows:

- It creates an inventory file of all available Nodes, Ports, and PathRecords, including all their fields.
- It verifies the existing inventory, with all the object fields, and matches it to a pre-saved one.
- A Multicast Compliancy test.
- An Event Forwarding test.
- A Service Record registration test.
- An RMPP stress test.
- A Small SA Queries stress test.

It is recommended that after installing OpenSM, the user should run "***osmtest*** -f c" to generate the inventory file, and immediately afterwards run "***osmtest*** -f a" to test OpenSM.

Another recommendation for OsmTest usage is to create the inventory when the IB fabric is stable, and occasionally run "osmtest -v" to verify that nothing has changed.

The following table provides a full description of all ***osmtest*** options.

Table 2 - OsmTest Command Line Options.

Option	Effect
-g <GUID in hex> --guid <GUID in hex>	OsmTest will bind to the port with the provided GUID. Default is to present to user the available GUIDs, and to select one of them. Use this option to avoid the interaction required in order to select the port. It is possible to enter: "echo 1 | opensm" to select the first GUID.
-i <filename> --inventory <filename>	This option specifies the name of the inventory file. Normally, ***osmtest*** expects to find an inventory file with which it validates real-time information received from the SA during testing. Default: ***osmtest*** uses the file 'osmtest.dat'.
-f <c|a|v|s|e|f|m|q> --flow <c|a|v|s|e|f|m|q>	The actual test flow run by OsmTest: c = create an inventory file with all nodes, ports and paths. a = run all validation tests (expects an input inventory) v = only validate the given inventory file. s = run service registration, un-registration and lease. e = run event forwarding test. f = flood the SA with queries according to the stress mode. m = multicast flow. q = QoS info - Dump VLArb and SLtoVL tables. Default flow: all the above but QoS
-M <mode>	This option specifies the mode of the Multicast flow: -M 1 - Short MC flow (default), single mode. -M 2 - Short MC flow, multiple flow. -M 3 - Long MC flow, single mode. -M 4 - Long MC flow, multiple mode
-s <level> --stress <level>	This option runs the specified stress test instead of the normal test suite. Stress test options are: -s 1 - Single-MAD response SA queries -s 2 - Multi-MAD (RMPP) response SA queries. Default: stress testing is not performed.

Table 2 - OsmTest Command Line Options.

Option	Effect
-t <milliseconds> --timeout <milliseconds>	This option specifies the time in milliseconds used for transaction timeouts (request to response). Default: 100ms.
-v --verbose	This option increases the log verbosity level. The -v option may be specified multiple times to further increase the verbosity level.
-V	This option sets the verbosity level to the maximum and forces log flushing.
-l <file_name> --log_file <file_name>	This option is used to specify the log file name. By default the log goes to standard output.
-d <number> --debug <number>	This option specifies debug behavior. The number following -d selects the debug option to be enabled (can be specified multiple times): -d 0 - Unused. -d 1 - Do not scan and compare Path Record (should be used on large clusters as #Paths = Nodes^2) -d 2 - Force log flushing after each log message. -d 3 - Unused. Default: no debug options are enabled.

4 UPDN Unicast Routing Algorithm

OpenSM offers two routing engines:

1. Min Hop Algorithm - based on the minimum hops to each node where the path length is optimized.
2. UPDN Unicast routing algorithm - also based on the minimum hops to each node, but it is constrained to ranking rules. This algorithm should be chosen if the subnet is not a pure Fat Tree, and a deadlock may occur due to a loop in the subnet.

The UPDN algorithm is installed as part of the OpenSM package and is placed under the lib directories. The algorithm is described in the following sections:

- "UPDN Algorithm Purpose" (page 15)
- "UPDN Algorithm Usage" (page 16)

4.1 UPDN Algorithm Purpose

The UPDN algorithm is designed to prevent deadlocks from occurring in loops of the subnet. A loop-deadlock is a situation in which it is no longer possible to send data between any two hosts connected through the loop. As such, the UPDN routing algorithm should be used if the subnet is not a pure Fat Tree, and one of its loops may experience a deadlock (due, for example, to high pressure).[1]

The UPDN algorithm is based on the following main stages:

1. Auto-detect root nodes - based on the HCA hop length from any switch in the subnet, a statistical histogram is built for each switch (hop num vs. number of occurrences). If the histogram reflects a specific column (higher than others) for a certain node, then it is marked as a root node. Since the algorithm is statistical, it may not find any root nodes. The list of the root nodes found by this auto-detect stage is used by the ranking process stage.

   Note 1: The user can override the node list manually (see Section 4.2).

   Note 2: If this stage cannot find any root nodes, and the user did not specify a GUID list file, OpenSM defaults back to the Min Hop routing algorithm.

2. Ranking process - All root switch nodes (found in stage 1) are assigned a rank of 0. Using the BFS algorithm, the rest of the switch nodes in the subnet are ranked incrementally. This ranking aids in the process of enforcing rules that ensure loop-free paths.
3. Min Hop Table setting - after ranking is done, a BFS algorithm is run from each (HCA or switch) node in the subnet. During the BFS process, the FDB table of each switch node traversed by BFS is updated, in reference to the starting node, based on the ranking rules and GUID values.

At the end of the process, the updated FDB tables ensure loop-free paths through the subnet.

1. To learn more about deadlock-free routing, see the article *"Deadlock Free Message Routing in Multiprocessor Interconnection Networks"* by William J Dally and Charles L Seitz (1985).

4.2 UPDN Algorithm Usage

4.2.1 Activation through *OpenSM*

Use '-u' for activating the UPDN algorithm

Use '-a <guid_list_file>' for adding an UPDN GUID file that contains the root nodes for ranking.

If the '-a' option is not used, OpenSM uses its auto-detect root nodes algorithm.

If no GUID list file is specified, OpenSM uses its auto-detect root nodes algorithm.

Notes on the GUID list file:

1. A valid GUID file specifies one GUID in each line. Lines with an invalid format will be discarded.
2. The user should specify the root switch guids. However, it is also possible to specify HCA guids; OpenSM will use the GUID of the switch (if it exists) that connects the HCA to the subnet as a root node.

Revision History

Table 3 - Revision History Table

Date	Revision	Description
September 2006	1.40	OpenSM is now installed as a Windows service
October 2005	1.30	First revision of WinIB OpenSM UM