

True Scale Fabric Suite FastFabric Command Line Interface

Reference Guide

July 2014



INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document 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.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

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

Any software source code reprinted in this document is furnished for informational purposes only and may only be used or copied and no license, express or implied, by estoppel or otherwise, to any of the reprinted source code is granted by this document.

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

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

Copyright © 2014, Intel Corporation. All rights reserved.



Contents

1.0	Introduction	19
1.1	Common Tool Options	19
1.1.1	-?	19
1.1.2	--help	19
1.1.3	-p	19
1.1.4	-s	20
1.1.5	-C	20
1.1.6	-I	20
1.2	Intended Audience	20
1.3	Related Materials	20
1.4	Documentation Conventions	20
1.5	Technical Support	21
1.6	License Agreements	21
2.0	Selection of Devices	23
2.1	Selection of Hosts	23
2.1.1	Host List Files	23
2.1.1.1	Host List File Format	23
2.1.2	Explicit host names	24
2.2	Selection of Chassis	24
2.2.1	Chassis List Files	25
2.2.1.1	Chassis List File Format	25
2.2.2	Explicit Chassis names	25
2.2.3	Selection of slots within a chassis	25
2.3	Selection of Switches	26
2.3.1	Switch List Files	27
2.3.1.1	Switch List File Format	27
2.3.2	Explicit Switch names	28
2.4	Selection of local Ports (subnets)	28
2.4.1	Port List Files	29
2.4.1.1	Port List File Format	29
2.4.2	Explicit ports	30
3.0	Basic Command Line Tools	31
3.1	Introduction	31
3.2	Basic Single Host Operations	31
3.2.1	clear_p1stats, clear_p2stats	31
3.2.1.1	Usage	32
3.2.2	iba_capture	32
3.2.2.1	Usage	32
3.2.2.2	Options	32
3.2.2.3	Examples	32
3.2.3	iba_showmc	32
3.2.3.1	Usage	33
3.2.3.2	Options	33
3.2.4	iba_hca_rev	33
3.2.4.1	Usage	33
3.2.4.2	Options	33
3.2.5	iba_mon	33
3.2.5.1	Usage	33
3.2.5.2	Options	33
3.2.6	iba_portconfig	34
3.2.6.1	Usage	34



3.2.6.2	Options	34
3.2.6.3	Example	35
3.2.7	iba_portdisable	35
3.2.7.1	Usage	35
3.2.7.2	Options	35
3.2.7.3	Example	36
3.2.8	iba_portenable	36
3.2.8.1	Usage	37
3.2.8.2	Options	37
3.2.8.3	Example	37
3.2.9	iba_portinfo	38
3.2.9.1	Usage	38
3.2.9.2	Options	38
3.2.9.3	Example	38
3.2.10	p1info, p2info	38
3.2.10.1	Usage	38
3.2.10.2	Options	38
3.2.11	p1stats, p2stats	39
3.2.11.1	Usage	39
3.2.12	iba_portstats	39
3.2.12.1	Usage	39
3.2.12.2	Options	39
3.2.12.3	Example	39
3.2.13	iba_portclear	39
3.2.13.1	Usage	39
3.2.13.2	Options	40
3.2.13.3	Example	40
3.2.14	iba_resolve_hca_port	40
3.2.14.1	Usage	40
3.2.14.2	Options	40
3.2.15	iba_sorthosts	40
3.2.15.1	Usage	40
3.2.15.2	Options	41
3.2.15.3	Example input	41
3.2.15.4	Resulting output	41
3.2.16	iba_verifyhosts	41
3.2.16.1	Usage	41
3.2.16.2	Options	42
3.2.16.3	Environment	42
3.2.16.4	Example	42
3.2.17	iba_xlat_topology	43
3.2.17.1	Usage	44
3.2.17.2	Options	44
3.2.18	iba_xlat_topology_cust	45
3.2.18.1	Usage	45
3.2.18.2	Options	45
3.2.19	s20tune	46
3.2.19.1	Usage	46
3.2.19.2	Options	46
3.3	Basic Setup and Administration Tools	47
3.3.1	fastfabric	47
3.3.1.1	Usage	47
3.3.1.2	Example	47
3.3.2	iba_config	47
3.3.2.1	Usage	47
3.3.2.2	Options	48
3.3.3	captureall	48



3.3.3.1	Usage.....	49
3.3.3.2	Options	49
3.3.3.3	Details.....	49
3.3.3.4	Host Capture Examples.....	50
3.3.3.5	Chassis Capture Examples.....	50
3.3.4	pingall	50
3.3.4.1	Usage.....	50
3.3.4.2	Options	51
3.3.4.3	Example	51
3.3.5	setup_ssh	51
3.3.5.1	Usage.....	51
3.3.5.2	Options	51
3.3.5.3	Example	52
3.3.5.4	Host Initial key exchange.....	52
3.3.5.5	Refreshing local systems known hosts	53
3.3.5.6	Chassis Initial key exchange.....	53
3.3.5.7	Chassis Refreshing local systems known hosts	53
3.3.6	cmdall	53
3.3.6.1	Usage.....	53
3.3.6.2	Options	54
3.3.6.3	Host Examples.....	54
3.3.6.4	Chassis Examples	54
3.4	File Management Tools.....	55
3.4.1	scpall.....	55
3.4.1.1	Usage.....	55
3.4.1.2	Options	55
3.4.1.3	Example	56
3.4.2	uploadall	56
3.4.2.1	Usage.....	56
3.4.2.2	Options	56
3.4.2.3	Example	57
3.4.3	downloadall	57
3.4.3.1	Usage.....	57
3.4.3.2	Options	57
3.4.3.3	Example	58
3.4.4	Simplified Editing of Node-Specific Files.....	58
3.4.5	Simplified Setup of Node-Generic Files	58
3.5	Fabric Analysis Tools.....	59
3.5.1	fabric_info.....	59
3.5.1.1	Usage.....	59
3.5.1.2	Options	59
3.5.1.3	Example	59
3.5.1.4	Example output	59
3.5.1.5	Output Definitions.....	59
3.5.2	showallports	60
3.5.2.1	Usage.....	60
3.5.2.2	Options	60
3.5.2.3	Example	61
3.5.3	iba_report	61
3.5.3.1	Usage.....	62
3.5.3.2	Options	62
3.5.3.3	Report Types (abridged)	62
3.5.3.4	Examples.....	63
3.5.3.5	Basics of Using iba_report.....	63
3.5.3.6	Scriptable output	68
3.5.3.7	Sample Output	68
3.5.3.8	Converting iba_report output to excel importable files - xml_extract.....	77



3.5.3.9	Remove All Specified XML Tags - xml_filter	80
3.5.3.10	Re-indenting XML files - xml_indent	81
3.5.4	iba_findgood	81
3.5.4.1	Usage	82
3.5.4.2	Options	82
3.5.4.3	Usage Examples	82
3.5.5	iba_saquery	82
3.5.5.1	Usage	83
3.5.5.2	Options	83
3.5.5.3	Node Types	83
3.5.5.4	GIDs	83
3.5.5.5	Output Types	83
3.5.6	iba_getvf	85
3.5.6.1	Usage	85
3.5.6.2	Options	85
3.5.6.3	Usage Examples	86
3.5.6.4	Sample Outputs	86
3.5.7	iba_getvf_env	86
3.5.8	iba_gen_ibnodes	86
3.5.8.1	Usage	86
3.5.8.2	Options	86
3.5.8.3	Environment	87
3.5.8.4	Usage Examples	87
3.5.9	iba_gen_chassis	87
3.5.9.1	Usage	87
3.5.9.2	Options	88
3.5.9.3	Environment	88
3.5.9.4	Usage Examples	88
3.5.10	iba_gen_esm_chassis	88
3.5.10.1	Usage	88
3.5.10.2	Options	88
3.5.10.3	Environment	89
3.5.10.4	Usage Examples	89
3.5.11	iba_smaquery	89
3.5.11.1	Usage	89
3.5.11.2	Options	89
3.5.11.3	Usage Examples	90
3.5.12	iba_paquery	91
3.5.12.1	Usage	91
3.5.12.2	Options	91
3.5.12.3	Output Types	92
3.5.12.4	Usage Examples	93
3.5.13	iba_pmaquery	93
3.5.13.1	Usage	93
3.5.13.2	Options	93
3.5.13.3	Usage Examples	94
3.5.13.4	Sample Outputs	94
3.5.14	iba_fequery	95
3.5.14.1	Usage	95
3.5.14.2	Options	95
3.5.14.3	Output Types	96
3.5.14.4	Examples	96
3.5.15	iba_ccaquery	97
3.5.15.1	Usage	97
3.5.15.2	Options	97
3.5.15.3	Examples	97
3.5.16	iba_extract_bad_links	98
3.5.16.1	Usage	98



3.5.16.2	Options	98
3.5.16.3	Examples.....	98
3.5.17	iba_disable_ports	98
3.5.17.1	Usage.....	98
3.5.17.2	Options	98
3.5.17.3	Environment	99
3.5.17.4	Examples.....	99
3.5.18	iba_enable_ports.....	99
3.5.18.1	Usage.....	99
3.5.18.2	Options	99
3.5.18.3	Examples.....	100
3.5.18.4	Environment	100
3.5.19	iba_disable_hosts.....	100
3.5.19.1	Usage.....	100
3.5.19.2	Options	100
3.5.19.3	Examples.....	100
3.5.20	iba_extract_lids	100
3.5.20.1	Usage.....	100
3.5.20.2	Options	101
3.5.20.3	Examples.....	101
3.6	Advanced Chassis Initialization and Verification	101
3.6.1	iba_chassis_admin	101
3.6.1.1	Usage.....	101
3.6.1.2	Options	101
3.6.1.3	For example	102
3.6.2	iba_chassis_admin Chassis Operations	103
3.6.2.1	upgrade.....	103
3.6.2.2	configure	103
3.6.2.3	reboot	104
3.6.2.4	getconfig	104
3.6.2.5	fmconfig	104
3.6.2.6	fmgetconfig.....	104
3.6.2.7	fmcontrol.....	104
3.7	Externally Managed Switch Initialization and Verification.....	104
3.7.1	iba_switch_admin.....	104
3.7.1.1	Usage.....	105
3.7.1.2	Options	105
3.7.1.3	Example	105
3.7.2	iba_switch_admin Operations.....	106
3.7.2.1	reboot	106
3.7.2.2	upgrade.....	106
3.7.2.3	configure	107
3.7.2.4	info	107
3.7.2.5	hwvdpd	107
3.7.2.6	ping	108
3.7.2.7	fwverify	108
3.7.2.8	capture.....	108
3.7.2.9	getconfig	108
3.8	Advanced Host Initialization and Verification	108
3.8.1	iba_host_admin	108
3.8.1.1	Usage.....	108
3.8.1.2	Options	108
3.8.1.3	Example	109
3.8.2	iba_host_admin Host Operations	110
3.8.2.1	load	110
3.8.2.2	upgrade.....	110
3.8.2.3	configipoib	110
3.8.2.4	reboot	111



3.8.2.5	sacache	111
3.8.2.6	ipoibping	111
3.8.2.7	mpiperf	111
3.8.2.8	mpiperfdeviation	112
3.8.3	Interpreting the iba_host_admin, iba_chassis_admin and iba_switch_admin log files	113
3.9	Health Check and Baselining Tools	114
3.9.1	Usage Model	114
3.9.2	Common Operations and Options	114
3.9.2.1	For example	115
3.9.2.2	For example	115
3.9.3	fabric_analysis	116
3.9.3.1	Usage	116
3.9.3.2	Options	116
3.9.3.3	Example	116
3.9.3.4	Health Check	117
3.9.3.5	Baseline	117
3.9.3.6	Full analysis	117
3.9.3.7	True Scale Fabric items checked against the baseline	118
3.9.3.8	True Scale Fabric Items that are also checked during health check	119
3.9.4	chassis_analysis	119
3.9.4.1	Usage	119
3.9.4.2	Options	119
3.9.4.3	Example	119
3.9.4.4	Health Check	120
3.9.4.5	Baseline	120
3.9.4.6	Full analysis	121
3.9.4.7	Chassis items checked against the baseline	122
3.9.4.8	Chassis Items also checked during healthcheck	123
3.9.5	hostsm_analysis	123
3.9.5.1	Usage	123
3.9.5.2	Options	123
3.9.5.3	Example	123
3.9.5.4	Health Check	124
3.9.5.5	Baseline	124
3.9.5.6	Full analysis	124
3.9.5.7	Host SM items checked against the baseline	124
3.9.5.8	Host SM items also checked during healthcheck	124
3.9.6	esm_analysis	124
3.9.6.1	Usage	124
3.9.6.2	Options	124
3.9.6.3	Example	125
3.9.6.4	Health Check	125
3.9.6.5	Baseline	125
3.9.6.6	Full analysis	126
3.9.6.7	Chassis SM items that are checked against the baseline	126
3.9.6.8	Chassis SM items also checked during healthcheck	126
3.9.7	all_analysis	126
3.9.7.1	Usage	126
3.9.7.2	Options	127
3.9.7.3	Example	127
3.9.7.4	Manual and Automated Usage	127
3.9.8	Re-establishing Health Check baseline	128
3.9.9	Interpreting the Health Check Results	129
3.9.10	Interpreting Health Check .changes Files	131
4.0	Complete Descriptions of Command Line Tools	137



4.1	Introduction	137
4.2	Basic Single Host Operations.....	137
4.2.1	fastfabric.....	137
4.2.1.1	Usage.....	137
4.2.1.2	Example	137
4.2.2	iba_config	137
4.2.2.1	Usage.....	137
4.2.2.2	Options	138
4.2.3	p1info, p2info	138
4.2.3.1	Usage.....	139
4.2.3.2	Options	139
4.2.4	p1stats, p2stats	139
4.2.4.1	Usage.....	139
4.2.5	clear_p1stats, clear_p2stats	139
4.2.5.1	Usage.....	139
4.2.6	iba_cabletest	139
4.2.6.1	Usage.....	139
4.2.6.2	Options	139
4.2.6.3	Environment Variables.....	140
4.2.6.4	Example	140
4.2.7	iba_capture	140
4.2.7.1	Usage.....	141
4.2.7.2	Options	141
4.2.7.3	Notes	141
4.2.7.4	Example	141
4.2.8	iba_expand_file.....	141
4.2.8.1	Usage.....	141
4.2.8.2	Options	142
4.2.8.3	Example	142
4.2.9	iba_linkanalysis.....	142
4.2.9.1	Usage.....	142
4.2.9.2	Options	142
4.2.9.3	Environment Variables.....	143
4.2.9.4	Example	143
4.2.10	iba_showmc	143
4.2.10.1	Usage.....	144
4.2.10.2	Options	144
4.2.10.3	Example	144
4.2.10.4	Environment Variables.....	144
4.2.11	iba_hca_rev.....	144
4.2.11.1	Usage.....	144
4.2.11.2	Options	144
4.2.12	iba_portenable.....	144
4.2.12.1	Usage.....	144
4.2.12.2	Options	145
4.2.12.3	Example	145
4.2.13	iba_portdisable	146
4.2.13.1	Usage.....	146
4.2.13.2	Options	146
4.2.13.3	Example	146
4.2.14	iba_portconfig.....	147
4.2.14.1	Usage.....	147
4.2.14.2	Options	147
4.2.14.3	Example	148
4.2.15	iba_portinfo.....	148
4.2.15.1	Usage.....	148
4.2.15.2	Options	148



4.2.15.3	Example.....	149
4.2.16	iba_portstats.....	149
4.2.16.1	Usage	149
4.2.16.2	Options.....	149
4.2.16.3	Example.....	149
4.2.17	iba_portclear.....	149
4.2.17.1	Usage	149
4.2.17.2	Options.....	149
4.2.17.3	Example.....	150
4.2.18	iba_resolve_hca_port	150
4.2.18.1	Usage	150
4.2.18.2	Options.....	150
4.2.19	iba_sorthosts	150
4.2.19.1	Usage	150
4.2.19.2	Options.....	150
4.2.19.3	Example input.....	151
4.2.19.4	Resulting output.....	151
4.2.20	iba_verifyhosts.....	151
4.2.20.1	Usage	151
4.2.20.2	Options.....	152
4.2.20.3	Environment	152
4.2.20.4	Example.....	153
4.2.21	iba_xlat_topology	153
4.2.21.1	Usage	154
4.2.21.2	Options.....	154
4.2.22	iba_xlat_topology_cust.....	155
4.2.22.1	Usage	156
4.2.22.2	Options.....	156
4.2.23	HCA port thresholding using iba_mon	156
4.2.23.1	Usage	156
4.2.23.2	Options.....	156
4.2.24	s20tune.....	157
4.2.24.1	Usage	157
4.2.24.2	Options.....	157
4.3	Basic Setup and Administration Tools	158
4.3.1	pingall.....	158
4.3.1.1	Usage	158
4.3.1.2	Options.....	158
4.3.1.3	Example.....	158
4.3.1.4	Environment Variables	158
4.3.2	check_rsh.....	159
4.3.2.1	Usage	159
4.3.2.2	Options.....	159
4.3.2.3	Example.....	159
4.3.2.4	Environment Variables	159
4.3.3	setup_ssh.....	159
4.3.3.1	Usage	160
4.3.3.2	Options.....	160
4.3.3.3	Example.....	160
4.3.3.4	Environment Variables	161
4.3.3.5	Host Initial Key Exchange	161
4.3.3.6	Host Refreshing Local Systems Known Hosts	162
4.3.3.7	Chassis Initial Key Exchange.....	162
4.3.3.8	Chassis Refreshing Local Systems Known Hosts.....	162
4.3.4	cmdall.....	163
4.3.4.1	Usage	163
4.3.4.2	Options.....	163



4.3.4.3	Host Examples.....	163
4.3.4.4	Chassis Examples	163
4.3.4.5	Environment Variables.....	164
4.3.5	captureall.....	164
4.3.5.1	Usage.....	164
4.3.5.2	Options	165
4.3.5.3	Host Capture Examples.....	166
4.3.5.4	Chassis Capture Examples.....	166
4.3.5.5	Environment Variables.....	166
4.4	File Management Tools.....	167
4.4.1	scpall.....	167
4.4.1.1	Usage.....	167
4.4.1.2	Options	167
4.4.1.3	Example	168
4.4.1.4	Environment Variables.....	168
4.4.2	uploadall	168
4.4.2.1	Usage.....	168
4.4.2.2	Options	169
4.4.2.3	Example	169
4.4.2.4	Environment Variables.....	169
4.4.3	downloadall	170
4.4.3.1	Usage.....	170
4.4.3.2	Options	170
4.4.3.3	Example	170
4.4.3.4	Environment Variables.....	171
4.4.4	Simplified Editing of Node-Specific Files.....	171
4.4.5	Simplified Setup of Node-Generic Files	171
4.5	Fabric Analysis Tools.....	172
4.5.1	fabric_info.....	172
4.5.1.1	Usage.....	172
4.5.1.2	Options	172
4.5.1.3	Environment Variables.....	172
4.5.1.4	Example output	173
4.5.1.5	The output is as follows	173
4.5.2	showallports	174
4.5.2.1	Usage.....	174
4.5.2.2	Options	174
4.5.2.3	Example	174
4.5.2.4	Environment Variables.....	175
4.5.3	iba_report	175
4.5.3.1	Usage.....	176
4.5.3.2	Options	176
4.5.3.3	Report Types.....	177
4.5.3.4	Point Syntax	179
4.5.3.5	Examples.....	180
4.5.3.6	Basics of Using iba_report.....	182
4.5.3.7	Simple Topology Verification.....	187
4.5.3.8	Advanced Topology Verification.....	188
4.5.3.9	Augmented Report Information.....	195
4.5.3.10	Focused Reports	195
4.5.3.11	Advanced Focus.....	196
4.5.3.12	Focus Examples.....	196
4.5.3.13	Scriptable output	197
4.5.3.14	Using iba_report to monitor for fabric changes.....	198
4.5.3.15	Sample Output	198
4.5.3.16	Snapshots.....	211
4.5.4	iba_reports.....	212
4.5.4.1	Usage.....	212



4.5.4.2	Options	212
4.5.4.3	Example.....	213
4.5.4.4	Environment Variables	213
4.5.5	Converting iba_report output to excel importable files - xml_extract.....	213
4.5.5.1	Usage	214
4.5.5.2	Options.....	214
4.5.5.3	Details	214
4.5.5.4	Sample Use and Output	215
4.5.5.5	Sample Scripts.....	216
4.5.6	iba_extract_perf	216
4.5.7	iba_extract_error.....	217
4.5.8	iba_extract_stat	217
4.5.9	iba_extract_stat2.....	217
4.5.10	iba_extract_link.....	218
4.5.11	Remove All Specified XML Tags - xml_filter	218
4.5.11.1	Usage	218
4.5.11.2	Options.....	218
4.5.12	Re-indenting XML files - xml_indent	218
4.5.12.1	Usage	219
4.5.12.2	Options.....	219
4.5.13	Creating iba_report topology_input files - xml_generate.....	219
4.5.13.1	Usage	219
4.5.13.2	Options.....	219
4.5.13.3	Details	219
4.5.13.4	Using xml_generate to create topology input files	220
4.5.14	Sample Script and Output - iba_gen_topology	221
4.5.14.1	iba_topology_links.txt.....	221
4.5.14.2	iba_topology_CAs.txt.....	221
4.5.14.3	iba_topology_SWs.txt	221
4.5.14.4	iba_topology_SMs.txt	222
4.5.14.5	Sample Output.....	222
4.5.15	iba_findgood	224
4.5.15.1	Usage	225
4.5.15.2	Options.....	225
4.5.15.3	Usage Examples.....	226
4.5.16	iba_saquery	226
4.5.16.1	Usage	226
4.5.16.2	Options.....	226
4.5.16.3	Node Types	227
4.5.16.4	GIDs.....	227
4.5.16.5	Output Types.....	227
4.5.17	iba_getvf	230
4.5.17.1	Usage	230
4.5.17.2	Options.....	231
4.5.17.3	Usage Examples.....	231
4.5.17.4	Sample Outputs	231
4.5.18	iba_getvf_env	231
4.5.19	iba_gen_ibnodes	231
4.5.19.1	Usage	231
4.5.19.2	Options.....	232
4.5.19.3	Environment.....	233
4.5.19.4	Usage Examples.....	233
4.5.20	iba_gen_chassis	233
4.5.20.1	Usage	233
4.5.20.2	Options.....	233
4.5.20.3	Environment.....	234



4.5.20.4	Usage Examples	234
4.5.21	iba_gen_esm_chassis	234
4.5.21.1	Usage.....	234
4.5.21.2	Options	234
4.5.21.3	Environment	234
4.5.21.4	Usage Examples	235
4.5.22	iba_fequery	235
4.5.22.1	Usage:	235
4.5.22.2	Options:	235
4.5.22.3	Output Types:	236
4.5.22.4	Usage Examples:	237
4.5.23	iba_smaquery	237
4.5.23.1	Usage.....	237
4.5.23.2	Options	237
4.5.23.3	Usage Examples	238
4.5.24	iba_paquery	239
4.5.24.1	Usage.....	239
4.5.24.2	Options	239
4.5.24.3	Output Types	240
4.5.24.4	Usage Examples	241
4.5.25	iba_pmaquery.....	241
4.5.25.1	Usage.....	241
4.5.25.2	Options	242
4.5.25.3	Usage Examples	242
4.5.25.4	Sample Outputs.....	242
4.5.26	iba_ccaquerry	243
4.5.26.1	Usage.....	243
4.5.26.2	Options	243
4.5.26.3	Examples.....	243
4.5.27	iba_smjobmgmt.....	244
4.5.27.1	Usage.....	244
4.5.27.2	Options	244
4.5.27.3	Show Options	245
4.5.27.4	Examples.....	246
4.5.28	iba_smjobgen	246
4.5.28.1	Usage.....	246
4.5.28.2	Options	247
4.5.28.3	Create Options	247
4.5.28.4	Examples.....	247
4.5.29	iba_extract_bad_links	249
4.5.29.1	Usage.....	250
4.5.29.2	Options	250
4.5.29.3	Examples.....	250
4.5.30	iba_disable_ports	250
4.5.30.1	Usage.....	250
4.5.30.2	Options	250
4.5.30.3	Environment	251
4.5.30.4	Examples.....	251
4.5.31	iba_enable_ports.....	251
4.5.31.1	Usage.....	251
4.5.31.2	Options	251
4.5.31.3	Examples.....	252
4.5.31.4	Environment	252
4.5.32	iba_disable_hosts.....	252
4.5.32.1	Usage.....	252
4.5.32.2	Options	252
4.5.32.3	Examples.....	252
4.5.33	iba_extract_lids	252



4.5.33.1	Usage	252
4.5.33.2	Options	252
4.5.33.3	Examples	253
4.6	Advanced Chassis Initialization and Verification	253
4.6.1	iba_chassis_admin	253
4.6.1.1	Usage	253
4.6.1.2	Options	253
4.6.1.3	Example	254
4.6.1.4	Environment Variables	255
4.6.2	iba_chassis_admin Chassis Operations	255
4.6.2.1	upgrade	255
4.6.2.2	configure	256
4.6.2.3	reboot	256
4.6.2.4	getconfig	256
4.6.2.5	fmconfig	257
4.6.2.6	fmgetconfig	257
4.6.2.7	fmcontrol	257
4.7	Externally Managed Switch Initialization and Verification	257
4.7.1	iba_switch_admin	257
4.7.1.1	Usage	257
4.7.1.2	Options	258
4.7.1.3	Example	258
4.7.1.4	Environment Variables	259
4.7.2	iba_switch_admin Operations	259
4.7.2.1	reboot	259
4.7.2.2	upgrade	260
4.7.2.3	configure	260
4.7.2.4	info	260
4.7.2.5	hwvpd	261
4.7.2.6	ping	261
4.7.2.7	fwverify	261
4.7.2.8	capture	261
4.7.2.9	getconfig	262
4.8	Advanced Host Initialization and Verification	262
4.8.1	iba_host_admin	262
4.8.1.1	Usage	262
4.8.1.2	Options	262
4.8.1.3	Example	263
4.8.1.4	Details	263
4.8.1.5	Environment Variables	264
4.8.2	iba_host_admin Host Operations	265
4.8.2.1	load	265
4.8.2.2	upgrade	265
4.8.2.3	configipoib	265
4.8.2.4	reboot	266
4.8.2.5	sacache	266
4.8.2.6	ipoibping	266
4.8.2.7	mpiperf	266
4.8.2.8	mpiperfdeviation	267
4.9	Interpreting the iba_host_admin, iba_chassis_admin and iba_switch_admin log files	269
4.10	Health Check and Baselining Tools	270
4.10.1	Usage Model	270
4.10.2	Common Operations and Options	271
4.10.2.1	Example	271
4.10.3	fabric_analysis	272
4.10.3.1	Usage	272
4.10.3.2	Options	272



4.10.3.3	Example	273
4.10.3.4	Environment Variables	273
4.10.3.5	Details	274
4.10.3.6	Health Check	274
4.10.3.7	Baseline	275
4.10.3.8	Full analysis	275
4.10.3.9	True Scale Fabric items checked against the baseline	276
4.10.3.10	True Scale Fabric Items that are also checked during health check	276
4.10.4	chassis_analysis	277
4.10.4.1	Usage	277
4.10.4.2	Options	277
4.10.4.3	Example	277
4.10.4.4	Environment Variables	277
4.10.4.5	Details	278
4.10.4.6	Health Check	278
4.10.4.7	Baseline	278
4.10.4.8	Full analysis	279
4.10.4.9	Chassis items checked against the baseline	280
4.10.4.10	Chassis Items also checked during healthcheck	281
4.10.5	hostsm_analysis	281
4.10.5.1	Usage	281
4.10.5.2	Options	281
4.10.5.3	Example	281
4.10.5.4	Environment Variables	282
4.10.5.5	Health Check	282
4.10.5.6	Baseline	282
4.10.5.7	Full analysis	282
4.10.5.8	Host SM items checked against the baseline	282
4.10.5.9	Host SM items also checked during healthcheck	282
4.10.6	esm_analysis	282
4.10.6.1	Usage	283
4.10.6.2	Options	283
4.10.6.3	Example	283
4.10.6.4	Environment Variables	283
4.10.6.5	Health Check	284
4.10.6.6	Baseline	284
4.10.6.7	Full analysis	284
4.10.6.8	Chassis SM items that are checked against the baseline	285
4.10.6.9	Chassis SM items also checked during healthcheck	285
4.10.7	all_analysis	285
4.10.7.1	Usage	285
4.10.7.2	Options	285
4.10.7.3	Example	286
4.10.7.4	Environment Variables	286
4.10.8	Manual and Automated Usage	287
4.10.9	Re-establishing Health Check baseline	287
4.10.10	Interpreting the Health Check Results	288
4.10.11	Interpreting Health Check .changes Files	291
5.0	MPI Sample Applications	295
5.1	Latency/Bandwidth Deviation Test	296
5.2	OSU Latency	298
5.3	OSU Latency2	298
5.4	OSU Latency 3	298
5.5	OSU Multi Latency3	299
5.6	OSU Bandwidth	299
5.7	OSU Bandwidth2	299



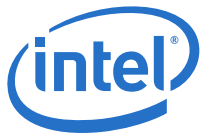
5.8	OSU Bandwidth3.....	299
5.9	OSU Multi Bandwidth3.....	300
5.10	OSU Bidirectional Bandwidth.....	300
5.11	OSU Bidirectional Bandwidth3.....	300
5.12	OSU All to All 3.....	300
5.13	OSU Broadcast 3.....	301
5.14	OSU Multiple Bandwidth/Message Rate.....	301
	5.14.1 Latency Test.....	302
	5.14.2 Multi-threaded Latency Test (only applicable for MVAPICH2 with threading support enabled).....	302
	5.14.3 Bandwidth Test.....	302
	5.14.4 Bidirectional Bandwidth Test.....	303
	5.14.5 Multiple Bandwidth / Message Rate test.....	303
	5.14.6 Multi-pair Latency Test.....	303
	5.14.7 Broadcast Latency Test.....	303
	5.14.8 One-Sided Put Latency Test (only applicable for MVAPICH2).....	303
	5.14.9 One-Sided Get Latency Test (only applicable for MVAPICH2).....	303
	5.14.10 One-Sided Put Bandwidth Test (only applicable for MVAPICH2).....	304
	5.14.11 One-Sided Get Bandwidth Test (only applicable for MVAPICH2).....	304
	5.14.12 One-Sided Put Bidirectional Bandwidth Test (only applicable for MVAPICH2).....	304
	5.14.13 Accumulate Latency Test (only applicable for MVAPICH2).....	304
5.15	High Performance Linpack (HPL).....	305
5.16	Pallas.....	306
5.17	Intel MPI Benchmark.....	307
5.18	MPI Fabric Stress Test.....	307
	5.18.1 All HCA latency.....	307
	5.18.2 run cabletest.....	308
	5.18.3 run batch cabletest.....	309
	5.18.3.1 Environment.....	310
	5.18.3.2 Examples.....	310
	5.18.4 gen_group_hosts.....	310
	5.18.5 run_multibw.....	311
	5.18.6 run_nxnlatbw.....	311
5.19	MPI Batch run_* scripts.....	311
	5.19.0.1 Usage.....	312
	5.19.0.2 Options.....	312
	5.19.0.3 Environment.....	312
	5.19.0.4 Examples.....	312
	5.19.1 SHMEM Batch run_* scripts.....	312
A	Port Counters Overview.....	315
A.1	Link Integrity.....	315
A.2	SymbolError Counter.....	315
	A.2.1 LinkErrorRecovery Counter.....	315
	A.2.2 LinkDowned Counter.....	316
	A.2.3 PortRcvErrors.....	316
	A.2.4 LocalLinkIntegrityErrors.....	317
	A.2.5 ExcessiveBufferOverrunErrors.....	317
A.3	Sma Congestion.....	317
	A.3.1 VL15Dropped.....	317
A.4	Congestion.....	318
	A.4.1 PortXmitDiscards.....	318
	A.4.2 PortXmitWait.....	318
	A.4.3 PortXmitCongestion.....	318
A.5	Security.....	318



A.5.1	PortXmitConstraintErrors	318
A.5.2	PortRcvConstraintErrors	319
A.6	Routing	319
A.6.1	PortRcvSwitchRelayErrors	319
A.6.2	Data Movement	319
A.6.3	PortXmitData	320
A.6.4	PortRcvData	320
A.6.5	PortXmitPkts	320
A.6.6	PortRcvPkts	320
A.6.7	PortXmitWait	320
A.7	Other	321
A.7.1	PortRcvRemotePhysicalErrors	321

Tables

1	Basic Command Line Tools listed by groups	31
2	iba_xlat_topology_cust	45
3	Input Combinations	84
4	Possible issues found in health check .changes files	133
5	iba_xlat_topology_cust	155
6	Input Combinations	229
7	Possible issues found in health check .changes files	293
8	Rank Assignment	301



Revision History

Date	Revision	Description
May 2013	001US	Initial Intel® release
January 2014	002US	Update iba_verifyhosts. See pages 41 and 151 .
July 2014	003US	Updated Support link in Section 1.5, "Technical Support" on page 21

§ §



1.0 Introduction

The Intel® FastFabric Toolset provides numerous powerful features, however for the initial user, the rich set of capabilities can be overwhelming. This reference guide is organized for ease of use at all levels of understanding. All of the commands are organized into groups, and the groups of commands are organized in sections that outline the commonly used options and operations. For new users, it is recommended to first learn the basic command line tools provided in the [Basic Command Line Tools](#) section. For a complete list and detailed description of the command line tools refer to [Complete Descriptions of Command Line Tools](#) section.

Some of the commands are only applicable when Linux is being used and will be marked with **(Linux)**. Similarly some of the commands are only applicable when Intel® True Scale Fabric OFED+ Host Software is being used on the hosts and will be marked with **(Host)**. All commands which are applicable only when Intel® Switches, or Chassis are being used will be marked with **(Switch)**. All remaining commands are generally applicable to all environments and will be marked with **(All)**.

Note: Some of the Linux commands may be applicable to other Unix-like operating systems. These may be used if it is desired to enable use of non-True Scale specific Intel® FastFabric Toolset (such as `cmdall`) against the given hosts.

The Intel® FastFabric Toolset is installed in directories which are part of the standard Linux root PATH. Most of the tools are installed in `/sbin`.

1.1 Common Tool Options

There are some common options to the assorted command line tools. These options are applicable to most of the tools:

1.1.1 -?

Will display Basic Usage information for any of the commands (as will any invalid option)

1.1.2 --help

Will display Complete Usage information for most of the commands

1.1.3 -p

Runs the operation/command in parallel. This means the operation is performed simultaneously on batches of 20 hosts. As such this option allows the overall time of an operation to be much lower. However, a side effect is that any output from the command will be bursty and intermingled. Therefore this option should be used for commands where there is no output or the output is of limited interest. For some commands (such as `scpall`), this will perform the operation in a quiet mode to limit output. If the user wants to change the number of parallel operations export `FF_MAX_PARALLEL=#` where # is the new number (such as 30).

For more advanced operations (such as `iba host admin`, `iba_chassis_admin` and `iba_switch_admin`), parallel operation is the default mode.

Parallel operation can also be disabled by setting `FF_MAX_PARALLEL` to 1.



1.1.4 -s

Prompt for password for admin on chassis or root on host. By default Intel® FastFabric Toolset operations against Intel® Chassis (such as `cmdall`, `captureall`, `showallports`, and `iba_chassis_admin`) obtain the chassis admin password from the `FF_CHASSIS_ADMIN_PASSWORD` environment variable which may be directly exported or part of `fastfabric.conf`. Alternatively the `-s` option may be used in which case the chassis admin password will be prompted for interactively. The password is prompted for once and the same password is then used to login to each chassis during the operation.

For hosts, this option is only applicable to `setup_ssh`.

Note: All versions of Intel® 12000 Chassis firmware permit ssh keys to be configured within the chassis for secure password-less login. In which case there is no need to configure a `FF_CHASSIS_ADMIN_PASSWORD` and `FF_CHASSIS_LOGIN_METHOD` can be `ssh`. Intel recommends to set up a secure ssh password-less login using `setup_ssh -C`. Consult the *Intel® True Scale Fabric Switches 12000 Series Users Guide* for more information.

1.1.5 -c

Specifies that the given operation should be performed against chassis. By default many Intel® FastFabric Toolset operations are performed against hosts. However, selected Intel® FastFabric Toolset commands (such as `cmdall`, `pingall`, and `captureall`) can also operate against Intel® Internally Managed Chassis. When `-c` is specified, the operation will be performed against chassis instead of hosts. The selection of chassis options are discussed in [“Selection of Chassis” on page 24](#)).

1.1.6 -I

Specifies that the given operation should be performed against externally-managed switches (such as the Intel® 12200 Switch). By default Intel® FastFabric Toolset operations are performed against hosts. However, selected Intel® FastFabric Toolset commands (such as `showallports`) can also operate against externally-managed switches. When specified, the operation will be performed against switches instead of hosts. The selection of switches options are discussed in [“Selection of Switches” on page 26](#)).

1.2 Intended Audience

This manual is intended to provide network administrators and other qualified personnel a reference to command line interface (CLI) command options, definitions, and examples for the FastFabric software.

1.3 Related Materials

- *Intel® True Scale Fabric Suite FastFabric User Guide*
- *Intel® True Scale Fabric Software Installation Guide*
- *Intel® True Scale Fabric Suite Software Release Notes*

1.4 Documentation Conventions

This guide uses the following documentation conventions:

- **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, and links to Web sites are shown in **underlined blue**. For example:
 - Table 9-2 lists problems related to the user interface and remote agent.
 - See “Installation Checklist” on page 3-6.
 - For more information, visit www.intel.com.
- Text in **bold** font indicates user interface elements such as a menu items, buttons, check boxes, or column headings. For example:
 - Click the **Start** button, point to **Programs**, point to **Accessories**, and then click **Command Prompt**.
 - Under **Notification Options**, select the **Warning Alarms** check box.
- Text in **Courier** font indicates a file name, directory path, or command line text. For example:
 - To return to the root directory from anywhere in the file structure:
Type `cd /root` and press **ENTER**.
 - Enter the following command: `sh ./install.bin`
- Key names and key strokes are indicated with **uppercase**:
 - Press **ctrl+P**.
 - Press the **up arrow** key.
- Text in *italics* indicates terms, emphasis, variables, or document titles. For example:
 - For a complete listing of license agreements, refer to the Intel® *Software End User License Agreement*.
 - What are *shortcut keys*?
 - To enter the date type *mm/dd/yyyy* (where *mm* is the month, *dd* is the day, and *yyyy* is the year).
- Topic titles between quotation marks identify related topics either within this manual or in the online help, which is also referred to as *the help system* throughout this document.

1.5 Technical Support

Intel True Scale Technical Support for products under warranty is available during local standard working hours excluding Intel Observed Holidays. For customers with extended service, consult your plan for available hours. For Support information, see the Support link at www.intel.com/truescale.

1.6 License Agreements

Refer to the Intel® *Software End User License Agreement* for a complete listing of all license agreements affecting this product.







2.0 Selection of Devices

2.1 Selection of Hosts

For operations that are performed against a set of hosts, there are multiple ways to specify the hosts on which to operate:

- Small sets of hosts can be easily specified on the command line using the `-h` option discussed below.
- When multiple commands are performed against the same small set of hosts, the environment variable `HOSTS` can be used to specify a space separated list of hosts.
- For groups of hosts that will be used often, a file may be created listing the hosts. The default file is `/etc/sysconfig/iba/hosts` that should list all hosts in the cluster except the host running Intel® FastFabric Toolset itself. Such a file may then be specified using the `-f` option or the `HOSTS_FILE` environment variable.

Within the tools, the options are considered in the following order, the first item listed below that is specified is used for the given command.

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

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

2.1.1 Host List Files

The `-f` option may be used to provide the name of a file containing the list of hosts on which to operate. The default is `/etc/sysconfig/iba/hosts`. In some fabrics it may be useful to create multiple files in `/etc/sysconfig/iba` representing different subsets of the fabric on which the user may operate. For example:

```
/etc/sysconfig/iba/hosts-mpi: list of MPI hosts
/etc/sysconfig/iba/hosts-fs: list of file server hosts
/etc/sysconfig/iba/hosts: list of all hosts except for the Intel® FastFabric
Toolset node
/etc/sysconfig/iba/allhosts: list of all hosts including the Intel®
FastFabric Toolset node
```

If a relative path is specified for the `-f` option, the current directory will be checked first, followed by `/etc/sysconfig/iba/`

2.1.1.1 Host List File Format

Below is a sample host list file:

```
# this is a comment

192.168.0.4 # host identified by IP address

n001      # host identified by resolvable TCP/IP name

include /etc/sysconfig/iba/hosts-mpi # included file
```



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

Hosts may be specified by IP address or a resolvable TCP/IP host name. Typically, host names are used for readability. Also, some Intel® FastFabric Toolset commands will translate the supplied host names to IPoIB hostnames, in which case names are generally easier to translate than numeric IP addresses. Typically management network hostnames are specified. However, if desired, IPoIB hostnames or IP addresses may be used. This can accelerate large file transfers and other operations.

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

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

2.1.2 Explicit host names

When hosts are explicitly specified using the `-h` option or the `HOSTS` environment variable, a space separated list of host names (or IP addresses) may be supplied. For example: `-h host1 host2 host3`.

2.2 Selection of Chassis

For operations which are performed against a set of chassis, there are multiple ways to specify the chassis on which to operate:

- Small sets of chassis can be easily specified on the command line using the `-H` option discussed below
- When multiple commands will be performed against the same small set of chassis, the environment variable `CHASSIS` can be used to specify a space separated list of chassis.
- For groups of chassis which will be used often, a file may be created listing the chassis. The default file is `/etc/sysconfig/iba/chassis` which should list all chassis in the cluster. Such a file may then be specified using the `-F` option or the `CHASSIS_FILE` environment variable.

Within the tools, the options are considered in the following order, the first item listed below that is specified is used for the given command.

1. `-H` option
2. `CHASSIS` environment variable
3. `-F` option
4. `CHASSIS_FILE` environment variable
5. `/etc/sysconfig/iba/chassis` file

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



2.2.1 Chassis List Files

The `-F` option may be used to provide the name of a file containing the list of Intel® Chassis to operate on. The default is `/etc/sysconfig/iba/chassis`. In some fabrics it may be useful to create multiple files in `/etc/sysconfig/iba` representing different subsets of the fabric on which the user may operate. For example:

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

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

2.2.1.1 Chassis List File Format

Below is a sample chassis file:

```
# this is a comment

192.168.0.5 # chassis IP address

edge1 # chassis resolvable TCP/IP name

include /etc/sysconfig/iba/corechassis # included file
```

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

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

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

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

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

2.2.2 Explicit Chassis names

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

2.2.3 Selection of slots within a chassis

Normally, operations are performed against the management card in the chassis. For operations such as `cmdall`, the command is executed against the management interface for the given chassis. For more sophisticated operations such as firmware update, a directory with firmware for each chassis card type can be supplied and all cards in the chassis will be updated with the appropriate firmware from that directory.



However, in some cases it may be desirable to perform operations against a specific subset of cards within the chassis. In this case the chassis IP address or name within a chassis list or a chassis file can be augmented with a list of slot numbers on which to operate. This is done in the form:

```
chassis:slot1,slot2,...
```

For example:

```
i9k229:0
```

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

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

Note: There must be no spaces within the chassis name and/or slot list.

This format is used by `cmdall` and `chassis firmware update`. This format may be used any place a chassis name or IP address is valid, such as the `-H` option, the `CHASSIS` environment variable or `chassis list` files. The slot number specified is ignored on some operations (such as `pingall`). Only slots containing management cards may be specified with this format. For the remainder of slot usages in the chassis, the `chassisQuery` command can be executed against a given chassis to identify which slots have management cards.

Note: For any operation, care should be taken that a given chassis is listed only once with all relevant slots as part of that single specification. This is important so that parallel operations do not cause conflicting concurrent operations against a given chassis.

2.3 Selection of Switches

For operations that are performed against a set of fixed configuration externally-managed switches, there are multiple ways to specify the switch on which to operate:

- Small sets of switches can be easily specified on the command line using the `-N` option discussed below.
- When multiple commands will be performed against the same small set of switches, the environment variable `IBNODES` can be used to specify a space separated list of switches
- For groups of switches which will be used often, a file may be created listing the switches. The default file is `/etc/sysconfig/iba/ibnodes` that should list all switches in the cluster. Such a file may then be specified using the `-L` option or the `IBNODES_FILE` environment variable.

Within the tools, the options are considered in the following order, the first item listed below which is specified is used for the given command.

1. `-N` option
2. `IBNODES` environment variable
3. `-L` option
4. `IBNODES_FILE` environment variable
5. `/etc/sysconfig/iba/ibnodes` file

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



2.3.1 Switch List Files

The `-L` option may be used to provide the name of a file containing the list of Intel® Switches on which to operate. The default is `/etc/sysconfig/iba/ibnodes`. In some fabrics it may be useful to create multiple files in `/etc/sysconfig/iba` representing different subsets of the fabric on which the user may operate.

If a relative path is specified for the `-L` option or `IBNODES_FILE`, the current directory will be checked first, followed by `/etc/sysconfig/iba/`.

2.3.1.1 Switch List File Format

Below is a sample switch list file:

```
# this is a comment

0x00066a00d9000138,i9k138 # Node GUID with desired Name

0x00066a00d9000139,i9k139 # Node GUID with desired Name

0x00066a00d9000140:1:2,i9k140 # Node GUID with port and Name

include /etc/sysconfig/iba/moreswitches # included file
```

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

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

The `iba_gen_ibnodes` can be used to help locate externally managed switches in the fabric and generate an `ibnodes` file. The `iba_gen_ibnodes` tool will by default provide the proper distance value relative to the FastFabric node from which it was run. This capability requires use of InfiniBand® Trade Association (IBTA) standard TraceRecord queries which are not supported by openSM, but can be supplied by the Intel® True Scale Fabric Suite Fabric Manager. Alternatively the `iba_gen_ibnodes -R` option can suppress generation of this field.

In a typical pure fat tree topology with externally managed switches as edge switches and internally managed switches as core switches, the user can also easily manually specify proper distance by simply specifying 1 for the distance value of the switch next to the FastFabric node. Note that in such a topology all other switches are an equal length from the FastFabric node and a missing distance value will cause them to be treated as having a distance value which is larger than any other found in the file. Therefore the other switches would be rebooted first and the FastFabric node's switch would be rebooted last.

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

The `hca:port` may be used to specify which local port (subnet) to use to access the switch. If this is omitted, all local ports specified will be checked for the switch and the first port found to be able to access the switch will be used to access it. See the *FastFabric Command Line Interface Reference Guide* for more information about how to specify and `hca:port` value.



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

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

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

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

```
iba_saquery -t sw -o nodeguid
```

Note: The preceding command will report all switch node GUIDs, including those of internally-managed chassis such as the Intel® 12000. GUIDs for internally-managed chassis cannot be specified for use in the `ibnodes` file.

2.3.2 Explicit Switch names

When switches are explicitly specified using the `-N` option or the `IBNODES` environment variable, a space separated list of GUIDs (optionally with `hca:port` and/or name) may be supplied. For example:

```
-N '0x00066a00d9000138,i9k138 0x00066a00d9000139,i9k139'
```

2.4 Selection of local Ports (subnets)

Many commands (such as `iba_reports`, `fabric_info`, `iba_switch admin`, `fabric_analysis`, and `all_analysis`) permit a specific set of local Host Channel Adapter (HCA) ports to be used for fabric access. The default is to use the first active port. However, for Fabric Management Nodes connected to more than one subnet, it is necessary to specify the local HCA and port such that the desired subnet will be analyzed. When the non-default behavior is desired, there are multiple ways to specify the local ports to use:

- Small sets of ports can be easily specified on the command line using the `-p` option discussed below.
- When multiple commands will be performed against the same small set of ports, the environment variable `PORTS` can be used to specify a space separated list of ports.
- For groups of ports that will be used often, a file may be created listing the ports. The default file is `/etc/sysconfig/iba/ports`. That file should list all local ports connected to unique subnets. Such a file may then be specified using the `-t` option or the `PORTS_FILE` environment variable.

Within the tools, the options that are considered in the following order, the first item listed below that is specified is used for the given command.

1. `-p` option
2. `PORTS` environment variable
3. `-t` option



4. PORTS_FILE environment variable
5. /etc/sysconfig/iba/ports file
6. default of the first active port on system (0:0 port specification)

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

2.4.1 Port List Files

The `-t` option or the `PORTS_FILE` environment variable may be used to provide the name of a file containing the list of local HCA ports to use. The default is `/etc/sysconfig/iba/ports`. In some fabrics it may be useful to create multiple files in `/etc/sysconfig/iba` representing different subsets of the ports on which the user may operate. For example:

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

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

2.4.1.1 Port List File Format

Below is a sample port list file:

```
# this is a comment

1:1 # first port on 1st HCA

1:2 # second port on 1st HCA

2:1 # first port on 2nd HCA

3:0 # first active port on 3rd HCA

include /etc/sysconfig/iba/ports-plane2 # included file
```

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

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

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

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

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



2.4.2 Explicit ports

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

§ §



3.0 Basic Command Line Tools

3.1 Introduction

This section provides a description of each Intel® FastFabric Toolset command line tool and its most commonly used parameters. For new users, it is recommended to first learn the basic command line tools provided in this section. For a complete list and detailed description of the command line tools refer to [Complete Descriptions of Command Line Tools](#) section.

Table 1 list the tools in groups according to where they are used. These tools are part of the FastFabric enablement Stack Tools:

Table 1. Basic Command Line Tools listed by groups

Group	Tool	Summary
Basic Single Host Operations	<code>clear_p1stats</code> , <code>clear_p2stats</code>	Clears the port performance counters for port 1 or port 2 respectively.
	<code>iba_capture</code>	Captures supporting information for a problem report from the host.
	<code>iba_hca_rev</code>	
	<code>iba_mon</code>	A daemon process which can monitor the port state and statistics of all HCAs on the given host.
	<code>iba_portconfig</code>	
	<code>iba_portdisable</code>	
	<code>iba_portenable</code>	
	<code>iba_portinfo</code>	
	<code>p1info</code> , <code>p2info</code>	Shows the port status for port 1 or port 2 respectively.
	<code>p1stats</code> , <code>p2stats</code>	Shows the port performance counters for port 1 or port 2 respectively.
Basic Setup and Administration Tools	<code>iba_portstats</code>	
	<code>captureall</code>	
Fabric Analysis Tools	<code>fabric_info</code>	Provides a brief summary of the components in the fabric.
	<code>iba_saquery</code>	

These tools are described in the following sections.

3.2 Basic Single Host Operations

These basic command line tools are available on each host where the Intel® True Scale Fabric OFED+ Host Software True Scale Fabric Stack Tools have been installed. These tools are mainly used to enable Intel® FastFabric Toolset operations against cluster nodes, however they can also be directly used on an individual host.

3.2.1 `clear_p1stats`, `clear_p2stats`

(Host) Clears the port performance counters for port 1 or port 2 respectively. On systems with more than 1 HCA, port 1 or port 2 counters on all HCAs will be cleared.



3.2.1.1 Usage

```
clear_plstats
```

```
clear_p2stats
```

3.2.2 iba_capture

(Host) Captures critical system information into a zipped tar file. The resulting tar file should be sent to Customer Support along with any IntelIB problem report regarding this system.

3.2.2.1 Usage

```
iba_capture [-d detail] output_tar_file
```

3.2.2.2 Options

-d detail - level of detail of capture

1 - Normal - Obtains local information from host

2 - Fabric - In addition to "Normal", also obtains basic fabric information by queries to the SM and fabric error analysis using iba_report.

3 - Fabric+FDB - In addition to "Fabric", also obtains all the switch forwarding tables from the SM and the server multicast membership.

4 - Analysis - In addition to "Fabric+FDB", also obtains all_analysis results. If all_analysis has not yet been run, it is run as part of the capture.

Note:

Detail levels 2-4 can be used when fabric operational problems occur. If the problem is most likely node specific, detail level 1 should be sufficient. Detail levels 2-4 require an operational Fabric Manager. Typically your support representative will request a given detail level. If a given detail level takes excessively long or fails to be gathered, try a lower detail level.

For detail levels 2-4, the additional information is only available on a node with Intel® FastFabric Toolset installed. The information is gathered for every fabric specified in the /etc/sysconfig/iba/ports file.

output_tar_file - The name of a file to be created by iba_capture. The file name specified will be overwritten if it already exists. It is recommended to use the .tgz suffix in the filename supplied. If the filename given does not have a .tgz suffix, the .tgz suffix will be added.

3.2.2.3 Examples

```
iba_capture mycapture.tgz
```

```
iba_capture -d 3 030127capture.tgz
```

Note:

The resulting host capture file can require significant amounts of space on the host. The actual size will vary but sizes can be multiple megabytes. As such, it is recommended to ensure that adequate disk space is available on the host system.

3.2.3 iba_showmc

(Linux) Displays the True Scale Multicast groups created for the fabric along with the CA ports which are a member of each multicast group. This command can be helpful when attempting to analyze or debug True Scale multicast usage by applications or ULPs such as IPoIB.



3.2.3.1 Usage

```
iba_showmc [-v]
```

or

```
iba_showmc --help
```

3.2.3.2 Options

--help - produce full help text

-v - verbose output, show name of each member

3.2.4 iba_hca_rev

(Linux) Displays information about the HCAs in the system including model numbers, serial numbers, board revisions, and other HCA model-specific information.

3.2.4.1 Usage

```
iba_hca_rev [-v]
```

3.2.4.2 Options

-v - reports additional information about Mellanox adapter firmware, including detailed output of the configuration options and verification of the firmware image.

3.2.5 iba_mon

(Host) iba_mon is a daemon process which can be started on individual hosts to monitor the port state and statistics of all HCAs on the given host.

3.2.5.1 Usage

```
iba_mon [-v|-q] [-d] [-c file] [-f facility]
```

3.2.5.2 Options

-v - verbose output

-q - no output

-d - daemon (detach from terminal)

-c - config file, default is /etc/sysconfig/iba/iba_mon.conf

-f - syslog facility, default is local6

Normally, iba_mon is run as a background process started by the /etc/init.d/iba_mon initialization script. The iba_config or INSTALL commands may be used to configure iba_mon to be started automatically at system boot time.

The iba_mon.conf file (see *Intel® True Scale Fabric Suite FastFabric User Guide* for more information) defines the statistics that iba_mon will monitor and how often it will clear them for threshold analysis.



When `iba_mon` detects a port state change (for example, a port going down or becoming active) it will log output to syslog at the syslog facility level specified. Similarly, when `iba_mon` detects a configured threshold has been exceeded for a statistic over the specified interval, it will log the affected statistic and its value over the interval.

Note: It is recommended to not run `iba_mon` when fabric level tools (such as `all_analysis`, `fabric_analysis`, `iba_report`, **iba_reports** or the Intel® True Scale Fabric Suite Fabric Manager's Performance Manager) are being used for port statistics analysis. Alternatively, if desired, `iba_mon` can be run provided the statistics being centrally-monitored are configured with a threshold of 0 in `iba_mon.conf`, such that `iba_mon` will not monitor or clear the given statistic.

3.2.6 iba_portconfig

(Host or Switch) Controls configuration and state of a specified HCA port on the local host or a remote switch.

3.2.6.1 Usage

```
iba_portconfig [-v] [-D] [-l lid [-m dest_port]] [-h hca] [-p port] [-z] [-S state]
               [-P physstate] [-w width] [-s speed]
```

3.2.6.2 Options

- v – Verbose output
- D – Do not cycle port through disabled physstate
- l lid – Destination lid, default is local port
- m dest_port – Destination port, default is port with given lid useful to access switch ports
- h hca – HCA to send by/to, default is 1st HCA
- p port – Port to send by/to, default is 1st port
- z – Do not get port info first, clear most port attributes
- S state – New State (default is 0)
 - 0 – no-op
 - 1 – down
 - 2 – init
 - 3 – armed
 - 4 – active
- P physstate – New physical State (default is 0)
 - 0 – no-op
 - 1 – sleep
 - 2 – polling
 - 3 – disabled
- s speed – New link speeds enabled (default is 0)
 - 0 – no-op
 - 1 – 2.5 Gb/s
 - 2 – 5 Gb/s
 - 4 – 10 Gb/s



To enable multiple speeds, use the sum of the desired speeds.

255 – Enable all speeds supported by given HCA port

`-w width` – New link widths enabled (default is 0)

0 – no-op

1 – 1x

2 – 4x

4 – 8x

8 – 12x

To enable multiple widths, use sum of desired widths

255 – Enable all widths supported by given HCA port

`-K mkey` – SM management key to access remote ports

3.2.6.3 Example

```
iba_portconfig -w 1
```

```
iba_portconfig -p 2 -h 2 -w 3
```

The port configuration is transient in nature. If the given host is rebooted or its True Scale Fabric Stack is restarted, the port will revert to its default configuration and state. Typically, the default state is to have the port enabled with all speeds and widths supported by the given HCA port enabled.

To access switch ports using this command, the `-l` and `-m` options must be given. The `-l` option will specify the lid of switch port 0 (the logical management port for the switch) and `-m` will specify the actual switch port to access. If SMA mkeys are being used, the `-K` option will also be needed. By default the Intel® SM does not use SMA mkeys, in which case this option does not need to be used.

Note: On many DDR HCA and switch models, in order to enable DDR (5.0Gb) operation, SDR (2.5 Gb) operation must also be enabled. For example, the `-s2` option may yield a port which does not come up, in which case `-s3` is preferred.

Note: The `/etc/init.d/iba_portconfig` script is provided as an example of changing port speed everytime the server boots. This script can be edited then scheduled using `chkconfig` to control link settings on any set of HCA ports

Caution: When using this command to disable or reconfigure switch ports, if the final port in the path between the Fabric Management Node and the switch is disabled or fails to come online, then `iba_portenable` will not be able to reenables it. In which case the switch CLI and/or a switch reboot may be needed to correct the situation.

3.2.7 iba_portdisable

(Host or Switch) Disables a specified HCA port on the local host or a remote switch. May also be used to change port operational parameters as part of disabling the port.

3.2.7.1 Usage

```
iba_portdisable [-v] [-l lid [-m dest_port]] [-h hca] [-p port] [-w width] [-s speed] [-K mkey]
```

3.2.7.2 Options

`-v` – Verbose output



- l *lid* – Destination lid, default is local port
- m *dest_port* – Destination port, default is port with given lid useful to access switch ports
- h *hca* – HCA to send by/to, default is 1st HCA
- p *port* – Port to send by/to, default is 1st port
- s *speed* – New link speeds enabled (default is 0)
 - 0 – no-op
 - 1 – 2.5 Gb/s
 - 2 – 5 Gb/s
 - 4 – 10 Gb/sTo enable multiple speeds, use the sum of desired speeds.
255 – Enable all speeds supported by given HCA port
- w *width* – New link widths enabled (default is 0)
 - 0 – no-op
 - 1 – 1x
 - 2 – 4x
 - 4 – 8x
 - 8 – 12xTo enable multiple widths, use sum of desired widths
255 – Enable all widths supported by given HCA port
- K *mkey* – SM management key to access remote ports

3.2.7.3 Example

```
iba_portdisable
```

```
iba_portdisable -p 2 -h 2
```

The port disabled state is transient in nature. If the given host is rebooted or its True Scale Fabric stack is restarted, the port will revert to its default configuration and state. Typically, the default state has the port enabled with all speeds and widths supported by the given HCA port enabled.

To access switch ports using this command, the -l and -m options must be given. The -l option will specify the lid of switch port 0 (the logical management port for the switch) and -m will specify the actual switch port to access. If SMA mkeys are being used, the -K option will also be needed. By default the Intel® SM does not use SMA mkeys, in which case this option does not need to be used.

Note: On many DDR HCA and switch models, in order to enable DDR (5.0Gb) operation, SDR (2.5 Gb) operation must also be enabled. For example, the -s2 option may yield a port which does not come up, in which case -s3 is preferred.

Caution: When using this command to disable switch ports, if the final port in the path between the Fabric Management Node and the switch is disabled, then `iba_portenable` will not be able to reenale it. In which case the switch CLI and/or a switch reboot may be needed to correct the situation.

3.2.8 iba_portenable

(Host or Switch) Enables a specified HCA port on the local host or remote switch. May also be used to change port operational parameters as part of enabling the port.



3.2.8.1 Usage

```
iba_portenable [-v] [-D] [-l lid [-m dest_port]] [-h hca] [-p port] [-w width] [-s speed] [-K mkey]
```

3.2.8.2 Options

- v – Verbose output
- D – Do not cycle port through disabled physstate
- l *lid* – Destination lid, default is local port
- m *dest_port* – Destination port, default is port with given lid useful to access switch ports
- h *hca* – HCA to send by/to, default is 1st hca
- p *port* – Port to send by/to, default is 1st port
- s *speed* – New link speeds enabled (default is 0)
 - 0 – no-op
 - 1 – 2.5 Gb/s
 - 2 – 5 Gb/s
 - 4 – 10 Gb/s

To enable multiple speeds, use the sum of desired speeds.
255 – Enable all speeds supported by given HCA port
- w *width* – New link widths enabled (default is 0)
 - 0 – no-op
 - 1 – 1x
 - 2 – 4x
 - 4 – 8x
 - 8 – 12x

To enable multiple widths, use sum of desired widths
255 – Enable all widths supported by given HCA port
- K *mkey* – SM management key to access remote ports

3.2.8.3 Example

```
iba_portenable
```

```
iba_portenable -p 2 -h 2
```

The port enablement is transient in nature. If the given host/switch is rebooted or its True Scale Fabric stack is restarted, the port will revert to its default configuration and state. Typically, the default state has the port enabled with all speeds and widths supported by the given HCA/switch port enabled.

To access switch ports using this command, the -l and -m options must be given. The -l option will specify the lid of switch port 0 (the logical management port for the switch) and -m will specify the actual switch port to access. If SMA mkeys are being used, the -K option will also be needed. By default the Intel® SM does not use SMA mkeys, in which case this option does not need to be used.



Note: On many DDR HCA and switch models, in order to enable DDR (5.0Gb) operation, SDR (2.5 Gb) operation must also be enabled. For example, the `-s2` option may yield a port which does not come up, in which case `-s3` is preferred.

3.2.9 iba_portinfo

(Host or Switch) Displays configuration and state of a specified HCA port on the local host or a remote switch.

3.2.9.1 Usage

```
iba_portinfo [-v] [-l lid [-m dest_port]] [-h hca] [-p port] [-K mkey]
```

3.2.9.2 Options

- `-v` – Verbose output
- `-l lid` – Destination lid, default is local port
- `-m dest_port` – Destination port, default is port with given lid useful to access switch ports
- `-h hca` – HCA to send by/to, default is 1st HCA
- `-p port` – Port to send by/to, default is 1st port
- `-K mkey` – SM management key to access remote ports

3.2.9.3 Example

```
iba_portinfo -p 1  
iba_portinfo -p 2 -h 2 -l 5 -m 18
```

To access switch ports using this command, the `-l` and `-m` options must be given. The `-l` option will specify the lid of switch port 0 (the logical management port for the switch) and `-m` will specify the actual switch port to access. If SMA mkeys are being used, the `-K` option will also be needed. By default the Intel® SM does not use SMA mkeys, in which case this option does not need to be used.

3.2.10 p1info, p2info

(Host) Shows the port status for port 1 or port 2 respectively. On systems with more than 1 HCA, port 1 or port 2 status on all HCAs will be displayed. Is also used to show QSFP information for HCAs.

3.2.10.1 Usage

```
p1info [-q]  
p2info [-q]
```

3.2.10.2 Options

- `-q` – Show QSFP info if available



3.2.11 p1stats, p2stats

(Host) Shows the port performance counters for port 1 or port 2 respectively. On systems with more than 1 HCA, port 1 or port 2 counters on all HCAs will be shown.

3.2.11.1 Usage

```
p1stats
```

```
p2stats
```

3.2.12 iba_portstats

(Host or Switch) Displays port performance counters of a specified HCA port on the local host or a remote switch.

3.2.12.1 Usage

```
iba_portstats [-v] [-d level] [-E] [-l lid [-m dest_port]] [-h hca] [-p port] [-s sl]
```

3.2.12.2 Options

- v – Verbose output
- d – Output more detailed debug prints
- E – Act on extended counters
- l *lid* – Destination lid, default is local port
- m *dest_port* – Destination port, default is port with given lid useful to access switch ports
- h *hca* – HCA to send by/to, default is 1st HCA
- p *port* – Port to send by/to, default is 1st port
- s *sl* – Service level to send by/to, default is SM SL

3.2.12.3 Example

```
iba_portstats -p 2 -h 2 -l 5 -m 18
```

To access switch ports using this command, the -l and -m options must be given. The -l option will specify the lid of switch port 0 (the logical management port for the switch) and -m will specify the actual switch port to access corresponding to the specified LID.

3.2.13 iba_portclear

(Host or Switch) Clears port performance counters of a specified HCA port on the local host or a remote switch.

3.2.13.1 Usage

```
iba_portclear [-v] [-d level] [-E] [-l lid [-m dest_port]] [-h hca] [-p port] [-s sl]
```



3.2.13.2 Options

- v – Verbose output
- d – Output more detailed debug prints
- E – Act on extended counters
- l *lid* – Destination lid, default is local port
- m *dest_port* – Destination port, default is port with given lid useful to access switch ports
- h *hca* – HCA to send by/to, default is 1st HCA
- p *port* – Port to send by/to, default is 1st port
- s *sl* – Service level to send by/to, default is SM SL

3.2.13.3 Example

```
iba_portclear -p 2 -h 2 -l 5 -m 18
```

To access switch ports using this command, the -l and -m options must be given. The -l option will specify the lid of switch port 0 (the logical management port for the switch) and -m will specify the actual switch port to access.

Note: On Intel® True Scale switches, a clear of the normal or extended counters will clear both 32 bit and 64 bit data movement counters (PortXmitData, PortRecvPkts, etc)

3.2.14 iba_resolve_hca_port

(Host) `iba_resolve_hca_port` permits the OFED+ style HCA number and port number arguments to be converted to an OFED+ style HCA name and physical port number. This can be useful when writing scripts that can accept FastFabric-style arguments, and interact directly with OFED commands.

3.2.14.1 Usage

```
iba_resolve_hca_port hca port
```

3.2.14.2 Options

- hca* – HCA to send via, default is 1st HCA
- port* – Port to send via, default is 1st active port

3.2.15 iba_sorthosts

The `iba_sorthosts` command will sort its stdin in a typical host name order. Hosts are stored alphabetically by any alpha numeric prefix and then sorted numerically by any numeric suffix. Leading zeros in the numeric suffix are optional. This command does not remove duplicates, any duplicates are listed in adjacent lines.

This command can be useful to build `mpi_hosts` input files for applications or cable test which places hosts in order by name.

3.2.15.1 Usage

```
iba_sorthosts < hostlist > mpi_hosts
```




or

```
iba_sorthosts --help
```

3.2.15.2 Options

--help – Produce full help text

Sort the hostlist alphabetically (case insensitively) then numerically hostnames may end in a numeric field which may optionally have leading zeros.

3.2.15.3 Example input

```
osd04
osd1
compute20
compute3
mgmt1
mgmt2
login
```

3.2.15.4 Resulting output

```
compute3
compute20
login
mgmt1
mgmt2
osd1
osd04
```

3.2.16 iba_verifyhosts

iba_verifyhosts is a tool to help perform single node verification. The actual verification is performed using `/root/hostverify.sh`. A sample of `hostverify.sh` is provided in `/opt/iba/samples/hostverify.sh` and should be reviewed and edited to set appropriate configuration and performance expectations and select which tests to run by default. See `/opt/iba/samples/hostverify.sh` and `/sbin/iba_verifyhost.sh` for more information.

Note: iba_verifyhosts now supports systems with multiple HCAs. To configure for multiple HCAs, the variables at the start of the script `hostverify.sh` located at `/opt/iba/samples/` must be edited.

3.2.16.1 Usage

```
iba_verifyhosts [-kc] [-f hostfile] [-u upload_file] [-d upload_dir] [-h 'hosts']
[-T timelimit] [test ...]
```

or



```
iba_verifyhosts --help
```

3.2.16.2 Options

--help - Produce full help text.

-k - At start and end of verification, kill any existing hostverify or xhpl jobs on the hosts

-c - Copy hostverify.sh to hosts first, useful if you have edited it

-f *hostfile* - File with hosts in cluster, default is /etc/sysconfig/iba/hosts

-h *hosts* - List of hosts to ping

-u *upload_file* - Filename to upload `hostverify.res` to after verification to allow backup and review of the detailed results for each node. The default upload destination file is `hostverify.res`. If `-u ''` is specified, no upload will occur.

-d *upload_dir* - Directory to upload result from each host. Default is `uploads`

-T *timelimit* - Timelimit (in seconds) for host to complete tests. Default is 300 seconds (5 minutes)

test - One or more specific tests to run (see `/opt/iba/samples/hostverify.sh` for a list of available tests). This verifies basic node configuration and performance by running `/root/hostverify.sh` on all specified hosts.

Prior to using this, edit `/opt/iba/samples/hostverify.sh` to set proper expectations for node configuration and performance. Then be sure to use the `-c` option on first run for a given node so that `/opt/iba/samples/hostverify.sh` gets copied to each node as `/root/hostverify.sh`.

A summary of results is appended to `FF_RESULT_DIR/verifyhosts.res`. A punchlist of failures is also appended to `FF_RESULT_DIR/punchlist.csv`. Only failures are shown on stdout.

3.2.16.3 Environment

HOSTS - List of hosts, used if `-h` option not supplied

HOSTS_FILE - File containing list of hosts, used in absence of `-f` and `-h`

UPLOADS_DIR - Directory to upload to, used in absence of `-d`

FF_MAX_PARALLEL - Maximum concurrent operations

3.2.16.4 Example

```
iba_verifyhosts -c
```

```
iba_verifyhosts -h 'arwen elrond'
```

```
HOSTS='arwen elrond' iba_verifyhosts
```



3.2.17 iba_xlat_topology

`iba_xlat_topology`, accompanied by `topology.xlsx`, `linksum_180.csv` and `linksum_360.csv` provide the capability to document the topology of a customer cluster, and generate a topology XML file based on that topology ("translate" the spread sheet to a topology file). The topology file can be used to bring up and verify the cluster.

`topology.xlsx` provides a standard format for representing each external link in a cluster. Each link contains Source, Destination, and Cable fields with one link per row of the spread sheet. The cells cannot contain commas. Source and Destination fields each have the following columns:

- Rack Group
- Rack
- Name (primary name)
- Name-2 (secondary name)
- Port number
- Port Type.

The Cable fields have the following columns:

- Label
- Length
- Details.

The Rack Group and Rack names are individually optional. If either column is completely empty it will be ignored. If the Rack Group or Rack field is empty on a particular row, the script will default the value in that field to the closest previous value (Defaulting the field to a non-empty value. The first row must have a value.). Name and Name-2 provide the name of the node which is output as the NodeDesc using the following information:

- NodeType – Name or Name-2
- Host – hostname or hostdetails
- Edge Switch – switchname
- Core Leaf – corename or Lnnn
- Core Spine – corename or Snnn (used only in internal core switch links)

For hosts Name-2 is optional and is output as NodeDetails in the topology XML file; also HCA-1 is appended to Name (see `-c` option). For core leaves (and spines) Name and Name-2 are concatenated (see `-c` option).

Port contains the port number. If the Port field is empty on a host node, the script will default to 1.

Type contains the node type. If the Type field is empty on a particular row, the script will default the value to the closest previous value (at least the first row must have a value). The type values are:

- NodeType – Type
- Host – CA
- Edge Switch – SW
- Core Leaf – CL
- Core Spine – CS (used only in internal core switch links)



Cable values are optional and have no special syntax. If the cable information is present it will appear in the topology XML file as CableLabel, CableLength and CableDetails respectively.

The `iba_xlat_topology` script reads the `topology`, `linksum_180` and `linksum_360` CSV (Comma-Separated-Values) files. The `topology.csv` file is created from the `topology.xlsx` spread sheet by saving the Fabric Links tab as a CSV file to `topology.csv`. The `topology.csv` file should be inspected to ensure that each row contains the correct and same number of comma separators. Any extraneous entries on the excel spread sheet can cause the CSV output to have extra fields.

The script produces as an output one or more topology files starting with `topology.0:0.xml`. Output at the top level as well as Group, Rack, and Switch level can be produced. Input files must be present in the same directory from which the script operates.

3.2.17.1 Usage

```
iba_xlat_topology [-d level -v level -i level -c char -K -?]
```

3.2.17.2 Options

- `-d level` – Output detail level (default 0), values are additive
 - 1 – Edge switch topology files
 - 2 – Rack topology files
 - 4 – Rack group topology files
- `-v level` – Verbose level (0-8, default 2)
 - 0 – No output
 - 1 – Progress output
 - 2 – Reserved
 - 4 – Time stamps
 - 8 – Reserved
- `-i level` – Output indent level (0-15, default 0)
- `-c char` – NodeDesc concatenation char (default SPACE)
- `-K` – Do not clean temporary files
- `-?` – Print this output

The output detail level specifies the level to which the script will produce topology XML files. By default the top level is always produced, but edge switch, rack and rack group topology files can also be produced. If the output at the group or rack level is specified, then group or rack names must be provided in the spread sheet. Detailed output can be specified in any combination. A directory for each topology XML file will be created hierarchically, with group directories (if specified) at the highest level, followed by rack and edge switch directories (if specified).

The concatenation character (`-c char`) is used when creating NodeDesc values (Name to Name-2, Name to HCA-1, and so on). A space is used by default, but another character (ex. underscore) can be specified.

The `-K` option is used to prevent temporary files (in each topology directory) from being removed. Temporary files contain lists (CSV) of links, CAs, and switches used to create a topology XML file. They are not normally needed after a topology file is created, but they are used in the creation of `linksum_180.csv` and `linksum_360.csv`, or can be retained for subsequent inspection or processing.



The `linksum_180.csv` and `linksum_360.csv` are provided as stand-alone source files. However, they can be recreated (or modified) from the spread sheet, if needed, by performing the following steps:

1. Save each of the following from the `topology.xlsx` Excel file as individual as CSV files
 - **Internal 180 Links** tab as `linksum_180.csv`
 - **Internal 360 Links** tab as `linksum_360.csv`
 - **Fabric Links** tab as `topology.csv`
2. For each saved `topology.csv` file, run the script with the `-K` option.

Upon completion of the script, save the top level `linksum.csv` file as `linksum_180.csv` or `linksum_360.csv` as appropriate.

3.2.18 iba_xlat_topology_cust

The script `iba_xlat_topology_cust` has been added, accompanied by `topology_cust.xlsx`. The script and spread sheet provide a sample alternative to the standard-format topology capability to document the topology of a customer cluster (see `iba_xlat_topology`). The alternative is provided for situations in which a customer chooses not to define a fabric topology using the standard-format spread sheet and `iba_xlat_topology.topology_cust.xlsx` provides an alternative for representing each external link in a cluster. `iba_xlat_topology_cust` translates the CSV form of the alternate spread sheet cluster tab(s) to the standard CSV form used by `iba_xlat_topology`. In using the alternative, a user would modify the sample spread sheet as needed to fit specific needs, then modify the script as needed to translate the spread sheet CSV output to the standard-format CSV output.

Like the standard format, each link contains source, destination and cable fields with one link per line (row) of the spread sheet. Link fields must not contain commas. Source and Destination fields are each a concatenation of name and port information in the following forms (N/n is a host/switch/port number; names not of the form 'ib' or 'C' are taken to be host names):

Table 2. iba_xlat_topology_cust

NodeType	Source/Destination
Host	hostN
Edge Switch	ibNpN
Core Leaf	CnLnnnpN

Cable values `CableLength` and `CableDetails` are optional and have no special syntax. If present, they are placed in the standard-format CSV file exactly as they appear. `CableLabel` is created automatically by `iba_xlat_topology_cust` as the concatenation (see `-c` option below) of Source and Destination. Rack Group and Rack are not supported in `topology_cust.xlsx`. `iba_xlat_topology_cust` leaves these fields empty in the standard-format CSV file.

3.2.18.1 Usage

```
iba_xlat_topology_cust -t topology_prime [-s topology_second] -T topology_out [-v level] [-i level] [-c char] [-K] [-?]
```

3.2.18.2 Options

`-t topology_prime` – Primary topology CSV input file



- s *topology_second* - Secondary topology CSV input file
- T *topology_out* - Topology CSV output file
- v *level* - Verbose level (0-8, default 2)
 - 0 - No output
 - 1 - Progress output
 - 2 - Reserved
 - 4 - Time stamps
 - 8 - Reserved
- i *level* - Screen output indent level (0-15, default 0)
- c *char* - Concatenation char (default SPACE)
- K - DO NOT clean temporary files
- ? - Print this output

The -t *topology_prime* option specifies the primary CSV input file and must be present. If needed -s *topology_second* can specify a secondary CSV input file. It will be appended to the primary for processing. The -T *topology_out* option specifies the CSV output file name and must be specified. The concatenation character (-c *char*) is used when creating Cable Label values. A space is used by default, but another character (e.g., underscore) can be specified. -K is used to prevent temporary files from being removed. Temporary files contain CSV data used during processing. They are not needed after the standard-format CSV file is created, but they can be retained for subsequent inspection or processing.

3.2.19 s20tune

(Host) s20tune is a daemon process which can be started on individual hosts to monitor the port state and speed of all HCAs on the given host. This process must be run on any hosts which have HCAs which do not support an IBTA compliant DDR or QDR link speed negotiation. This tool monitors for the link to stay down for 10 seconds or more and then restores the enabled speed to match the supported speed, hence restarting the speed negotiation process. For more information on Fabric Manager based link speed negotiation, see the *Intel® True Scale Fabric Suite Fabric Manager User Guide*.

3.2.19.1 Usage

```
s20tune -F -C [-v|-q] [-D] [-h hca]
```

3.2.19.2 Options

- h *hca* - HCA to monitor, default is 1st HCA
- F - Force speed. For a port which is down for 10 seconds or more, force the enabled speed to match the supported speeds for the HCA port. This helps to recover from cable pulls when doing auto negotiation through the LinkSpeedOverride option in the Intel® FM.
- C - Do no check link performance
- v - Verbose output
- q - Quiet mode
- D - Daemon (detach from terminal)



Normally, `s20tune` is run as a background process started by the `/etc/init.d/s20tune` initialization script. The `iba_config` or `./INSTALL` commands may be used to configure `s20tune` to be started automatically at system boot time.

If it is desired to manually force the link speed, `s20tune` should not be run.

Note: `s20tune` has additional arguments which are not documented above. It is recommended not to use such arguments unless directed to do so by Intel Support.

Note: `s20tune` is only required for Intel® 12000 switches with firmware older than 5.0.3. When using non-Intel® HCAs, it is recommended to use newer Switch Firmware in conjunction with the HCA Firmware supplied with Intel® True Scale Fabric OFED+ Host Software.

3.3 Basic Setup and Administration Tools

3.3.1 fastfabric

(Switch and Host) Takes the user to the top-level FastFabric text user interface (TUI) menu for setup and configuration.

3.3.1.1 Usage

```
fastfabric
```

3.3.1.2 Example

```
>fastfabric

Intel FastFabric IB Tools

Version: X.X.X.X.X

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

X) Exit
```

3.3.2 iba_config

(Switch and Host) Allows the user to configure FastFabric.

3.3.2.1 Usage

```
iba_config [-r root] [-v|-vv] [-F|-u|-s|-e comp] [-E comp] [-D comp] [--fwupdate
asneeded|always] [--user_queries|--no_user_queries] [--answer keyword=value]
```

or



```
iba_config -C
```

or

```
iba_config -V
```

3.3.2.2 Options

--help – Produce full help text

-F – Upgrade HCA Firmware with default options

--fwupdate *asneeded|always* – Select firmware update auto update mode

asneeded – Update or downgrade to match version in this release

always – Rewrite with this release version even if matches. The default is to upgrade as needed but do not downgrade. This option is ignored for interactive installs.

-u – Uninstall all ULPs and drivers with default options

-s – Enable autostart for all installed drivers

-r – Specify alternate root directory, default is /

-e *comp* – Uninstall the given component with default options. This can appear more than once on command line

-E *comp* – Enable autostart of a given component. This can appear with -D or more than once on the command line.

-D *comp* – Disable autostart of given component. This can appear with -E or more than once on command line

-v – Verbose logging.

-VV – Very verbose debug logging.

-C – Output list of supported components.

-V – Output Version.

--user_queries – Permit non-root users to query the fabric (default).

--no_user_queries – Non-root users cannot query the fabric.

--answer *keyword=value* – Provides an answer to a question that may occur during the operation. Answers to questions not asked are ignored. Invalid answers result in prompting for interactive installs, or use of the default for non-interactive.

Possible Questions:

UserQueries – Permits non-root users to query the fabric.

SinglePort – Enable Intel® HCA single-port mode. The default options retain the existing configuration files.

3.3.3 captureall

(Switch and Host) Captures supporting information for a problem report from all hosts or Intel® Chassis and uploads to this system



3.3.3.1 Usage

```
captureall [-Cp] [-f hostfile] [-F chassisfile] [-S] [-D level] [-L nodefile] [-n hosts] [file]
```

or

```
captureall --help
```

3.3.3.2 Options

- help – Produce full help text
- C – Perform capture against chassis, default is hosts
- p – Perform capture in parallel [for a host capture this only affects the upload phase]
- f *hostfile* – File with hosts in cluster, default is /etc/sysconfig/iba/hosts
- F *chassisfile* – File with chassis in cluster, default is /etc/sysconfig/iba/chassis
- S – Securely prompt for password for administrator on a chassis
- D *level* – Level of detail of the capture (only used for host captures, ignored for chassis captures)
 - 1 – Normal – Obtains local information from each host
 - 2 – Fabric – In addition to “Normal”, also obtains basic fabric information by queries to the SM and fabric error analysis using *iba_report*.
 - 3 – Fabric+FDB – In addition to “Fabric”, also obtains all the switch forwarding tables from the SM.
 - 4 – Analysis – In addition to “Fabric+FDB”, also obtains all_analysis results. If all_analysis has not yet been run, it is run as part of the capture.
- L *nodefile* – A file containing a list of the nodes in the cluster. The default file is /etc/sysconfig/iba/ibnodes.
- n *hosts* – Performs a capture against the externally-managed Intel® 12000 and/or HP BLc Intel 4X QDR InfiniBand switches. The default is *hosts*.
- file* – Name for capture file (if the filename given does not have a .tgz suffix, .tgz will be appended)

3.3.3.3 Details

When a host captureall is performed, *iba_capture* will be run to create the specified capture file within *~root* on each host (with the .tgz suffix added as needed). The files will be uploaded and unpacked into a matching directory name within *upload_dir/hostname/* on the local system. The default file name is *hostcapture*.

Note: Detail levels 2-4 can be used when fabric operational problems occur. If the problem is most likely node specific, detail level 1 should be sufficient. Detail levels 2-4 require an operational Fabric Manager. Typically your support representative will request a given detail level. If a given detail level takes excessively long or fails to be gathered, try a lower detail level.

For detail levels 2-4, the additional information is only gathered on the node running the captureall command. The information is gathered for every fabric specified in the /etc/sysconfig/iba/ports file.



When a chassis capture all is performed, the `chassis capture` CLI command will be run on each chassis and its output will be saved to `upload_dir/chassisname/file` on the local system. The default file name is `chassiscapture`.

For both host and chassis capture, the uploaded captures will be combined into a `tgz` file with the file name specified and the suffix `.all.tgz` added

3.3.3.4 Host Capture Examples

```
captureall
```

The above example creates a `hostcapture` directory in `./uploads/HOSTNAME/` for each host in `/etc/sysconfig/iba/hosts` then creates `hostcapture.all.tgz`.

```
captureall mycapture
```

The above example creates a `mycapture` directory in `./uploads/HOSTNAME/` for each host in `/etc/sysconfig/iba/hosts` then creates `mycapture.all.tgz`.

3.3.3.5 Chassis Capture Examples

```
captureall -C
```

The above example creates a `chassiscapture` file in `./uploads/CHASSISNAME/` for each chassis in `/etc/sysconfig/iba/chassis` then creates `chassiscapture.all.tgz`.

```
captureall -C mycapture
```

The above example creates a `mycapture.tgz` file in `./uploads/CHASSISNAME/` for each chassis in `/etc/sysconfig/iba/chassis` then creates `mycapture.all.tgz`.

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

When performing operations against chassis, set up of `ssh` keys is recommended (see ["setup_ssh" on page 51](#)). If `ssh` keys are not set up, all chassis must be configured with the same admin password and use of the `-S` option is recommended. The `-S` option avoids the need to keep the password in configuration files.

Note: The resulting host capture files can require significant amounts of space on the Intel® FastFabric Toolset host. Actual size will vary, but sizes can be multiple megabytes per host. As such it is recommended to ensure adequate space is available on the Intel® FastFabric Toolset system. In many cases it may not be necessary to run `captureall` against all hosts or chassis, but rather a representative subset may be sufficient. Consult with your support representative for further information.

3.3.4 pingall

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

3.3.4.1 Usage

```
pingall [-Cp] [-f hostfile] [-F chassisfile]
```

or



```
pingall --help
```

3.3.4.2 Options

```
--help - Produce full help text
-C - Performs a ping against a chassis. The default is hosts
-p - Ping all hosts/chassis in parallel
-f hostfile - File with hosts in cluster, default is /etc/sysconfig/iba/hosts
-F chassisfile - File with chassis in cluster default is
/etc/sysconfig/iba/chassis
```

3.3.4.3 Example

```
pingall
pingall -C
```

Note: This command pings all hosts/chassis found in the specified host/chassis file. The use of `-C` option merely selects the default file and/or environment variable to use. For this command it is valid to use a file which lists both hosts and chassis.

3.3.5 setup_ssh

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

3.3.5.1 Usage

```
setup_ssh [-C] [-U] [-s] [-f hostfile] [-F chassisfile] [-h 'HOSTS'] [-H 'chassis']
[-i ipoib_suffix] [-u user] [-S] [-R] [-p] [-P]
```

or

```
setup_ssh --help
```

3.3.5.2 Options

```
--help - produce full help text
-C - perform operation against chassis, default is hosts.
-U - only perform connect (to enter in local hosts, known hosts). When run in this
mode, -S and -s options are ignored).
-s - use ssh/scp to transfer files, default is rsh/rcp.
-f hostfile - file with hosts in cluster, default is /etc/sysconfig/iba/hosts.
-F chassisfile - file with chassis in cluster default is
/etc/sysconfig/iba/chassis
-h HOSTS - list of hosts to setup.
-H chassis - list of chassis to ping
-i ipoib_suffix - suffix to apply to host names to create IPoIB host names. The
default is -ib.
-u user - user on remote system to allow this user to ssh to, default is current
user code.
```



- S – securely prompt for password for user on remote system.
- R – skip setup of ssh to localhost.
- p – perform operation against all chassis or hosts in parallel.
- P – skip ping of host (for ssh to devices on internet with ping firewalled).

3.3.5.3 Example

```
setup_ssh -s -S -i ''  
setup_ssh -C
```

Intel® FastFabric Toolset provides additional flexibility in the translation between IPoIB and management network hostnames. Refer to *Configuration of IPoIB Name Mapping* in the *Intel® True Scale Fabric Suite FastFabric User Guide* for more information.

Setup_ssh provides an easy way to create ssh keys and distribute them to the hosts or chassis in the cluster. Many of the FastFabric tools (as well as many versions of MPI) require ssh be set up for password-less operation. Therefore, setup_ssh is an important setup step.

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

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

Note: The meaning of the -C option has changed, in previous releases -C selected the mode of operation (-U now serves that purpose).

3.3.5.4 Host Initial key exchange

When run without the -U option, setup_ssh will perform the initial key exchange and enable password-less ssh and scp. The key exchange can be accomplished for hosts using ssh and scp (in a password prompting manner) using the -s option or using password-less rsh and rcp (omitting the -s option).

The preferred way to use setup_ssh for initial key exchange is with the -s and -S options. This requires all hosts have been configured with the same password for the specified "user" (typically root). In this mode the password will be prompted for once and then ssh and scp are used in conjunction with that password to complete the setup for the hosts. Use in this manner also avoids the need to setup rsh/rcp/rlogin (which can be a security risk).

If -s is used without the -S option, the user will be prompted by ssh and scp for each host as they are setup. There will be multiple prompts per host. For a handful of hosts this is manageable, however for a significant number of hosts this can become cumbersome. Therefore, the -S option is recommended in this case.

If the -s option is not specified, rsh and rcp will be used to perform the ssh key exchange. This requires password-less rcp and rlogin be enabled on each host (check_rsh can perform verification).

Setup_ssh will configure password-less ssh/scp for both the management network and IPoIB. Typically, the management network will be used for Intel® FastFabric Toolset operations while IPoIB will be used for MPI and other applications. If IPoIB is not yet



running (for example, during initial cluster installation True Scale Fabric software will not yet be installed on all the hosts), the `-i` option can be specified with an empty string:

```
setup_ssh -i ''
```

This will cause the last part of the setup of ssh for IPoIB to be skipped.

3.3.5.5 Refreshing local systems known hosts

If hosts have IP addresses added (for example, by installing True Scale Fabric software and enabling IPoIB), IP addresses changed, MAC addresses changed or other aspects have changed (such as server OS reinstallation), the local hosts `ssh known_hosts` file can be refreshed by running `setup_ssh` with the `-U` option. This option will not transfer the keys, but rather will connect to each host (management network and IPoIB) in order to refresh the ssh keys. Existing entries for the specified hosts are replaced within the local `known_hosts` file. When run in this mode the `-S` and `-s` options are ignored. This mode assumes ssh has previously been setup for the hosts, as such no files are transferred to the specified hosts and no passwords should be required.

Typically after completing the installation and booting of True Scale Fabric software, `setup_ssh` will need to be rerun with the `-U` option to update the `known_hosts` file

3.3.5.6 Chassis Initial key exchange

When run without the `-U` option, `setup_ssh` will perform the initial key exchange and enable password-less ssh and scp. For chassis, the key exchange uses scp and the chassis CLI. Login to the chassis during this command will be through the configured mechanism for chassis login. Typically the `-S` option should be used when doing initial setup of ssh keys for a chassis. For chassis the `-s` option is ignored

The preferred way to use `setup_ssh` for initial key exchange is with the `-S` option. This requires all chassis have been configured with the same password for admin. In this mode the password will be prompted for once and then the `FF_CHASSIS_LOGIN_METHOD` and scp are used in conjunction with that password to complete the setup for the chassis. Use in this manner also avoids the need to setup the chassis password in `fastfabric.conf` (which can be a security risk).

For chassis the `-i` option is ignored.

3.3.5.7 Chassis Refreshing local systems known hosts

If chassis have IP addresses changed, MAC addresses changed or other aspects have changed, the local hosts `ssh known_hosts` file can be refreshed by running `setup_ssh` with the `-U` option. This option will not transfer the keys, but rather will connect to each chassis in order to refresh the ssh keys. Existing entries for the specified chassis are replaced within the local `known_hosts` file. When run in this mode the `-S` options is ignored. This mode assumes ssh has previously been setup for the chassis, as such no files are transferred to the specified hosts and no passwords should be required.

3.3.6 cmdall

(Linux and Switch) Executes a command on all hosts or Intel® Chassis. This is very powerful and can be used for everything from configuring servers or chassis, verifying that they are running, starting and stopping host processes, etc.

3.3.6.1 Usage

```
cmdall [-Cpq] [-f hostfile] [-F chassisfile] [-h 'hosts'] [-H 'chassis'] [-u user]
```



```
[-S] [-m 'marker'] [-T timelimit] [-P] 'cmd'
```

or

```
cmdall --help
```

3.3.6.2 Options

--help – Produce full help text

-C – Perform command against chassis, default is hosts

-p – Run command in parallel on all hosts

-q – Quiet mode, do not show command to execute

-f *hostfile* – File with hosts in cluster, default is /etc/sysconfig/iba/hosts

-F *chassisfile* – File with chassis in cluster default is /etc/sysconfig/iba/chassis

-u *user* – The user to perform the command as. For hosts, the default is current user code. For chassis, the default is admin (this argument is ignored)

-S – Securely prompt for password for admin on chassis

-m 'marker' – Marker for end of chassis command output if omitted defaults to chassis command prompt this may be a regular expression

-T *timelimit* – Time limit in seconds when running host commands default is -1 (infinite)

-P – Output hostname/chassis name as prefix to each output line this can make script processing of output easier

3.3.6.3 Host Examples

```
cmdall date
```

```
cmdall 'uname -a'
```

3.3.6.4 Chassis Examples

```
cmdall -C 'ismPortStats -noprompt'
```

Note:

All commands performed with `cmdall` must be non-interactive in nature. `cmdall` will wait for the command to complete before proceeding. For example, when running host commands such as `rm`, the `-i` option (interactively prompt before removal) should not be used (Note that this option is sometimes part of a standard bash alias list). Similarly, when running chassis commands such as `fwUpdateChassis`, the `-reboot` option should not be used (this option causes an immediate reboot therefore, the command never returns). Similarly, the chassis command `reboot` should not be executed using `cmdall`. Instead use the `iba chassis admin reboot` Intel® FastFabric Toolset command to reboot one or more chassis. For further information about individual chassis CLI commands consult the *Intel® True Scale Fabric Switches 12000 Series CLI Reference Guide*. For further information about Linux OS commands, consult the Linux man pages and any other documentation supplied with the OS by the OS supplier.



When performing `cmdall` against hosts, internally `ssh` is used. The command `cmdall` requires that password-less `ssh` be setup between the host running Intel® FastFabric Toolset and the hosts `cmdall` is operating against. The `setup_ssh` FastFabric tool can aid in setting up password-less `ssh`.

When performing `cmdall` against a set of chassis, all chassis must be configured with the same admin password or the `setup_ssh` FastFabric tool can be used to setup password-less `ssh` to the chassis.

When performing operations against chassis, setup of `ssh` keys is recommended (see “[setup_ssh](#)” on page 51). If `ssh` keys are not setup, use of the `-S` option is recommended. This avoids the need to keep the password in configuration files.

3.4 File Management Tools

The following tools aid in copying files to and from large groups of nodes in the fabric.

Internally, these tools make use of SCP and require that password-less SSH/SCP be setup between the host running Intel® FastFabric Toolset and the hosts files that are being transferred to and from. The `setup_ssh` FastFabric tool can aid in setting up password-less SSH/SCP.

3.4.1 `scpall`

(Linux) The `scpall` tool permits efficient copying of files or directories from the current system to multiple hosts in the fabric. When copying large directory trees, performance can be improved by using the `-t` option. This will tar and compress the tree, then transfer the resulting compressed tarball to each node (and untar it on each node).

This can provide a powerful facility for copying data files, operating system files or even applications to all the hosts (or a subset of hosts) within the fabric.

3.4.1.1 Usage

```
scpall [-p] [-r] [-f hostfile] source_file ... dest_file
```

```
scpall -t [-p] [-f hostfile] [source_dir [dest_dir]]
```

or

```
scpall --help
```

3.4.1.2 Options

`--help` – Produce full help text

`-r` – Recursive copy of directories

`-p` – Perform copy in parallel

`-t` – Optimized recursive copy of directories using tar

`-f hostfile` – File with hosts in cluster, default is `/etc/sysconfig/iba/hosts`.

`source_file` – The name of files to copy from this system, relative to the current directory. Multiple files may be listed.

`source_dir` – The name of directory to copy from this system, relative to the current directory.



dest_file or *dest_dir* - The name of the file or directory on the destination system to copy to. It is relative to the home directory of the specified user (an absolute path name may be specified if desired).

When performing directory copies using the `-t` option, the destination directory is optional. If not specified it defaults to the present directory name. If both the source and destination directory names are omitted, they both default to the current directory name.

3.4.1.3 Example

```
# copy a single file
scpall MPI-PMB /root/MPI-PMB

# efficiently copy an entire directory tree
scpall -p -t /opt/iba/src/mpi_apps /opt/iba/src/mpi_apps

# copy a group of files
scpall a b c /root/tools/
```

Note: The tool `scpall` can only copy from this system to a group of systems in the cluster. The `user@` style syntax cannot be used in the arguments to `scpall`.

To copy from hosts in the cluster to this host, use `uploadall`.

3.4.2 uploadall

(Linux) Copies one or more files from a group of hosts to this system. Since the file name will be the same on each host, a separate directory on this system is created for each host and the file is copied to it. This is a convenient way to upload log files or configuration files for review. It can also be used in conjunction with `downloadall` to upload a host specific configuration file, edit it for each host and download the new version to all the hosts.

3.4.2.1 Usage

```
uploadall [-rp] [-f hostfile] source_file ... dest_file

or

uploadall --help
```

3.4.2.2 Options

`--help` - Produce full help text

`-p` - Perform copy in parallel on all hosts

`-r` - Recursive upload of directories

`-f hostfile` - File with hosts in cluster, default is `/etc/sysconfig/iba/hosts`

source_file - The name of files to copy to this system, relative to the current directory. Multiple files may be listed.

dest_file - The name of the file or directory on this system to copy to. It is relative to `upload_dir/HOSTNAME`.



A local directory within `upload_dir/` will be created for each host being uploaded from. Each uploaded file will be copied to `upload_dir/HOSTNAME/dest_file`. If more than one source file is specified, `dest_file` will be treated as a directory name and the directories `upload_dir/HOSTNAME/dest_file/` will be created for each host and the `source_files` will be uploaded to those directories.

3.4.2.3 Example

```
# upload two files from all hosts
uploadall capture.tgz /etc/sysconfig/ifs_fm.xml

# upload network config files from all hosts
uploadall -r -p /etc/sysconfig/network-scripts network-scripts

# upload two files to a specific subdirectory of upload_dir
uploadall capture.tgz /etc/sysconfig/ifs_fm.xml pre-install
```

The above example copies `capture.tgz` and `/etc/sysconfig/ifs_fm.xml` to `./uploads/HOSTNAME/preinstall/` where a `HOSTNAME` directory is created for each host in `/etc/sysconfig/iba/hosts`.

Note: The `uploadall` tool can only copy from a group of systems in a cluster to this system. The `user@` style syntax cannot be used in the arguments to `uploadall`.

To copy files from this host to hosts in the cluster use `scpall` or `downloadall`.

3.4.3 downloadall

(Linux) Copies one or more files to a group of hosts from a system. Since the file contents to copy may be different for each host, a separate directory on this system is used for the source files for each host. This can also be used in conjunction with `uploadall` to upload a host-specific configuration file, edit it for each host and download the new version to all the hosts.

3.4.3.1 Usage

```
downloadall [-rp] [-f hostfile] [-d download_dir] source_file ... dest_file

or

downloadall --help
```

3.4.3.2 Options

- `--help` - Produce full help text
- `-p` - Perform copy in parallel on all hosts
- `-r` - Recursive download of directories
- `-f hostfile` - File with hosts in cluster. The default is `/etc/sysconfig/iba/hosts`.
- `-d download_dir` - The directory to download files to. The default is `downloads`. If not specified, the environment variable `DOWNLOADS_DIR` will be used. If that is not exported the default will be used.



source_file – The name of files to copy from the system. Multiple files may be listed. The option *source_file* is relative to *download_dir/HOSTNAME*.

A local directory within *download_dir/* must exist for each host being downloaded to each downloaded file will be copied from *download_dir/HOSTNAME/source_file*.

dest_file – The name of the file or directory on the destination hosts to copy to.

If more than one source file is specified, *dest_file* will be treated as a directory name. The given directory must already exist on the destination hosts (the copy will fail for hosts where the directory does not exist).

3.4.3.3 Example

```
# copy two files to all hosts
downloadall ics_srp.cfg ics_inic.cfg /etc/sysconfig
```

Note: The tool `downloadall` can only copy from this system to a group of hosts in the cluster. The `user@` style syntax cannot be used in the arguments to `downloadall`.

To copy files from hosts in the cluster to this host use `uploadall`.

3.4.4 Simplified Editing of Node-Specific Files

(Linux) The combination of `uploadall` and `downloadall` provide a powerful yet simple to use mechanism for reviewing and/or editing node-specific files without the need to login to each node.

This is best explained with an example.

Assume the file `/etc/sysconfig/network-scripts/ifcfg-ib1` needs to be reviewed and possibly edited for each host. This file would typically contain the IP configuration information for IPoIB and may contain a unique IP address per host.

To upload the file from all the hosts:

```
uploadall /etc/sysconfig/network-scripts/ifcfg-ib1 ifcfg-ib1
```

Edit the uploaded files with an editor, such as `vi`:

```
vi uploads/*/ifcfg-ib1
```

If by using the editor, the file was changed for some or all of the hosts, it can then be downloaded to all the hosts:

```
downloadall -d uploads ifcfg-ib1 /etc/sysconfig/network-scripts/ifcfg-ib1
```

3.4.5 Simplified Setup of Node-Generic Files

(Linux) In contrast `scpall` can provide a powerful yet simple to use mechanism for transferring files to all nodes that are generic (for example, not node-specific).

For example, if all nodes in the cluster will use the same DNS server and TCP/IP name resolution, they may be quickly set as follows:

Create an appropriate local file with the desired information. For example:

```
vi resolv.conf
```

Copy the file to all hosts:



```
scpall resolv.conf /etc/resolv.conf
```

3.5 Fabric Analysis Tools

3.5.1 fabric_info

`fabric_info` provides a brief summary of the components in the fabric. `fabric_info` uses the first active port on the given local host to perform its analysis. `fabric_info` is supplied as part of both Intel® FastFabric Toolset and the Intel® True Scale Fabric Tools.

3.5.1.1 Usage

```
fabric_info
```

or

```
fabric_info --help
```

3.5.1.2 Options

--help – produce full help text

3.5.1.3 Example

```
fabric_info
```

3.5.1.4 Example output

```
# fabric_info

Fabric Information:

SM: i9k229 Guid: 0x00066a00d8000229 State: Master

SM: i9k3ff Guid: 0x00066a00d90003ff State: Standby

Number of CAs: 17

Number of CA Ports: 22

Number of Switch Chips: 6

Number of Links: 29

Number of 1x Ports: 2
```

3.5.1.5 Output Definitions

SM – each subnet manger (SM) running in the fabric is listed along with its node name, port GUID and present SM state (Master, Standby, etc).

Number of CAs – number of unique channel adapters (CA) in the fabric. A CA with two-connected ports is counted as a single CA.

Note: Channel adapters include both HCAs in servers as well as TCAs within IO Modules, Native Storage, etc.

Number of CA Ports – number of connected CA ports in the fabric.



Number of Switch Chips – number of unique switches in the fabric.

Note: A large switch may be composed of many unique switch chips.

Number of Links – number of links in the fabric. Note that a large switch may have internal links.

Number of 1x Ports – number of ports in the fabric running at 1x speed. Typically such ports represent a bad cable connection, a bad cable, too long a cable or perhaps faulty hardware on one side of the link.

`fabric_info` can be very useful as a quick assessment of the fabric state. `fabric_info` can be run against a known good fabric to identify its components and then later run to see if anything has changed about the fabric configuration or state. When used in this manner it can be used to quickly identify if CAs are down, links are missing, SMs are missing, etc.

For more extensive fabric analysis, see [“iba_report” on page 61](#) and [“iba_reports” on page 212](#). Also see `iba_top` in the *Intel® True Scale Fabric Suite FastFabric User Guide*.

3.5.2 showallports

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

Note: `iba_report` and `iba_reports` are newer and more powerful Intel® FastFabric Toolset commands. For general fabric analysis, use `iba_report` or `iba_reports` with options such as `-o errors` and/or `-o slowlinks` to perform a more efficient analysis of link speeds and errors.

3.5.2.1 Usage

```
showallports [-C|-I] [-f hostfile] [-F chassisfile] [-L ibnodefile] [-S]
```

or

```
showallports --help
```

3.5.2.2 Options

`--help` – Produce full help text

`-C` – Perform operation against chassis; the default is `hosts`

`-I` – Perform operation against nodes; the default is `hosts`

`-f hostfile` – File with hosts in cluster; the default is `/etc/sysconfig/iba/hosts`

`-F chassisfile` – File with chassis in cluster; the default is `/etc/sysconfig/iba/chassis`

`-L ibnodefile` – File with nodes in the cluster; the default is `/etc/sysconfig/iba/ibnodes`

`-S` – Securely prompt for password for administrator on chassis



3.5.2.3 Example

```
showallports
showallports -C
showallports -I
```

When performing `showallports` against hosts, internally SSH is used. `showallports` requires that password-less SSH be setup between the host running Intel® FastFabric Toolset and the hosts `showallports` is operating against. The `setup_ssh` FastFabric tool can aid in setting up password-less SSH.

When performing operations against chassis, setup of ssh keys is recommended (see [“setup_ssh” on page 51](#)). If ssh keys are not setup, all chassis must be configured with the same admin password and use of the `-S` option is recommended. The `-S` option avoids the need to keep the password in configuration files.

When performing `showallports` against externally-managed switches it requires a Fabric Management Node with Intel® FastFabric Toolset installed. Typically this will be the node from which `showallports` is being run. If desired, an alternate node may be specified by the `-M` option or `MGMT_HOST` environment variable.

3.5.3 iba_report

(All) `iba_report` provides powerful fabric analysis and reporting capabilities. It must be run on a host connected to the True Scale Fabric with Intel® FastFabric Toolset installed.

`iba_report` obtains all its data in an IBTA-compliant manner. Therefore, it will interoperate with both Intel® and 3rd party components, provided those components are IBTA compliant and implement the IBTA optional features required by `iba_report`.

`iba_report` requires that the subnet manager implement all the IBTA SA queries defined in the standard (such as SM Info records, Link Records, Trace Routes, Port Records, Node Records, etc). As such, it is recommended that the Intel® True Scale Fabric Suite Fabric Manager (FM) version 4.0 or later be used. `iba_report` requires all end nodes to implement the PMA PortCounters (IBTA mandatory counters). Also any end nodes which report support of a IBTA device management agent must implement the IOU Info, IOC Profile and Service Entry queries as outlined in the IBTA 1.1 standard.

`iba_report` also supports operation with the FM PM/PA. When `iba_report` detects the presence of a PA, it automatically issues any required PortCounter queries and clears to the PA to access the PMs running totals. If a PA is not detected, then `iba_report` will directly access the PMAs on all the nodes. The `-M` option (refer to [“iba_report” on page 175](#)) can force access to the PMA even if a PA is present.

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

`iba_report` obtains and displays 64-bit data movement counters from the FM PM/PA or directly from the fabric using the `-M` option (refer to [“iba_report” on page 175](#)). Snapshot's generated by this version of `iba_report` in conjunction with the `-s` option, may report value out of range errors when used as input to older versions of



`iba_report`. However, the thresholds specified in `iba_mon.conf` and other input config files continue to only support 32-bit values for data movement counter thresholds.

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

Each run of `iba_report` obtains up to date information from the fabric. At the start of the run `iba_report` will take a few seconds to obtain all the fabric data, then it will output it to `stdout`. The reports are sorted by GUIDs and other permanent information such that they can be rerun in the future and produce output in the same order even if components have been rebooted. This is useful for comparison using simple tools like `diff`. `iba_report` permits multiple reports to be requested for a single run (for example, 1 of each report type).

3.5.3.1 Usage

```
iba_report [-v][-q] [-o report] [-d level] [-x] [-s] [-i seconds] [-C]
```

or

```
iba_report --help
```

3.5.3.2 Options

- `--help` - Produce full help text
- `-v/--verbose` - Verbose output
- `-q/--quiet` - Disable progress reports
- `-o/--output report` - Report type for output. Refer to ["Report Types \(abridged\)"](#).
- `-d/--detail level` - Level of detail 0-n for output, default is 2
- `-x/--xml` - Output in XML
- `-s/--stats` - Get performance statistics for all ports
- `-i/--interval seconds` - Obtain performance statistics over interval seconds, clears all statistics, waits interval seconds, then generates report. Implies `-s`
- `-C/--clear` - Clear performance stats for all ports. Only stats with error thresholds are cleared. A clear occurs after generating the report.

3.5.3.3 Report Types (abridged)

- `comps` - Summary of all systems and SMs in fabric
- `brcomps` - Brief summary of all systems and SMs in fabric
- `nodes` - Summary of all node types and SMs in fabric
- `brnodes` - Brief summary of all node types and SMs in fabric



ious – Summary of all IO units in the fabric

lids – Summary of all LIDs in fabric

links – Summary of all links

extlinks – Summary of links external to systems

slowlinks – Summary of links running slower than expected

slowconfiglinks – Summary of links configured to run slower than supported, includes slowlinks

slowconnnlinks – Summary of links connected with mismatched speed potential, includes slowconfiglinks

misconfiglinks – Summary of links configured to run slower than supported

misconnnlinks – Summary of links connected with mismatched speed potential

errors – Summary of links whose errors exceed counts in the configuration file

otherports – Summary of ports not connected to the fabric

all – Comp, nodes, ious, links, extlinks, slowconnnlinks, and errors reports

none – No report, useful if just want to clear statistics

3.5.3.4 Examples

iba_report can generate hundreds of different reports. Following is a list of some commonly generated reports:

Analyze a fabric for bad cables:

```
iba_report -o slowlinks -o errors
```

Analyze a fabric for bad cables or misconfigured ports:

```
iba_report -o slowconfiglinks -o errors
```

Analyze a fabric for bad cables or misconfigured ports or misconnected ports:

```
iba_report -o slowconnnlinks -o errors
```

Clear all the port counters in the fabric:

```
iba_report -C -o none
```

Check all port counters, clear them, then recheck:

```
iba_report -o errors -C; sleep 10; iba_report -o errors
```

Clear all port counters, wait 10 seconds, then check

```
Iba_report -i 10 -o errors
```

3.5.3.5 Basics of Using iba_report

iba_report can be run with no options at all. In this mode it provides a brief list of the nodes in the fabric (the brnodes report). The report organizes nodes as CAs, Switches and Routers. It also includes a summary of all the SMs in the fabric.



The following is a sample of an iba_report for a small fabric:

```
[root@duster root]# iba_report

Node Type Brief Summary

14 Connected CAs in Fabric:

NodeGUID          Type Name
-----
Port LID  PortGUID          Width Speed
-----
0x0002c9020020e0d4 CA coyote1
      1 0x000d 0x0002c9020020e0d5  4x  2.5Gb
0x00066a00580001e0 CA VEx in Chassis 0x00066a005000010c, Slot 2
      2 0x0014 0x00066a02580001e0  4x  2.5Gb
0x00066a0098000001 CA julio
      1 0x000c 0x00066a00a0000001  4x  2.5Gb
0x00066a00980001b8 CA orc
      1 0x000b 0x00066a00a00001b8  4x  2.5Gb
0x00066a0098000380 CA goblin
      1 0x000a 0x00066a00a0000380  4x  2.5Gb
0x00066a0098000384 CA cuda
      1 0x0005 0x00066a00a0000384  1x  2.5Gb
      2 0x0006 0x00066a01a0000384  4x  2.5Gb
0x00066a00980003a6 CA erik
      1 0x0015 0x00066a00a00003a6  4x  2.5Gb
      2 0x0016 0x00066a01a00003a6  4x  2.5Gb
0x00066a00980006a2 CA goblin
      1 0x000f 0x00066a00a00006a2  4x  2.5Gb
0x00066a0098000849 CA rockaway
      2 0x000e 0x00066a01a0000849  4x  2.5Gb
0x00066a0098002813 CA brady
      1 0x0002 0x00066a00a0002813  4x  2.5Gb
      2 0x0003 0x00066a01a0002813  4x  2.5Gb
0x00066a0098002854 CA brady
      1 0x0004 0x00066a00a0002854  4x  2.5Gb
      2 0x0008 0x00066a01a0002854  4x  2.5Gb
0x00066a0098003f81 CA ibm345
      1 0x0007 0x00066a00a0003f81  4x  2.5Gb
```




0x00066a009800447b CA duster

```
1 0x0011 0x00066a00a000447b 4x 2.5Gb
```

2 0x0012 0x00066a01a000447b 4x 2.5Gb

0x00066a0098004a73 CA erik

```
1 0x0009 0x00066a00a0004a73 4x 2.5Gb
```

3 Connected Switches in Fabric:

NodeGUID	Type	Name
----------	------	------

Port	LID	PortGUID	Width	Speed
------	-----	----------	-------	-------

```
0x00066a00280002cd SW InfiniCon Systems InfiniFabric (Sw A Dev A)
```

```
0 0x0013 0x00066a00280002cd Noop      Noop
```

3 4x 2.5Gb

5	4x	2.5Gb
---	----	-------

```
0x00066a00d8000123 SW InfiniCon Systems InfinIO9024
```

```
0 0x0001 0x00066a00d8000123 4x 2.5Gb
```

1	4x	2.5Gb
---	----	-------

2 1x 2.5Gb

3 4x 2.5Gb

4 4x 2.5Gb

5	4x	2.5Gb
---	----	-------

6 4x 2.5Gb

7	4x	2.5Gb
---	----	-------

8	4x	2.5Gb
---	----	-------

9	4x	2.5Gb
---	----	-------

10	4x	2.5Gb
----	----	-------

11	4x	2.5Gb
----	----	-------

12	4x	2.5Gb
----	----	-------

14 4x 2.5Gb

15	4x	2.5Gb
----	----	-------

16 4x 2.5Gb

17 4x 2.5Gb

18	4x	2.5Gb
----	----	-------

19 4x 2.5Gb

20 4x 2.5Gb



```
0x00066a10280002cd SW InfiniCon Systems InfiniFabric (Sw A Dev B)
```

```
0 0x0010 0x00066a10280002cd Noop      Noop
2                                     4x    2.5Gb
4                                     4x    2.5Gb
```

```
1 Connected SMs in Fabric:
```

State	GUID	Name
Master	0x00066a00d8000123	InfiniCon Systems InfinIO9024

Each `iba_report` allows for various levels of detail. Increasing detail is shown as further indentation of the additional information. The `-d` option to `iba_report` controls the detail level. The default is 2. Values from 0-n are permitted. The maximum detail per report varies, but most have less than five detail levels.

For example, the above report when run at detail level 0 outputs:

```
[root@duster root]# iba_report -d 0
```

```
Node Type Brief Summary
```

```
14 Connected CAs in Fabric:
```

```
3 Connected Switches in Fabric:
```

```
1 Connected SMs in Fabric:
```

The following is a summary of fabric components and is very similar to `fabric_info`. At the next level of detail, the report has more detail:

```
[root@duster root]# iba_report -d 1
```

```
Node Type Brief Summary
```

```
14 Connected CAs in Fabric:
```

NodeGUID	Type	Name
0x0002c9020020e0d4	CA	coyote1
0x00066a00580001e0	CA	VEx in Chassis 0x00066a005000010c, Slot 2
0x00066a0098000001	CA	julio
0x00066a00980001b8	CA	orc
0x00066a0098000380	CA	goblin
0x00066a0098000384	CA	cuda
0x00066a00980003a6	CA	erik
0x00066a00980006a2	CA	goblin
0x00066a0098000849	CA	rockaway
0x00066a0098002813	CA	brady
0x00066a0098002854	CA	brady



```
0x00066a0098003f81 CA ibm345
```

```
0x00066a009800447b CA duster
```

```
0x00066a0098004a73 CA erik
```

```
3 Connected Switches in Fabric:
```

NodeGUID	Type	Name
0x00066a00280002cd	SW InfiniCon Systems	InfiniFabric (Sw A Dev A)
0x00066a00d8000123	SW InfiniCon Systems	InfinIO9024
0x00066a10280002cd	SW InfiniCon Systems	InfiniFabric (Sw A Dev B)

```
1 Connected SMS in Fabric:
```

State	GUID	Name
Master	0x00066a00d8000123	InfiniCon Systems InfinIO9024

The previous examples were all performed with a single report, the brnodes (Brief Nodes) report. However this is just one of the many topology reports which `iba_report` can generate. The others include:

- `nodes` – a more verbose form of brnode which can provide much greater levels of detail to drill down into all the details of every node, even down to all the port state, IOUs/IOCs/Services, Port counters.
- `comps` and `brcomps` are very similar to brnodes and nodes, except the reports are organized around systems. The grouping into systems is based on system image guides for each node. This report will help to present more complex systems (such as servers with multiple HCAs or large switches composed of multiple switch chips).

Note:

All Intel® Switches implement a system image GUID and will therefore be properly grouped. However, some third-party devices do not implement the system image GUID and may report a value of 0. In such a case `iba_report` will treat each component as an independent system.

- `links` – This report presents all the links in the fabric. The output is very concise and helps to identify the connectivity between nodes in the fabric. This includes both internal (inside a large switch or system) and external ports (cables).
- `extlinks` – All the external links in the fabric (for example, those between different systems). This omits links internal to a single system. Identification of a system is through `SystemImageGuid`.
- `ious` – This is somewhat similar to the nodes reports, however the focus is around IOUs/IOCs and IO Services in the fabric. This report can be used to identify various IO devices in the fabric and their capabilities (such as the IBTA compliant direct-attach storage).
- `otherports` – All the ports which are not connected to this fabric. This report will identify additional ports on CAs or Switches which are not connected to this fabric. For switches these represent unused ports. For CAs these may be ports connected to other fabrics or unused ports.

The above reports are all summaries of the present state of the fabric. These reports can be very helpful to analyze the configuration of the fabric and or verify it was installed consistent with the desired design and configuration.



The `iba_report` does not stop there. Additionally, `iba_report` has reports that will help to analyze the operational characteristics of the fabric and help to identify bottlenecks and faulty components in the fabric.

To assist in this area, the `iba_report` also supports the following reports:

- `slowlinks` – identifies links which are running slower than expected. This helps to pinpoint bad cables or components in the fabric, such as a 4x cable that is poorly-connected and therefore only runs at 1x link width. The analysis includes both link speed and width.
- `slowconfiglinks` – this extends on the `slowlinks` report to also report links which have been configured (most likely by software) to run at a width or speed below their potential. Such as DDR capable links which have been forced to run at SDR rates.
- `slowconnl links` – this further extends on the `slowconfiglinks` report to also report links which are cabled such that one of the ends of the link will never run to its potential. Such as a DDR capable HCA connected to an SDR switch.
- `misconfiglinks` – this is similar to `slowconfiglinks` in that it reports links which have been configured to run below their potential. However it does not include links which are running slower than expected.
- `misconnl links` – this is similar to `slowconnl links` in that it reports links which have been connected between ports of different speed potential. However it does not include links which are running slower than expected, nor links which have been configured to run slower than their potential.
- `errors` – this performs a single point in time analysis of the PMA port counters for every node and port in the fabric. All the counters are compared against configured thresholds (defaults are those in the `iba_mon.conf` file). Any link whose counters exceed these thresholds are listed (and depending on the detail level the exact counter and threshold will be reported). This is a powerful way to identify marginal links in the fabric such as bad or loose cables or damaged components. The `iba_mon.si.conf` file can also be used to check for any non-zero values for signal integrity (SI) counters.

The above set of reports can therefore be very powerful ways to obtain point in time status and problem analysis for the fabric.

3.5.3.6 Scriptable output

The `iba_report` permits custom scripting. The `-x` option permits output reports to be generated in XML format. The XML hierarchy is similar to the textual reports. Use of XML permits other XML tools (such as PERL XML extensions) to easily parse `iba_report` output such that scripts can be created to further search and refine report output formats.

The `xml_extract` FastFabric tool can easily convert between XML files and delimited text files. See the [“Converting iba_report output to excel importable files - xml_extract” on page 77](#) for more information.

This allows `iba_report` to be integrated into custom scripts. It can also be used to generate customer-specific new report formats, cross reference `iba_report` with other site-specific information, etc.

3.5.3.7 Sample Output

3.5.3.7.1 Analysis of all ports in fabric for errors, inconsistent connections, bad cables

```
[root@duster root]# iba_report -o errors -o slowconnl links
```



Links running slower than faster port Summary

Links running slower than expected:

20 of 20 Links Checked, 0 Errors found

Links configured to run slower than supported:

Rate	MTU	NodeGUID	Port	Type	Name
2.5g	2048	0x00066a0098000384	1	CA	cuda
1x	2.5Gb	1-4x	2.5Gb		
<-> 0x00066a00d8000123 2 SW InfiniCon Systems InfinIO9024					
1-4x	2.5Gb	1-4x	2.5Gb		

20 of 20 Links Checked, 1 Errors found

Links connected with mismatched speed potential:

20 of 20 Links Checked, 0 Errors found

Links with errors > threshold Summary

Configured Error Thresholds:

SymbolErrorCounter	100
LinkErrorRecoveryCounter	3
LinkDownedCounter	3
PortRcvErrors	100
PortRcvRemotePhysicalErrors	100
PortXmitDiscards	100
PortXmitConstraintErrors	10
PortRcvConstraintErrors	10
LocalLinkIntegrityErrors	3
ExcessiveBufferOverrunErrors	3
VL15Dropped	100

Rate	MTU	NodeGUID	Port	Type	Name
10g	2048	0x00066a0098000001	1	CA	julio
<-> 0x00066a00d8000123 8 SW InfiniCon Systems InfinIO9024					



```
LinkDownedCounter: 5 Exceeds Threshold: 3

10g 2048 0x00066a00980001b8    1 CA orc
<->      0x00066a00d8000123   10 SW InfiniCon Systems InfinIO9024
LinkDownedCounter: 5 Exceeds Threshold: 3

10g 2048 0x00066a0098000380    1 CA goblin
SymbolErrorCounter: 65535 Exceeds Threshold: 100
LinkErrorRecoveryCounter: 255 Exceeds Threshold: 3
PortRcvErrors: 65535 Exceeds Threshold: 100
<->      0x00066a00d8000123   15 SW InfiniCon Systems InfinIO9024
SymbolErrorCounter: 41079 Exceeds Threshold: 100
LinkErrorRecoveryCounter: 188 Exceeds Threshold: 3

10g 2048 0x00066a00980003f81    1 CA ibm345
<->      0x00066a00d8000123   12 SW InfiniCon Systems InfinIO9024
SymbolErrorCounter: 9533 Exceeds Threshold: 100
LinkErrorRecoveryCounter: 46 Exceeds Threshold: 3
PortRcvErrors: 617 Exceeds Threshold: 100

20 of 20 Links Checked, 4 Errors found
```

3.5.3.7.2 Obtain very detailed information about nodes

Note: To shorten the length of the output, the following example focuses on only 1 node.

```
[root@duster root]# iba_report -o nodes -F node:erik -d 5 -s

Node Type Summary Focused on:

System: 0x00066a0098004a73

Node: 0x00066a00980003a6 CA erik
Node: 0x00066a0098004a73 CA erik

13 Connected CAs in Fabric:

Name: erik

NodeGUID: 0x00066a00980003a6 Type: CA

Ports: 2 PartitionCap: 64 SystemImageGuid: 0x00066a0098004a73
```



```

BaseVer: 1 SmaVer: 1 VendorID: 0x66a DeviceID: 0x5a44 Rev: 0xa1
2 Connected Ports:
  PortNum: 1 LID: 0x0015 GUID: 0x00066a00a00003a6
    Neighbor: 0x00066a00d8000123 9 SW InfiniCon Systems InfinIO9024
    PortState: Active PhysState: LinkUp DownDefault: Pollg
    LID: 0x0015 LMC: 0 Subnet: 0xfe8000000000000
    SMLID: 0x0001 SMSL: 0 RespTimeout: 33 ms SubnetTimeout: 6 ms
    M_KEY: 0x0000000000000000 Lease: 0 s Protect: Readonly
    MTU: Active: 2048 Supported: 2048 VL Stall: 0
    LinkWidth: Active: 4x Supported: 1-4x Enabled: 1-4x
    LinkSpeed: Active: 2.5Gb Supported: 2.5Gb Enabled: 2.5Gb
    VLs: Active: 4+1 Supported: 4+1 HOQLife: 4 us
    Capability 0x02010048: CR CM SL Trap
    Violations: M_Key: 0 P_Key: 0 Q_Key: 0
    ErrorLimits: Overrun: 15 LocalPhys: 15 DiagCode: 0x0000
    P_Key Enforcement: In: Off Out: Off FilterRaw: In: Off Out: Off
    Performance: Transmit
      Xmit Data 0 MiB (0 Quads)
      Xmit Pkts 0
    Performance: Receive
      Rcv Data 0 MiB (0 Quads)
      Rcv Pkts 0
    Errors:
      Symbol Errors 0
      Link Error Recovery 0
      Link Downed 0
      Port Rcv Errors 0
      Port Rcv Rmt Phys Err 0
      Port Rcv Sw Relay Err 0
      Port Xmit Discards 0
      Port Xmit Constraint 0
      Port Rcv Constraint 0
      Local Link Integrity 0
      Exc. Buffer Overrun 0

```



```
VL15 Dropped                                0
PortNum:   2 LID: 0x0016 GUID: 0x00066a01a00003a6
Neighbor:  0x00066a00d8000123   7 SW InfiniCon Systems InfinIO9024
PortState: Active                PhysState: LinkUp   DownDefault: Pollg
LID:       0x0016                LMC: 0            Subnet: 0xfe800000000000
SMLID:     0x0001   SMLS: 0      RespTimeout:   33 ms  SubnetTimeout: 6 ms
M_KEY:     0x0000000000000000   Lease:    0 s      Protect: Readonly
MTU:       Active:    2048   Supported:    2048   VL Stall: 0
LinkWidth: Active:     4x   Supported:    1-4x   Enabled:    1-4x
LinkSpeed: Active:    2.5Gb   Supported:    2.5Gb   Enabled:    2.5Gb
VLs:       Active:     4+1   Supported:     4+1   HOQLife: 4096 ns
Capability 0x02010048: CR CM SL Trap
Violations: M_Key:    0 P_Key:    0 Q_Key:    0
ErrorLimits: Overrun: 15 LocalPhys: 15 DiagCode: 0x0000
P_Key Enforcement: In: Off Out: Off FilterRaw: In: Off Out: Off
Performance: Transmit
Xmit Data                                0 MiB (0 Quads)
Xmit Pkts                                0
Performance: Receive
Rcv Data                                0 MiB (0 Quads)
Rcv Pkts                                0
Errors:
Symbol Errors                            0
Link Error Recovery                       0
Link Downed                              0
Port Rcv Errors                           0
Port Rcv Rmt Phys Err                     0
Port Rcv Sw Relay Err                     0
Port Xmit Discards                         0
Port Xmit Constraint                       0
Port Rcv Constraint                       0
Local Link Integrity                      0
Exc. Buffer Overrun                       0
VL15 Dropped                              0
```




Name: erik

NodeGUID: 0x00066a0098004a73 Type: CA

Ports: 2 PartitionCap: 64 SystemImageGuid: 0x00066a0098004a73

BaseVer: 1 SmaVer: 1 VendorID: 0x66a DeviceID: 0x5a44 Rev: 0xa1

1 Connected Ports:

PortNum: 1 LID: 0x0009 GUID: 0x00066a00a0004a73

Neighbor: 0x00066a00d8000123 18 SW InfiniCon Systems InfinIO9024

PortState: Active PhysState: LinkUp DownDefault: Pollg

LID: 0x0009 LMC: 0 Subnet: 0xfe800000000000

SMLID: 0x0001 SMSL: 0 RespTimeout: 33 ms SubnetTimeout: 6 ms

M_KEY: 0x0000000000000000 Lease: 0 s Protect: Readonly

MTU: Active: 2048 Supported: 2048 VL Stall: 0

LinkWidth: Active: 4x Supported: 1-4x Enabled: 1-4x

LinkSpeed: Active: 2.5Gb Supported: 2.5Gb Enabled: 2.5Gb

VLs: Active: 4+1 Supported: 4+1 HOQLife: 4096 ns

Capability 0x02010048: CR CM SL Trap

Violations: M_Key: 0 P_Key: 0 Q_Key: 0

ErrorLimits: Overrun: 15 LocalPhys: 15 DiagCode: 0x0000

P_Key Enforcement: In: Off Out: Off FilterRaw: In: Off Out: Off

Performance: Transmit

Xmit Data 17 MiB (4508856 Quads)

Xmit Pkts 62623

Performance: Receive

Rcv Data 0 MB (238320 Quads)

Rcv Pkts 3310

Errors:

Symbol Errors 0

Link Error Recovery 0

Link Downed 0

Port Rcv Errors 0

Port Rcv Rmt Phys Err 0

Port Rcv Sw Relay Err 0

Port Xmit Discards 0

Port Xmit Constraint 0



```
Port Rcv Constraint      0
Local Link Integrity     0
Exc. Buffer Overrun      0
VL15 Dropped            0

2 Matching CAs Found
```

```
3 Connected Switches in Fabric:
0 Matching Switches Found
```

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

3.5.3.7.3 Obtain very detailed information about IOUs

Note: To shorten the length of the output, the following example focuses on only 1 IOC.

```
[root@duster root]# iba_report -o ious -F ioc:'Chassis 0x00066A005000010C, Slot 2,
IOC 2' -d 5
```

IOU Summary Focused on:

```
Ioc: 2 0x00066a02300001e0 Chassis 0x00066A005000010C, Slot 2, IOC 2
in Node: 0x00066a00580001e0 CA VEx in Chassis 0x00066a005000010c, Slot
```

1 IOUs in Fabric:

```
Name: VEx in Chassis 0x00066a005000010c, Slot 2
NodeGUID: 0x00066a00580001e0 Type: CA
Ports: 2 PartitionCap: 64 SystemImageGuid: 0x00066a00580001e0
BaseVer: 1 SmaVer: 1 VendorID: 0x66a DeviceID: 0x5a44 Rev: 0xa1
```

1 Connected Ports:

```
PortNum: 2 LID: 0x0013 GUID: 0x00066a02580001e0
Neighbor: 0x00066a00280002cd 3 SW InfiniCon Systems InfiniFabric
```

(Sw A Dev A)

```
PortState: Active      PhysState: LinkUp      DownDefault: Pollig
LID: 0x0013            LMC: 0                Subnet: 0xfe80000000000000
SMLID: 0x0001 SMSL: 0  RespTimeout: 33 ms SubnetTimeout: 56 ms
M_KEY: 0x0000000000000000 Lease: 0 s      Protect: Readonly
MTU: Active: 2048 Supported: 2048 VL Stall: 0
```



```

LinkWidth: Active:      4x Supported:    1-4x Enabled:    1-4x
LinkSpeed: Active:     2.5Gb Supported:  2.5Gb Enabled:    2.5Gb
VLs:      Active:      1+1 Supported:    4+1 HOQLife: 4096 ns
Capability 0x02090048: CR DM CM SL Trap
Violations: M_Key:      0 P_Key:        0 Q_Key:        0
ErrorLimits: Overrun: 15 LocalPhys: 15 DiagCode: 0x0000
P_Key Enforcement: In: Off Out: Off FilterRaw: In: Off Out: Off
Max IOCs:    3 Change ID:      9 DiagDeviceId: 0 Rom: 0
IocSlot:     2 GUID: 0x00066a02300001e0
ID String: Chassis 0x00066A005000010C, Slot 2, IOC 2
IO Class: 2000 SubClass: 66a Protocol: 0 Protocol Ver: 1
VendorID: 0x66a DeviceID: 0x30 Rev: 0x1
Subsystem: VendorID: 0x66a DeviceID: 0x30
Capability: 0x33: ST SF WT WF
Send Depth: 2 Size: 256; RDMA Read Depth: 0 RDMA Size: 4294967295
2 Services:
Name: InfiniNIC.InfiniConSys.Control:02
Id: 0x1000066a00000002
Name: InfiniNIC.InfiniConSys.Data:02
Id: 0x1000066a00000102

```

1 Matching IOUs Found

3.5.3.7.4 Identify connections and links composing the fabric

```
[root@duster root]# iba_report -o links
```

Link Summary

20 Links in Fabric:

Rate	MTU	NodeGUID	Port	Type	Name
10g	2048	0x00066a00280002cd	3	SW	InfiniCon Systems InfiniFabric (Sw A Dev A)
<->		0x00066a00580001e0	2	CA	VEx in Chassis 0x00066a005000010c, Slot 2
10g	2048	0x00066a00280002cd	5	SW	InfiniCon Systems InfiniFabric (Sw A Dev A)
<->		0x00066a10280002cd	4	SW	InfiniCon Systems InfiniFabric (Sw A Dev B)
10g	2048	0x00066a0098000001	1	CA	julio
<->		0x00066a00d8000123	8	SW	InfiniCon Systems InfinIO9024



```
10g 2048 0x00066a00980001b8 1 CA orc
<-> 0x00066a00d8000123 10 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a0098000380 1 CA goblin
<-> 0x00066a00d8000123 15 SW InfiniCon Systems InfinIO9024
2.5g 2048 0x00066a0098000384 1 CA cuda
<-> 0x00066a00d8000123 2 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a0098000384 2 CA cuda
<-> 0x00066a00d8000123 1 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a00980003a6 1 CA erik
<-> 0x00066a00d8000123 9 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a00980003a6 2 CA erik
<-> 0x00066a00d8000123 7 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a00980006a2 1 CA goblin
<-> 0x00066a00d8000123 20 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a0098000849 2 CA rockaway
<-> 0x00066a00d8000123 3 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a0098002813 1 CA brady
<-> 0x00066a00d8000123 19 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a0098002813 2 CA brady
<-> 0x00066a00d8000123 5 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a0098002854 1 CA brady
<-> 0x00066a00d8000123 11 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a0098002854 2 CA brady
<-> 0x00066a00d8000123 6 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a0098003f81 1 CA ibm345
<-> 0x00066a00d8000123 12 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a009800447b 1 CA duster
<-> 0x00066a00d8000123 4 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a009800447b 2 CA duster
<-> 0x00066a00d8000123 16 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a0098004a73 1 CA erik
<-> 0x00066a00d8000123 18 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a00d8000123 14 SW InfiniCon Systems InfinIO9024
<-> 0x00066a10280002cd 2 SW InfiniCon Systems InfiniFabric (Sw A Dev B)
```



3.5.3.8 Converting iba_report output to excel importable files - xml_extract

(Linux) `xml_extract` takes well-formed XML as input, extracts element values as specified by command line options, and outputs the data as lines (records) of data in delimited format (commonly referred to as comma-separated-values (CSV) format). `xml_extract` is intended to be used with `iba_report`, to parse and filter its XML output, and to allow the filtered output to be imported into other tools such as excel spread sheets and customer written scripts. `xml_extract` can also be used with any well-formed XML stream to extract element values into a delimited format.

3.5.3.8.1 Usage

```
xml_extract [-v] [-H] [-d delimiter] [-e extract_element]
            [-s suppress_element] [-X input_file]
            [-P param_file]
```

3.5.3.8.2 Options

`-e/--extract extract_element` - The name of the XML element to extract. Elements can be used multiple times; elements can be nested in any order, but are output in the order specified; an optional attribute (or attribute and value) can also be specified with elements:

```
-e element
-e element:attrName
-e element:attrName:attrValue
```

Elements can be specified multiple times, with a different attribute name or attribute value.

`-s/--suppress suppress_element` - The name of the XML element to suppress; can be used multiple times (in any order); supports the same syntax as `-e`.

`-d/--delimit delimiter` - Use delimiter (single character or string) as the delimiter between element names and element values; default is semicolon

`-X/--infile input_file` - Input XML from *input_file* instead of stdin

`-P/--pfile param_file` - Input command line options (parameters) from *param_file*

`-H/--noheader` - Do not output element name header record

`-v/--verbose` - Verbose output: 1) output progress reports during extraction; and 2) output prepended wildcard characters on element names in output header record.

`xml_extract` is a flexible and powerful tool to process an XML stream; it:

- Requires no specific element names to be present in the XML;
- Assumes no hierarchical relationship between elements;
- Allows extracted element values to be output in any order;
- Allows an element's value to be extracted only in the context (scope) of another (specified) element;
- Allows extraction to be suppressed during the scope of specified elements.

`xml_extract` takes the XML input stream from either stdin or a specified input file. `xml_extract` does not use nor require a connection to a True Scale Fabric.



`xml_extract` works from two lists of elements supplied as command line or input parameters. The first is a list of elements whose values are to be extracted ("extraction elements"). The second is a list of elements for which extraction is to be suppressed ("suppression elements"). When an extraction element is encountered (and extraction is not suppressed), the value of the element is extracted for later output in an "extraction record". An extraction record contains a value for all extraction elements (including those which have a null value).

When a suppression element is encountered, then no extraction will be performed during the extent of that element (start through end). Suppression is maintained for elements specified inside the suppression element, including elements which may happen to match extraction elements. Suppression can be used to prevent extraction in sections of XML which are present, but not of current interest (for example, `NodeDesc` or `NodeGUID` inside a `Neighbor` specification of `iba_report`).

During operation, `xml_extract` outputs an extraction record under the following conditions:

- One or more extraction elements containing a non-null value go out of scope (the element containing the extraction elements is ended) and a record containing the element values has not already been output
- A new and different value is specified for an extraction element and an extraction record containing the previous value has not already been output.

Element names (extraction or suppression) can be made context sensitive with an enclosing element name using the syntax `element1.element2`. In which case, `element2` will be extracted (or extraction will be suppressed) only when `element2` is enclosed by `element1`. The syntax also allows '*' to be specified as a wildcard. '`*.element3`' specifies `element3` enclosed by any element or sequence of elements (ex. `element1.element3` or `element1.element2.element3`). '`element1.*.element3`' specifies `element3` enclosed by `element1` with any number of (but at least 1) intermediate elements. `xml_extract` prepends any entered element name not containing a '*' (anywhere) with '*', matching the element regardless of the enclosing elements.

Note: Any element names which include a wildcard should be quoted to the shell attempting to wildcard match against filenames.

At the beginning of operation `xml_extract`, by default, outputs a delimited "header record" containing the names of the extraction elements. The order of the names is the same as specified on the command line and is the same order as that of the extraction record. Output of the header record can be disabled with the `-H` option. By default, element names are shown as they were entered on the command line. The `-v` option causes element names to be output as they are used during extraction, with any prepended wildcard characters.

Options (parameters) to `xml_extract` can be specified on the command line, with a parameter file, or using both methods. A parameter file is specified with `-P param_file`. When a parameter file specification is encountered on the command line, option processing on the command line is suspended, the parameter file is read and processed entirely, and then command line processing is resumed. Option syntax within a parameter file is the same as on the command line. Multiple parameter file specifications can be made, on the command line or within other parameter files. At each point that a parameter file is specified, current option processing is suspended while the parameter file is processed, then resumed. Options are processed in the order they are encountered on the command line or in parameter files. A parameter file can be up to 8192 bytes in size and may contain up to 512 parameters.



3.5.3.8.3 Sample Use and Output

The following shows a simple example of `iba_report` output filtered by `xml_extract`:

```
>iba_report -o comps -s -x | xml_extract -d \; -e NodeDesc -e SystemImageGUID -e NumPorts -s Neighbor
```

```
NodeDesc;SystemImageGUID;NumPorts
```

```
mindy2 HCA-1;0x0002c9020025a67b;2
```

```
MT25408 ConnectX Mellanox Technologies;0x0002c9030000079b;2
```

```
cuda;0x00066a009800413e;2
```

```
duster;0x00066a009800447b;2
```

```
stewie HCA-1;0x00066a0098007b70;2
```

```
InfiniCon System InfinIO 9024 Lite;0x00066a00d900045f;24
```

```
InfiniCon System InfinIO 9024 Lite;0x00066a00d9000479;24
```

```
InfiniCon System InfinIO 9024 Lite;0x00066a00d90004e9;24
```

```
i9k159 Spine 1, Chip A;0x00066a00da000159;24
```

```
i9k159 Spine 2, Chip A;0x00066a00da000159;24
```

```
i9k159 Leaf 2, Chip A;0x00066a00da000159;24
```

```
i9k159 Leaf 3, Chip A;0x00066a00da000159;24
```

```
i9k159 Leaf 1, Chip A;0x00066a00da000159;24
```

```
i9k159 Leaf 4, Chip A;0x00066a00da000159;24
```

```
i9k159 Leaf 6, Chip A;0x00066a00da000159;24
```

```
i9k159 Leaf 5, Chip A;0x00066a00da000159;24
```

```
i9k159 Leaf 8, Chip A;0x00066a00da000159;24
```

```
i9k159 Leaf 7, Chip A;0x00066a00da000159;24
```

```
i9k159 Spine 1, Chip B;0x00066a00da000159;24
```

```
i9k159 Spine 2, Chip B;0x00066a00da000159;24
```

```
i9k159 Spine 1, Chip A;;
```

```
i9k159 Spine 2, Chip A;;
```

3.5.3.8.4 Sample Scripts

Five sample scripts are available as examples of how to use `xml_extract` and as prototypes for customized scripts. They combine various calls to `iba_report` with a call to `xml_extract` with commonly used parameters.



3.5.3.8.5 iba_extract_perf

`iba_extract_perf` provides a report of all the performance counters in a format easily imported to excel for further analysis.

It generates a detailed `iba_report` component summary report and pipes the result to `xml_extract`, extracting element values for `NodeDesc`, `SystemImageGUID`, `PortNum`, and all the performance counters. Extraction is performed only from the Systems portion of the report which does not contain Neighbor information (the Neighbor and SMs portions are suppressed).

The implementation of the script is as follows:

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

3.5.3.8.6 iba_extract_error

`iba_extract_error` is very similar to `iba_extract_perf`, however it only reports error counters. Its output is easily imported into excel for further analysis of fabric errors.

It generates the same `iba_report` as `iba_extract_perf` but extracts error counters (a subset of the performance counters). Extraction from the Neighbor and SMs portions of the report is suppressed.

The implementation of the script is as follows:

```
iba_report -o comps -s -x -d 10 | xml_extract -d \; -e NodeDesc -e
SystemImageGUID -e PortNum -e SymbolErrors -e LinkErrorRecovery -e LinkDowned -e
PortRcvErrors -e PortRcvRemotePhysicalErrors -e PortRcvSwitchRelayErrors -e
PortXmitConstraintErrors -e PortRcvConstraintErrors -e LocalLinkIntegrityErrors -e
ExcessiveBufferOverrunErrors -s Neighbor -s SMs
```

3.5.3.8.7 iba_extract_link

`iba_extract_link` produces an excel importable summary of the fabric topology.

`iba_extract_link` generates an `iba_report` links report and pipes the result to `xml_extract`, extracting element values for Link, Cable and Port (the port element names are context-sensitive). `iba_extract_link` uses the same logic as `iba_extract_stat` to merge the 2 link records into a single record and remove redundant information.

The portion of the script which calls `iba_report` and `xml_extract` follows:

```
iba_report -x -o links | xml_extract -d \; -e Rate -e MTU -e LinkDetails -e
CableLength -e CableLabel -e CableDetails -e Port.NodeDesc -e Port.PortNum
```

3.5.3.9 Remove All Specified XML Tags - xml_filter

`xml_filter` is the opposite of `xml_extract`. It processes an XML file and removes all the specified tags. The remaining tags are output and indentation can also be reformatted.

3.5.3.9.1 Usage

```
xml_filter [-t|-k] [-l] [-i indent] [-s element] [-P param_file] [input_file]
```




3.5.3.9.2 Options

- t - Trim leading and trailing whitespace in tag contents.
 - k - In tags with whitespace containing new lines, keep the new lines as is (default is to format as an empty list).
 - l - Add comments with line numbers after each end tag. This can make comparison of resulting files easier since original line numbers will be available.
 - i *indent* - Set indentation to use per level (default 4).
 - s *element* - Name of XML element to suppress can be used multiple times, order does not matter.
 - P *param_file* - Read command parameters from *param_file*.
- input_file* - XML file to read. Default is stdin.

3.5.3.10 Re-indenting XML files - `xml_indent`

`xml_indent` can adjust indentation for easier human readability or remove it compact XML files.

3.5.3.10.1 Usage

```
xml_indent [-t|-k] [-i indent] [input_file]
```

3.5.3.10.2 Options

- t - Trim leading and trailing whitespace in tag contents.
 - k - In tags with whitespace containing new lines, keep the new lines as is (default is to format as an empty list).
 - i *indent* - Set indentation to use per level (default 4).
- input_file* - XML file to read. Default is stdin.

3.5.4 `iba_findgood`

The `iba_findgood` command can check for hosts which are pingable, ssh'able and active on the True Scale Fabric and produce a list of good hosts meeting all criteria. The resulting *good* file can then be used in as input to create `mpi_hosts` files for use running `mpi_apps` and the HCA-SW cabletest. Typical usage would be to identify good hosts which will undergo further testing and benchmarking during initial cluster staging and startup. This command assumes the Node Description for each host will be based on the *hostname* -s output in conjunction with an optional HCA-# suffix. These names are the default when using OFED.

When using a `/etc/sysconfig/iba/hosts` file which lists the IPoIB hostnames, this assumption may not be correct. The files created (good, alive, running, active, bad) are in `iba_sorthosts` order with all duplicates removed.

This command automatically generates the file `FF_RESULT_DIR/punchlist.csv`. This file provides a concise summary of the bad hosts found. This can be imported into excel directly as a *.csv file, or be cut/pasted into Excel, and then the "Data/Text to Columns" toolbar can be used to separate the information into multiple columns at the semicolons. Following is a sample of the output that is generated:

```
2012/01/06 11:13:48;trash;Doesn't ping
```



```
2012/01/06 11:13:48;mybadhost;Can't ssh
```

```
2012/01/06 11:13:48;mindy;No active port
```

For a given run a line is generated for each failing host. Hosts are reported exactly once for a given run. Therefore, a host that does not ping will NOT be listed as ""can't ssh"" nor ""No active port"". It should be noted that there may be cases where ports could be active for hosts that do not ping, especially if Ethernet host names are being used for the ping test. However, the lack of ping often implies there are other fundamental issues (e.g., PXE boot, inability to access DNS or DHCP to get proper host name and IP address, etc.), which implies that reporting hosts that do not ping also lack active ports will typically be of limited value.

Note that the approach `iba_findgood` uses to determine hosts with active ports is to query the SA for NodeDescriptions. As such, ports may be active for hosts that cannot be ssh'ed or pinged.

3.5.4.1 Usage

```
iba_findgood [-RA] [-d dir] [-f hostfile]
```

or

```
iba_findgood --help
```

3.5.4.2 Options

--help - produce full help text

-R - skip the running test (ssh), recommended if password-less ssh not setup.

-A - skip the active test, recommended if True Scale Fabric software or fabric is not up.

-d *dir* - directory in which to create alive, active, running, good and bad files default is `/etc/sysconfig/iba`.

-f *hostfile* - file with hosts in cluster, default is `/etc/sysconfig/iba/hosts`.

The files alive, running, active, good, and bad are created in the selected directory listing hosts passing each criteria. The good file can be used as input for an `mpi_hosts`. It will list each good host exactly once.

3.5.4.3 Usage Examples

```
iba_findgood
```

```
iba_findgood -f allhosts
```

3.5.5 iba_saquery

(All) `iba_saquery` can perform various queries of the subnet manager/subnet agent and provide detailed fabric information.

In many cases `iba_report` provides a more powerful tool, however in some cases `iba_saquery` is preferred, especially when dealing with virtual fabrics, service records and multicast.

The command `iba_saquery` is installed on all hosts as part of the OFED+, but it is also included in Intel® FastFabric Toolset. As such it can be a useful tool to run on the Intel® FastFabric Toolset host and is therefore also documented here.



By default `iba_saquery` uses the first active port on the local system. However, if the Fabric Management Node is connected to more than one fabric (for example, a subnet), the HCA and port may be specified to select the fabric whose SA is to be queried.

3.5.5.1 Usage

```
iba_saquery [-v] [-o type][-l lid] [-t type] [-s guid] [-n guid] [-k pkey] [-g guid] [-u gid] [-m gid] [-d name]
```

or

```
iba_saquery --help
```

3.5.5.2 Options

```
--help - Produce full help text
-v/--verbose - Verbose output
-o/--output type - Output type for query (default is node)
-l/--lid lid - Query a specific lid
-t/--type type - Query by node type
-s/--sysguid guid - Query by system image guid
-n/--nodeguid guid - Query by node guid
-k/--pkey pkey - Query a specific pkey
-g/--portguid guid - Query by port guid
-u/--portgid gid - Query by port gid
-m/--mcgid gid - Query by multicast gid
-d/--desc name - Query by node name/description
```

3.5.5.3 Node Types

```
ca - Channel adapter
sw - Switch
rtr - Router
```

3.5.5.4 GIDs

Specify a 64 bit subnet and 64 bit interface ID as:
subnet:interface.

For example:

```
0xfe80000000000000:0x00066a00a0000380
```

3.5.5.5 Output Types

```
systemguid - List of system image guides
nodeguid - List of node guides
```



portguid – List of port guides
 lid – List of lids
 desc – List of node descriptions/names
 node – List of node records
 portinfo – List of port info records
 sminfo – List of SM info records
 swinfo – List of switch info records
 vswinfo – List of vendor switch info records
 service – List of service records
 mcmember – List of multicast member records
 vfinfo – List of vFabrics
 vfinfocsv – List of vFabrics in CSV format
 vfinfocsv2 – List of vFabrics in CSV format [root@luanne sysconfig]#

The vfinfocsv and vfinfocsv2 output formats are designed to make it easier to script vfinfo queries. One line is output per vFabric of the form:

name:index:pkey:sl:mtu:rate

The only difference between these two formats is how the mtu and rate are output. vfinfocsv outputs them in human/text format such as 2048 and 40g. vfinfocsv2 outputs them as the IBTA enumerations defined for the SMA protocol such as 4 and 7. The iba_getvf command for a useful tool which is based on this capability of iba_saquery.

Table 3 shows the combinations of input (assorted query by options) and output (-o) that are permitted.

Table 3. Input Combinations

Input option	-o output permitted	-o output not permitted
None	systemguid, nodeguid, portguid, lid, desc, node, portinfo, sminfo, swinfo, service, mcmember, vfinfo, vfinfocsv, vfinfocsv2, vswinfo	
-t node_type	systemguid, nodeguid, portguid, lid, desc, node	portinfo, sminfo, swinfo, service, mcmember, vfinfo, vfinfocsv, vfinfocsv2, vswinfo
-l lid	systemguid, nodeguid, portguid, lid, desc, node, portinfo, swinfo, service, mcmember, vswinfo	sminfo, vfinfo, vfinfocsv, vfinfocsv2
-s system_image_guid	systemguid, nodeguid, portguid, lid, desc, node	portinfo, sminfo, swinfo, slvl, vlarb, pkey, guides, service, mcmember, inform, linfdb, ranfdb, mcfdb, trace, vfinfo, vfinfocsv, vfinfocsv2, vswinfo
-n node_guid	systemguid, nodeguid, portguid, lid, desc, node	portinfo, sminfo, swinfo, service, mcmember, vfinfo, vfinfocsv, vfinfocsv2, vswinfo
-k pkey	mcmember, path, vfinfo, vfinfocsv, vfinfocsv2	

**Table 3. Input Combinations (Continued)**

Input option	-O output permitted	-O output not permitted
-g port_guid	systemguid, nodeguid, portguid, lid, desc, node, service, mcmember	portinfo, sminfo, swinfo, vfinfo, vfinfocsv, vfinfocsv2, vswinfo
-u port_gid	service, mcmember	systemguid, nodeguid, portguid, lid, desc, node, portinfo, sminfo, swinfo, vfinfo, vfinfocsv, vfinfocsv2, vswinfo
-m multicast_gid	mcmember, vfinfo, vfinfocsv, vfinfocsv2	systemguid, nodeguid, portguid, lid, desc, node, portinfo, sminfo, swinfo, service, vswinfo
-d node_description	systemguid, nodeguid, portguid, lid, desc, node	portinfo, sminfo, swinfo, service, mcmember, vswinfo
-g port_guid	systemguid, nodeguid, portguid, lid, desc, node, service, mcmember	portinfo, sminfo, swinfo, vswinfo
-u port_gid	service, mcmember	systemguid, nodeguid, portguid, lid, desc, node, portinfo, sminfo, swinfo, vswinfo
-m multicast_gid	mcmember	systemguid, nodeguid, portguid, lid, desc, node, portinfo, sminfo, swinfo, service, vswinfo
-d name	systemguid, nodeguid, portguid, lid, desc, node	portinfo, sminfo, swinfo, service, mcmember, vfinfo, vfinfocsv, vfinfocsv2, vswinfo

3.5.6 iba_getvf

This command is designed to help when scripting application use of vFabrics, such as for mpirun parameters. It can fetch the Virtual Fabric info in a delimited format. It returns exactly 1 matching VF. When multiple VFs match the query, it prefers non-Default VFs which the calling server is a full member in. If multiple choices remain, it returns the one with the lowest VF Index (for example, Index typically represents order in config file). This algorithm is the same as that used by the Distributed SA.

The tool is intended to be part of additional scripts to help set PKey, SL, MTU and Rate when running MPI jobs.

3.5.6.1 Usage

```
iba_getvf [-h hca] [-p port] [-e] [-d vfname|-S serviceId|-m mcgid|-i vfIndex|-k pkey|-L sl]
```

or

```
iba_getvf --help
```

3.5.6.2 Options

- help - Produce full help text
- h hca - HCA to send by, default is 1st hca
- p port - Port to send by, default is 1st active port
- e - Output mtu and rate as enum values, 0=unspecified
- d vfname - Query by VirtualFabric Name
- S serviceId - Query by Application ServiceId



- m *gid* – Query by Application Multicast GID
- i *vfindex* – Query by VirtualFabric Index
- k *pkey* – Query by VirtualFabric PKey
- L *SL* – Query by VirtualFabric SL

3.5.6.3 Usage Examples

```
iba_getvf -d 'Compute'
iba_getvf -h 2 -p 2 -d 'Compute'
```

The output is of the form:

name:index:pkey:sl:mtu:rate

3.5.6.4 Sample Outputs

```
# iba_getvf -d Default
Default:0:0xffff:0:unlimited:unlimited
```

Options allow for query by VF Name, VF Index, Service ID, MGID, PKey or SL.

Internally this is based on the `iba_saquery -o vfinfocsv` command

3.5.7 iba_getvf_env

This is a script designed to be included in bash scripts. It provides the `iba_getvf_func` and `iba_getvf2_func` shell functions which can be invoked to query a vFabric's parameters and export the values in the specified shell variables to indicate the PKEY, SL, MTU and RATE associated with the vFabric. An example of its use is provided in `/opt/iba/src/mpi_apps/ofed.openmpi.params`

3.5.8 iba_gen_ibnodes

This tool analyzes the present fabric and produces a list of Intel® Externally Managed switches in the format required for use in the `/etc/sysconfig/iba/ibnodes` file.

3.5.8.1 Usage

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

or

```
iba_gen_ibnodes --help
```

3.5.8.2 Options

--help – Produce full help text

-t *portsfile* – File with list of local HCA ports used to access fabric(s) for analysis, default is `/etc/sysconfig/iba/ports`

-p *ports* – List of local HCA ports used to access fabric(s) for analysis. Default is 1st active port. *Ports* is specified as `hca:port`
0:0 = 1st active port in system



0:y = port y within system
 x:0 = 1st active port on HCA x
 x:y = HCA x, port y
 The first HCA in the system is 1. The first port on an HCA is 1.

- R - Do not attempt to get routes for computation of distance
- s - Update/resolve ibnodes switch names using topology XML data
- L *ibnodes_file* - Use *ibnodes_file* as ibnodes input (do not generate ibnodes data; must also use -s)
- o *output_file* - Write ibnodes data to *output_file* (default is stdout)
- T *topology_file* - Use *topology_file* XML to update ibnodes NodeDesc values
- X *snapshot_file* - Use *snapshot_file* XML for fabric link information (may contain '%P'; must also use -s)
- v *level* - Verbose level (0-8, default 0)
 - 0 - No output
 - 1 - Progress output
 - 2 - Reserved
 - 4 - Time stamps
 - 8 - Reserved
- K - Do not clean temporary files

3.5.8.3 Environment

ports - List of ports, used in absence of -t and -p

portsfile - File containing list of ports, used in absence of -t and -p

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

3.5.8.4 Usage Examples

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

3.5.9 iba_gen_chassis

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

3.5.9.1 Usage

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



or

```
iba_gen_chassis --help
```

3.5.9.2 Options

--help – Produce full help text

-t *portsfile* – File with list of local HCA ports used to access fabric(s) for analysis, default is /etc/sysconfig/iba/ports

-p *ports* – List of local HCA ports used to access fabric(s) for analysis. Default is 1st active port. This is specified as hca:port

0:0 = 1st active port in system

0:y = port y within system

x:0 = 1st active port on HCA x

x:y = HCA x, port y

The first HCA in the system is 1. The first port on an HCA is 1.

3.5.9.3 Environment

ports – List of ports, used in absence of -t and -p

portsfile – File containing list of ports, used in absence of -t and -p

3.5.9.4 Usage Examples

```
iba_gen_chassis
```

```
iba_gen_chassis -p '1:1 1:2 2:1 2:2'
```

```
iba_gen_chassis >> /etc/sysconfig/iba/chassis
```

or while editing the file use a vi command to include its output such as:

```
:r! iba_gen_chassis
```

3.5.10 iba_gen_esm_chassis

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

3.5.10.1 Usage

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

or

```
iba_gen_esm_chassis --help
```

3.5.10.2 Options

--help – Produce full help text

-u *user* – User to perform command as for chassis default is admin

-S – Securely prompt for password for user on chassis



`-t portsfile` – File with a list of local HCA ports used to access fabric(s) for analysis, default is `/etc/sysconfig/iba/ports`

`-p ports` – List of local HCA ports used to access fabric(s) for analysis. Default is 1st active port. This is specified as `hca:port`

`0:0` = 1st active port in system

`0:y` = port `y` within system

`x:0` = 1st active port on HCA `x`

`x:y` = HCA `x`, port `y`

The first HCA in the system is 1. The first port on an HCA is 1.

3.5.10.3 Environment

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

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

`portsfile` – File containing list of ports, used in absence of `-t` and `-p`

3.5.10.4 Usage Examples

```
iba_gen_esm_chassis
```

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

```
iba_gen_esm_chassis >> /etc/sysconfig/iba/esm_chassis
```

or while editing the file use a `vi` command to include its output such as:

```
:r! iba_gen_esm_chassis
```

3.5.11 iba_smaquery

(All) This tool can perform the majority of IBTA defined SMA queries and display the resulting response. It should be noted that each query is issued directly to the SMA and does not involve SM interaction.

3.5.11.1 Usage

```
iba_smaquery [-v] [-d] [-o otype] [-l lid]
[-m dest_port|inport,outputport] [-h hca] [-p port] [-K mkey]
[-f flid] [-b block] [hop hop ...]
```

3.5.11.2 Options

`-v` – Verbose output.

`-d` – Turn on debug.

`-o otype` – Output type. Valid otypes are: `nodedesc` (or `desc`), `nodeinfo` (or `node`), `portinfo`, `sminfo`, `swinfo`, `slvl`, `vlarb`, `pkey`, `guids`, `linfdb`, `ranfdb`, `mcfdb`, `vswinfo`, `portgroup`, `lidmask`.

`-l lid` – Destination lid, default is local port.

`-m dest_port` – Port in destination device to query.

`inport`, `outputport` – SLVL's input/output port – Switch only.
Default is to show all port combinations.



-h *hca* - HCA to send by, default is first HCA.

-p *port* - Port to send by, default is port 1.

-K *mkey* - SM management key to access remote ports.

-f *flid* - LID to lookup in forwarding table to select which LFT or MFT block to display. Default is to show entire table.

-b *block* - Block number of either guides, pkey, or ranfdb. Default is to show entire table.

3.5.11.3 Usage Examples

```
iba_smaquery -o nodedesc -l 6 # get nodedesc via lid routed
iba_smaquery -o nodedesc 1 3 # get nodedesc via directed route
                                # (2 dr hops)
iba_smaquery -o nodeinfo -l 2 3 # get nodeinfo via a combination of
                                # lid routed and directed route
                                # (1 dr hop)
iba_smaquery -o portinfo # get local port info
iba_smaquery -o portinfo -l 6 -m 1 # get port info of port 1 of lid 6
iba_smaquery -o slvl -l 6 # get slvl of CA at lid 6
iba_smaquery -o slvl -l 2 -m 2,3 # get slvl of Switch at lid 2
                                # with input port =2,output port=3
iba_smaquery -o mcfdb -l 2 -f 0xc004 # get a block of entries that
                                # includes mc lid 0xc004 from
                                # the MFT of the switch with lid 2
iba_smaquery -o mcfdb -l 2 -f 0xc004 -m 17 # same as above with position bit
iba_smaquery -o linfdb -l 2 -m 1 -f 1 # get a block of entries of port 1 that
                                # includes lid 1 entry from LFT
iba_smaquery -o ranfdb -l 2 -b 5 # get a fixed 10 blocks starting
                                # from block 5 from RFT
iba_smaquery -o guides -l 6 -b 0 # get block 0 of GUIDs
iba_smaquery -o vlarb -l 6 # get vlarb table entries from lid 6
iba_smaquery -o pkey -l 2 3 # get pkey table entries starting
                                # (lid routed to lid 2,
                                # then 1 dr hop to port 3)
iba_smaquery -o swinfo -l 2 # get switch info
iba_smaquery -o sminfo -l 1 # get SM info
```



```
iba_smaquery -o vswinfo -l 2          # get vendor switch info
iba_smaquery -o portgroup -l 2        # get port group info
iba_smaquery -o lidmask -l 2          # get lidmask info
```

3.5.12 iba_paquery

(All) iba_paquery can perform various queries of the performance management/performance administration agent and provide details about fabric performance. Refer to the *Intel® True Scale Fabric Suite Fabric Manager User Guide* for a description of the operation and client services of the PM/PA.

iba_paquery's operation is dependent on a FM version 6.0 or greater running as master SM/PM in the fabric.

By default, iba_paquery uses the first active port on the local system. However, if the Fabric Management Node is connected to more than one fabric (for example, a subnet), the HCA and port may be specified to select the fabric whose PA is to be queried.

3.5.12.1 Usage

```
iba_paquery [-v] [-h hca] [-p port] -o type [-g groupName] [-l nodeLid] [-P
portNumber] [-d delta] [-s select] [-f focus] [-S start] [-r range] [-n imgNum]
[-O imgOff] [-m moveImgNum] [-M moveImgOff]
```

3.5.12.2 Options

```
-v/--verbose - Verbose output
-h/--hca hca - HCA to send by, default is 1st hca
-p/--port port - Port to send by, default is 1st active port
-o/--output type - Output type
-g/--groupName groupName - Group name for groupInfo query
-l/--lid lid - LID of node for portCounters query
-P/--portNumber portNumber - Port number for portCounters query
-d/--delta delta - Delta flag for portCounters query - 0 or 1
-s/--select select - 16-bit select flag for clearing port counters
    Select bits (0 is least significant (rightmost))
    0 - SymbolErrorCounter
    1 - LinkErrorRecoveryCounter
    2 - LinkDownedCounter
    3 - PortRcvErrors
    4 - PortRcvRemotePhysicalErrors
    5 - PortRcvSwitchRelayErrors
    6 - PortXmitDiscards
    7 - PortXmitConstraintErrors
    8 - PortRcvConstraintErrors
    9 - LocalLinkIntegrityErrors
    10 - ExcessiveBufferOverrunErrors
    11 - VL15Dropped
```



12 – PortXmitData
13 – PortRcvData
14 – PortXmitPackets
15 – PortRcvPackets

-f/--focus *focus* – Focus select value for getting focus ports focus select values:
0x00020001 – Sorted by utilization - highest first
0x00020081 – Sorted by packet rate - highest first
0x00020101 – Sorted by utilization - lowest first
0x00030001 – Sorted by integrity errors - highest first
0x00030002 – Sorted by sma congestion errors - highest first
0x00030003 – Sorted by congestion errors - highest first
0x00030004 – Sorted by security errors - highest first
0x00030005 – Sorted by routing errors - highest first
0x00030006 – Sorted by adaptive routing - highest first

-S/--start *start* – Start of window for focus ports - should always be 0 for now

-r/--range *range* – Size of window for focus ports list

-n/--imgNum *imgNum* – 64-bit image number - may be used with groupInfo, groupConfig, portCounters (delta)

-O/--imgOff *imgOff* – Image offset - may be used with groupInfo, groupConfig, portCounters (delta)

-m/--moveImgNum *moveImgNum* – 64-bit image number - used with moveFreeze to move a freeze image

-M/--moveImgOff *moveImgOff* – Image offset - may be used with moveFreeze to move a freeze image

3.5.12.3 Output Types

classPortInfo – Class port info

groupList – List of PA groups

groupInfo – Summary statistics of a PA group – Requires -g option for groupName

groupConfig – Configuration of a PA group – Requires -g option for groupName

portCounters – Port counters of fabric port – Requires -l *lid* and -P *port* options, -d *delta* is optional

clrPortCounters – Clear port counters of fabric port – Requires -l *lid* and -P *port*, and -s *select* options

clrAllPortCounters – Clear all port counters in fabric

pmConfig – Retrieve PM configuration information

freezeImage – Create freeze frame for image ID – Requires -n *imgNum*

releaseImage – Release freeze frame for image ID – Requires -n *imgNum*

renewImage – Renew lease for freeze frame for image ID – Requires -n *imgNum*



`moveFreeze` – Move freeze frame from image ID to new image ID – Requires
`-n imgNum` and `-m moveImgNum`

`focusPorts` – Get sorted list of ports using utilization or error values (from group buckets)

`imageInfo` – Get configuration of a PA image (timestamps, etc.) – Requires
`-n imgNum`

3.5.12.4 Usage Examples

```
iba_paquery -o classPortInfo

iba_paquery -o groupList

iba_paquery -o groupInfo -g All

iba_paquery -o groupConfig -g All

iba_paquery -o portCounters -l 1 -P 1 -d 1

iba_paquery -o portCounters -l 1 -P 1 -d 1 -n 0x20000000d02 -O 1

iba_paquery -o clrPortCounters -l 1 -P 1 -s 0x0048 (clears PortXmitDiscards &
PortRcvErrors)

iba_paquery -o clrAllPortCounters -s 0x0048 (clears PortXmitDiscards &
PortRcvErrors)

iba_paquery -o getPMConfig

iba_paquery -o freezeImage -n 0x20000000d02

iba_paquery -o releaseImage -n 0xd01

iba_paquery -o renewImage -n 0xd01

iba_paquery -o moveFreeze -n 0xd01 -m 0x20000000d02 -M -2

iba_paquery -o focusPorts -g All -f 0x00030001 -S 0 -r 20

iba_paquery -o imageConfig -n 0x20000000d02
```

3.5.13 iba_pmaquery

(All) This is a low level tool which can perform individual PMA queries against a specific LID. It is very useful in displaying port runtime information.

3.5.13.1 Usage

```
iba_pmaquery [-v] [-d] [-o otype] [-l lid] [-m dest_port] [-b select] [-h hca] [-p
port]
```

3.5.13.2 Options

- `-v` – Verbose output
- `-d` – Turn on debug
- `-o otype` – Output type. Valid *otypes* are: `classportinfo`, `stats`, `extstats`, `clearstats`, `clearextstats`, `vendstats`, `clearvendstats`.
- `-l lid` – Destination lid, default is local port



-m *dest_port* - Port in destination device to query/clear required when using -l option for all but -o classportinfo

-b *select* - Counter select for clearstats, clearextstats, and clearvendstats. Default is to clear all.

-h *hca* - HCA to send by, default is first HCA

-p *port* - Port to send by, default is port 1

3.5.13.3 Usage Examples

```
iba_pmaquery -o classportinfo -l 6           # get PMA classportinfo
iba_pmaquery -o stats -l 6 -m 1              # get PMA PortCounters
iba_pmaquery -o clearstats -l 6 -m 1         # clear PMA PortCounters
iba_pmaquery -o clearstats -l 6 -m 1 -b 0xff # clear PMA error PortCounters
iba_pmaquery -o extstats -l 6 -m 1           # get PMA PortCountersExtended
iba_pmaquery -o clearextstats -l 6 -m 1      # clear PMA PortCountersExtended
iba_pmaquery -o vendstats -l 6 -m 1          # get PMA Vendor PortCounters
iba_pmaquery -o clearvendstats -l 6 -m 1     # clear PMA PortCounters
```

3.5.13.4 Sample Outputs

```
[root@luanne ~]# iba_pmaquery

Performance: Transmit

    Xmit Data                0 MiB (72 Quads)
    Xmit Pkts                 1

Performance: Receive

    Rcv Data                 0 MiB (0 Quads)
    Rcv Pkts                 0

Errors:

    Symbol Errors            0
    Link Error Recovery      0
    Link Downed              0
    Port Rcv Errors          0
    Port Rcv Rmt Phys Err    0
    Port Rcv Sw Relay Err    0
    Port Xmit Discards       2
    Port Xmit Constraint     0
    Port Rcv Constraint      0
    Local Link Integrity     0
```



```
Exc. Buffer Overrun          0
VL15 Dropped                 2
```

3.5.14 iba_fequery

(All) This tool can be helpful when testing or debugging PA operations to the FE. This tool performs the custom PA client/server queries. The output formats and arguments are very similar to `iba_paquery`.

3.5.14.1 Usage

```
iba_fequery [-v] [-a ipAdr | -h hostName] -o type [-g groupName] [-l nodeLid] [-P
portNumber] [-d delta] [-s select] [-f focus] [-S start] [-r range] [-n imgNum]
[-O imgOff] [-m moveImgNum] [-M moveImgOff]
```

3.5.14.2 Options

```
-v/--verbose - Verbose output.
-a/--ipAdr ipAdr - IP address of node running the FE.
-h/--hostName hostName - Host name of node running the FE.
-o/--output output - Output type.
-g/--groupName groupName - Group name for groupInfo query.
-l/--lid lid - LID of node for portCounters query.
-P/--portNumber portNumber - Port number for portCounters query.
-d/--delta delta - Delta flag for portCounters query - 0 or 1.
-s/--select select - 16-bit select flag for clearing port counters select bits (0 is
least significant (rightmost)):
    0 - SymbolErrorCounter          8 - PortRcvConstraintErrors
    1 - LinkErrorRecoveryCounter    9 - LocalLinkIntegrityErrors
    2 - LinkDownedCounter          10 - ExcessiveBufferOverrunErrors
    3 - PortRcvErrors              11 - VL15Dropped
    4 - PortRcvRemotePhysicalErrors 12 - PortXmitData
    5 - PortRcvSwitchRelayErrors    13 - PortRcvData
    6 - PortXmitDiscards            14 - PortXmitPackets
    7 - PortXmitConstraintErrors    15 - PortRcvPackets
-f/--focus focus - Focus select value for getting focus ports focus select values:
    0x00020001 - Sorted by utilization - highest first.
    0x00020081 - Sorted by packet rate - highest first.
    0x00020101 - Sorted by utilization - lowest first.
    0x00030001 - Sorted by integrity errors - highest first.
    0x00030002 - Sorted by sma congestion errors - highest first.
    0x00030003 - Sorted by congestion errors - highest first.
    0x00030004 - Sorted by security errors - highest first.
    0x00030005 - Sorted by routing errors - highest first.
    0x00030006 - Sorted by adaptive routing - highest first.
-S/--start start - Start of window for focus ports - should always be 0 for now.
```



-r/--range *range* - Size of window for focus ports list.

-n/--imgNum *imgNum* - 64-bit image number - may be used with groupInfo, groupConfig, portCounters (delta).

-O/--imgOff *imgOff* - Image offset - may be used with groupInfo, groupConfig, portCounters (delta).

-m/--moveImgNum *moveImgNum* - 64-bit image number - used with moveFreeze to move a freeze image.

-M/--moveImgOff *moveImgOff* - Image offset - may be used with moveFreeze to move a freeze image.

3.5.14.3 Output Types

classPortInfo - Class port info.

groupList - List of PA groups.

groupInfo - Summary statistics of a PA group - Requires -g option for *groupName*.

groupConfig - Configuration of a PA group - Requires -g option for *groupName*.

portCounters - Port counters of fabric port - Requires -l *lid* and -P *port* options, -d *delta* is optional.

pmConfig - Retrieve PM configuration information.

freezeImage - Create freeze frame for image ID - Requires -n *imgNum*.

releaseImage - Release freeze frame for image ID - Requires -n *imgNum*.

renewImage - Renew lease for freeze frame for image ID - Requires -n *imgNum*.

moveFreeze - Move freeze frame from image ID to new image ID - Requires -n *imgNum* and -m *moveImgNum*.

focusPorts - Get sorted list of ports using utilization or error values (from group buckets).

imageInfo - Get information about a PA image (timestamps, and so on.) - Requires -n *imgNum*.

3.5.14.4 Examples

```
iba_fequery -o classPortInfo
iba_fequery -h stewie -o classPortInfo
iba_fequery -a 172.21.2.155 -o classPortInfo
iba_fequery -o groupList
iba_fequery -o groupInfo -g All
iba_fequery -o groupConfig -g All
iba_fequery -h stewie -o groupInfo -g All
iba_fequery -a 172.21.2.155 -o groupInfo -g All
iba_fequery -o portCounters -l 1 -P 1 -d 1
```




```
iba_fequery -o portCounters -l 1 -P 1 -d 1 -n 0x20000000d02 -O 1
iba_fequery -o pmConfig
iba_fequery -o freezeImage -n 0x20000000d02
iba_fequery -o releaseImage -n 0xd01
iba_fequery -o renewImage -n 0xd01
iba_fequery -o moveFreeze -n 0xd01 -m 0x20000000d02 -M -2
iba_fequery -o focusPorts -g All -f 0x00030001 -S 0 -r 20
iba_fequery -o imageInfo -n 0x20000000d02
```

3.5.15 iba_ccaquery

The iba_ccaquery queries CCA on S20, Intel® HCAs, and Mellanox Devices. It will decide if the vendor specific CCA packets should be used or standard packets.

3.5.15.1 Usage

```
iba_ccaquery [-v] [-d] [-o otype] [-l lid] [-h hca] [-p port] [-K cckey] [-b block]
```

3.5.15.2 Options

- v – Verbose output
- d – Turn on debug
- o otype – Output type, valid otypes are:
classportinfo, key, info, log, swsetting, swportsetting, casetting,
ctltable, timestamp
- l lid – Destination lid, default is local port
- h hca – HCA to send via, default is 1st HCA
- p port – Port to send via, default is port 1
- K cckey – CC management key to access remote ports
- b block – Block number of swportsetting or ctltable. Default is to show entire table

3.5.15.3 Examples

```
iba_ccaquery -o classportinfo -l 6      # get CCA classportinfo
iba_ccaquery -o info -l 6               # get CCA Info
iba_ccaquery -o key -l 6                # get CCA KeyInfo
iba_ccaquery -o log -l 6                # get event log
iba_ccaquery -o swsetting -l 6          # get CCA Switch Setting
iba_ccaquery -o swportsetting -l 6 -b 0 # get CCA Switch Port Settings block 0
iba_ccaquery -o casetting -l 2          # get CCA CA Setting
iba_ccaquery -o ctltable -l 2           # get CCA CA Control Table
```



```
iba_ccaquery -o timestamp -l 2          # get CCA running timestamp
```

3.5.16 iba_extract_bad_links

(Linux) Produces a csv file listing all the links that exceed the present or specified `iba_report -o errors thresholds`. The output from this tool can be reviewed and supplied as input to `iba_disable_ports`.

3.5.16.1 Usage

```
iba_extract_bad_links [iba_report options]
```

or

```
iba_extract_bad_links --help
```

3.5.16.2 Options

iba_report options - Options will be passed to `iba_report`.

3.5.16.3 Examples

```
iba_extract_bad_links
```

```
iba_extract_bad_links -h 1 -p 2
```

3.5.17 iba_disable_ports

(Linux) Accepts a csv file listing links to disable. For each HCA-SW link, the switch side of the link is disabled. For each SW-SW link, the side of the link with the lower LID (that is typically the side closest to the SM) is disabled. This approach generally permits a future `iba_enable_ports` operation to re-enable the port once the issue is corrected or ready to be retested. When using the `-R` option this tool does not look at the routes, it disables the switch ports with the lower value LID. The list of disabled ports is tracked in `/etc/sysconfig/iba/disabled*.csv`.

3.5.17.1 Usage

```
iba_disable_ports [-R] [-t portsfile] [-p ports] [reason] < disable.csv
```

or

```
iba_disable_ports --help
```

3.5.17.2 Options

`--help` - Produce full help text

`-R` - Do not attempt to get routes for computation of distance instead just disable switch port with lower LID assuming that will be closer to this node

`-t portsfile` - File with list of local HCA ports used to access fabric(s) for operation, default is `/etc/sysconfig/iba/ports`.

`-p ports` - List of local HCA ports used to access fabric(s) for analysis default is 1st active port.

This is specified as `hca:port`

`0:0` = 1st active port in system

`0:y` = port y within system



`x:0` = 1st active port on HCA `x`

`x:y` = HCA `x`, port `y`

The first HCA in the system is 1. The first port on an HCA is 1.

reason – Optional text description of reason ports are being disabled, will be saved at the end of any new lines in the disabled file. For ports already in the disabled file, this is ignored.

`disable.csv` – File listing the links to disable. The list is of the form:

```
NodeGUID;PortNum;NodeType;NodeDesc;NodeGUID;PortNum;NodeType;NodeDesc;Reason
```

For each listed link, the switch port with the lower LID (closer to the SM) will be disabled. The `Reason` field is optional. The `Reason` field and any additional fields provided will be saved in the disabled file. An input file such as this can be generated by `iba_extract_bad_links` or `iba_extract_sel_links`.

3.5.17.3 Environment

The following environment variables are also used by this command:

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

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

3.5.17.4 Examples

```
iba_disable_ports 'bad cable' < disable.csv
```

```
iba_disable_ports -p '1:1 1:2 2:1 2:2' 'dead servers' < disable.csv
```

3.5.18 iba_enable_ports

(Linux) Accepts a disabled ports input file and re-enables the specified ports. The input file can be `/etc/sysconfig/iba/disabled*.csv` or a user-created subset of such a file. After enabling the port, it is removed from `/etc/sysconfig/iba/disabled*.csv`.

3.5.18.1 Usage

```
iba_enable_ports [-t portsfile] [-p ports] < disabled.csv
```

or

```
iba_enable_ports --help
```

3.5.18.2 Options

`--help` – Produce full help text

`-t portsfile` – File with list of local HCA ports used to access fabric(s) for operation, default is `/etc/sysconfig/iba/ports`

`-p ports` – List of local HCA ports used to access fabric(s) for analysis default is 1st active port.

This is specified as **hca:port**

`0:0` = 1st active port in system

`0:y` = port `y` within system

`x:0` = 1st active port on HCA `x`



`x:y` = HCA `x`, port `y`

The first HCA in the system is 1. The first port on an HCA is 1.

`disable.csv` – File listing the ports to enable. It is of the form

`NodeGUID;PortNum;NodeDesc`

A input file like this is generated in `/etc/sysconfig/iba/disabled*`

3.5.18.3 Examples

```
iba_enable_ports < disabled.csv
```

```
iba_enable_ports -p '1:1 1:2 2:1 2:2' < disabled.csv
```

3.5.18.4 Environment

The following environment variables are also used by this command:

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

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

3.5.19 iba_disable_hosts

(Linux) Searches for a set of hosts in the fabric and disables their corresponding switch port.

3.5.19.1 Usage

```
iba_disable_hosts [-h hca] [-p port] reason host ...
```

or

```
iba_disable_hosts --help
```

3.5.19.2 Options

`--help` – Produce full help text

`-h hca` – HCA to send through. The default is the first HCA

`-p ports` – Port to send through. The default is the first active port.

`reason` – Text description of reason hosts are being disabled, will be saved at end of any new lines in disabled file. For ports already in disabled file, this is ignored.

3.5.19.3 Examples

```
iba_disable_hosts 'bad DRAM' compute001 compute045
```

```
iba_disable_hosts -h 1 -p 2 'crashed' compute001 compute045
```

3.5.20 iba_extract_lids

(Linux) Supporting tool that generates a csv file listing the map of LIDs that are currently present in the fabric.

3.5.20.1 Usage

```
iba_extract_lids [-h hca] [-p port]
```



or

```
iba_extract_lids --help
```

3.5.20.2 Options

--help – Produce full help text

-h hca – HCA to send through. The default is the first HCA

-p ports – Port to send through. The default is the first active port.

3.5.20.3 Examples

```
- iba_extract_lids > lids.csv
```

```
- iba_extract_lids -h 2 -p 1 > lids.csv
```

3.6 Advanced Chassis Initialization and Verification

3.6.1 iba_chassis_admin

(Switch) `iba_chassis_admin` performs a number of multi-step operations. In general operations performed by `iba_chassis_admin` involve a login to one or more Intel® Chassis. `iba_chassis_admin` can perform initial chassis setup, firmware upgrades, reboot chassis and other operations.

3.6.1.1 Usage

```
iba_chassis_admin [-c] [-F chassisfile] [-P packages] [-I fm_bootstate] [-a action]
[-S] [-d upload_dir] operation ...
```

or

```
iba_chassis_admin --help
```

3.6.1.2 Options

--help – Produce full help text

-c – Clobber result files from any previous run before starting this run

-F chassisfile – File with chassis in cluster. The default is `/etc/sysconfig/iba/chassis`

-P packages – Filenames/directories of firmware images to install. For directories specified, all `.pkg` files in directory tree will be used. `shell` wild cards may also be used within quotes,

or for `fmconfig`, filename of FM configuration file to use,
or for `fmgetconfig`, filename to upload to (default `ifs_fm.xml`),

-a action – Action for supplied file.

For chassis upgrade

`push` – Ensure firmware is in primary or alternate

`select` – Ensure firmware is in primary

`run` – Ensure firmware is in primary and running

The default is `push`.

For chassis `fmconfig`:



push – Ensure config file is in chassis
run – After push restart FM on master, stop on slave
runall – After push restart FM on all MM
For chassis fmcontrol:
stop – Stop FM on all MM
run – Make sure FM running on master, stopped on slave
runall – Make sure FM running on all MM
restart – Restart FM on master, stop on slave
restartall – Restart FM on all MM

-I *fm_bootstate* – Fmconfig and fmcontrol install options
disable – Disable FM start at chassis boot
enable – Enable FM start on master at chassis boot
enableall – Enable FM start on all MM at chassis boot

-d *upload_dir* – Directory to upload FM config files to, default is uploads

-S – Securely prompt for password for user on chassis

operation – Operation to perform. Can be one or more of:
reboot – Reboot chassis, ensure they go down and come back
configure – Run wizard to perform chassis configuration
upgrade – Upgrade install of all chassis
getconfig – Get basic configuration of chassis
fmconfig – FM config operation on all chassis
fmgetconfig – Fetch FM config from all chassis
fmcontrol – Control FM on all chassis

3.6.1.3 For example

```
iba_chassis_admin -c reboot
```

```
iba_chassis_admin -a run -P '*.pkg' upgrade
```

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

- `test.res` – Appended with summary results of run
- `test.log` – Appended with detailed results of run
- `save_tmp/` – Contains a directory per failed test with detailed logs
- `test_tmp*/` – Intermediate result files while test is running

The `-c` option will remove all of the above.

For operations against chassis, setup of ssh keys (see “[setup_ssh](#)” on page 51) is recommended. If ssh keys are not setup, all chassis must be configured with the same admin password and use of the `-S` option is recommended. The `-S` option avoids the need to keep the password in configuration files.

Results from `iba_chassis_admin` are grouped into Test Suites, Test Cases and Test Items. A given run of `iba_chassis_admin` represents a single Test Suite. Within a Test Suite multiple Test Cases will occur, typically one Test Case per chassis being operated on. Some of the more complex operations may have multiple Test Items per Test Case. Each such item represents a major step in the overall Test Case.

Each `iba_chassis_admin` run appends to `test.res` and `test.log`, and creates temporary files in `test_tmp$PID` in the current directory. `test.res` will provide an overall summary of operations performed and their results. The same information will



also be displayed while `iba_chassis_admin` is executing. `test.log` will contain detailed information about what was performed. This will include the specific commands executed and the resulting output. The `test_tmp` directories will contain temporary files which reflect tests in progress (or killed). The logs for any failures will be logged in the `save_temp` directory with a directory per failed test case. If the same test case fails more than once, `save_temp` will retain the information from the first failure and subsequent runs of `iba_chassis_admin` will only append to `test.log`. It is recommended to review failures and use the `-c` option to remove old logs before subsequent runs of `iba_chassis_admin`.

`iba_chassis_admin` implicitly performs its operations in parallel. Twenty (20) parallel operations is the default.

3.6.2 iba_chassis_admin Chassis Operations

(Switch) All chassis operations will login to the chassis as chassis user admin. It is recommended to use the `-s` option to securely prompt for a password, in which case the same password is used for all chassis. Alternately, the password may be put in the environment or the `fastfabric.conf` file through `FF_CHASSIS_ADMIN_PASSWORD`.

Note: All versions of Intel® 12000 Chassis firmware permit SSH keys to be configured within the chassis for secure password-less login. In this case there is no need to configure a `FF_CHASSIS_ADMIN_PASSWORD` and `FF_CHASSIS_LOGIN_METHOD` can be SSH. Refer to “[setup_ssh](#)” on page 51, and the *Intel® True Scale Fabric Switches 12000 Series Users Guide* for more information.

3.6.2.1 upgrade

This upgrades the firmware on each chassis or slot specified. The `-P` option selects a directory containing `.pkg` files or provides an explicit list of `.pkg` files for the chassis and/or slots. The `-a` option selects the desired minimal state for the new firmware. For each chassis and/or slot selected for upgrade, the `.pkg` file applicable to that slot will be selected and used. If more than one `.pkg` file is specified of a given card type, the operation is undefined.

The upgrade is intelligent and does not upgrade chassis that already have the desired firmware in the desired state (as specified by `-a`).

When the `-a` option specifies `run`, chassis that are not already running the desired firmware will be rebooted. By selecting the proper `FF_MAX_PARALLEL` value, a rolling upgrade or a parallel upgrade may be accomplished. In most cases a parallel upgrade is recommended for expediency.

For more information about chassis firmware refer to the *Intel® True Scale Fabric Switches 12000 Series Users Guide*, *Intel® True Scale Fabric Switches 12000 Series Release Notes*, *Intel® 9000 Users Guide* and *SilverStorm 9000 Release Notes*.

3.6.2.2 configure

This runs the chassis setup wizard, which asks the user a series of questions. Once the wizard has finished prompting for configuration information, all the selected chassis are configured using the CLI interface according to the responses to the questions. The following items are configurable for all chassis:

- syslog server IP address, TCP/UDP port number, syslog facility code and the chassis LogMode
- NTP server
- local timezone
- maximum packet MTU

- VL Capability
- VL credit distribution
- Link Width supported
- IB Node Description
- IB Node Description format
- disable chassis auto clear of port counters

Note: In a fabric where FastFabric tools such as `iba_rfm` and `iba_top` that work in conjunction with the PM/PA to monitor port counters, it is required to disable the chassis port counter auto-clear feature.

3.6.2.3 **reboot**

This reboots the given chassis and ensures they go down and come back up by pinging them during the reboot process.

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

3.6.2.4 **getconfig**

This retrieves basic information from a chassis such as syslog, NTP configuration, timezone info, MTU Capability, VL Capability, VL Credit Distribution, Link Width and node description.

3.6.2.5 **fmconfig**

This updates the FM config file on each chassis specified. The `-P` option selects a file to transfer to the chassis. The `-a` option selects the desired minimal state for the new configuration and will control if the FM is started/restarted after the file is updated.

The `-I` option can be used to configure the FM start at boot for the selected chassis.

3.6.2.6 **fmgetconfig**

This uploads the FM config file from all selected chassis. The file is uploaded to the selected uploads directory. The `-P` option can specify the desired destination filename within the uploads directory.

3.6.2.7 **fmcontrol**

This allows the FM to be controlled on each chassis specified. The `-a` option selects the desired state for the FM.

The `-I` option can be used to configure the FM start at boot for the selected chassis.

3.7 **Externally Managed Switch Initialization and Verification**

3.7.1 **iba_switch_admin**

(Switch) `iba_switch_admin` performs a number of multi-step operations. against one or more Intel® externally-managed switches. `iba_switch_admin` can perform firmware upgrades, reboot switches as well as perform a variety of other operations.



3.7.1.1 Usage

```
iba_switch_admin [-c] [-L nodeFile] [-d upload_dir] [-S] [-s] [-P packages] [-a
action] [-O override] operation ...
```

or

```
iba_switch_admin --help
```

3.7.1.2 Options

--help – Produce full help text

-c – Clobber result files from any previous run before starting this run

-L *nodefile* – File with nodes in cluster. The default is
/etc/sysconfig/iba/ibnodes

-d *upload_dir* – Directory to upload capture files to (default is uploads)

-S – Securely prompt for password for user on remote system/chassis test – Test to run.

-s – Securely prompt for new password for switch – Valid only with upgrade operation.

-P *packages* – Filename/directory of firmware image to install. For the directory specified, .emfw files in the directory tree will be used. *shell* wild cards may also be used within quotes.

-a *action* – For firmware file for chassis upgrade

select – Ensure firmware is in primary

run – Ensure firmware is in primary and running

The default is select.

-O *override* – Override for firmware upgrades, bypass the previous firmware version checks, and force the update unconditionally.

operation – The *operation* argument can be one or more of the following:

reboot – Reboot switches, ensure they go down and come back

configure – Run wizard to setup switch node configuration

upgrade – Upgrade install of all switches.

info – Report firmware and hardware version, part number and capabilities of all switches.

hwvdp – Complete Vital Product Data (VPD) report of all switches.

ping – Ping all switches – Test for presence

fwverify – Report integrity of firmware of all nodes.

capture – Captures switch hardware and firmware state(s) of all nodes.

getconfig – Get port configurations of a externally managed switch. Outputs a summary of how many switches have each value for the various configuration settings. This helps to make it more obvious if all switches have the same configuration, and if not, indicates how many have each value.

3.7.1.3 Example

```
iba_switch_admin -c reboot
```

```
iba_switch_admin -P /root/ChassisFwX.X.X.X.X upgrade
```

```
iba_switch_admin -a run -P '*.emfw' upgrade
```



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

- `test.res` – Appended with summary results of run
 - `test.log` – Appended with detailed results of run
 - `save_tmp/` – Contains a directory per failed test with detailed logs
 - `test_tmp*/` – Intermediate result files while test is running
- The `-c` option will remove all of the above.

Results from `iba_switch_admin` are grouped into Test Suites, Test Cases and Test Items. A given run of `iba_switch_admin` represents a single Test Suite. Within a Test Suite multiple Test Cases will occur, typically one Test Case per being operated on. Some of the more complex operations may have multiple Test Items per Test Case. Each such item represents a major step in the overall Test Case.

Each `iba_switch_admin` run appends to `test.res` and `test.log` and creates temporary files in `test_tmp$PID` in the current directory. `test.res` will provide an overall summary of operations performed and their results. The same information will also be displayed while `iba_switch_admin` is executing. `test.log` will contain detailed information about what was performed. This will include the specific commands executed and the resulting output. The `test_tmp` directories will contain temporary files which reflect tests in progress (or killed). The logs for any failures will be logged in the `save_temp` directory with a directory per failed test case. If the same test case fails more than once, `save_temp` will retain the information from the first failure and subsequent runs of `iba_switch_admin` will only append to `test.log`. It is recommended to review failures and use the `-c` option to remove old logs before subsequent runs of `iba_switch_admin`.

`iba_switch_admin` implicitly performs its operations in parallel. Twenty (20) parallel operations is the default.

3.7.2 `iba_switch_admin` Operations

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

3.7.2.1 `reboot`

Reboots the given switches.

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

3.7.2.2 `upgrade`

Upgrades the firmware on each specified switch. The `-P` option selects a directory containing a `.emfw` file or provides an explicit `.emfw` file for the switches. If more than one `.emfw` file is specified, the operation is undefined. The `-a` option selects the desired minimal state for the new firmware. Only the `select` and `run` options are valid for this operation.

When the `-a` option specifies `run`, switches will be rebooted. By selecting the proper `FF_MAX_PARALLEL` value a rolling upgrade or a parallel upgrade may be accomplished. In most cases a parallel upgrade is recommended for expediency.

The upgrade process will also set the switch name. See discussion on “[Selection of Switches](#)” on page 26.



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

For more information about switch firmware refer to the *Intel® True Scale Fabric Switches 12000 Series Users Guide*, *Intel® True Scale Fabric Switches 12000 Series Release Notes*, *Intel® 9000 Users Guide* and *SilverStorm 9000 Release Notes*.

3.7.2.3 configure

This runs the switch setup wizard, which asks the user a series of questions. Once the wizard has finished prompting for configuration information, all the selected switches are configured according to the responses to the questions. The following items are configurable for all Intel® 12000 series switches:

- MTU
- VL Capability
- VL credit distribution
- Link Width Supported
- IB Node Description

Note: If 4X capability is not enabled in the user selection (for example, selecting 8X only, or selecting 1X/8X), 4X capability is added to port 1 for each switch being configured. This is so that it is always possible to “rescue” the switch with FastFabric (by connecting to it with 4X) should the link be unable to connect to with a link width other than 4X.

Note: Normally, the IB Node Description is updated automatically as part of a firmware upgrade, if it is configured properly in the `ibnodes` file. Update of the node description is also available with the `configure` option without the need for a firmware upgrade.

3.7.2.4 info

Queries the switches and displays the following information:

- Firmware version
- Hardware version
- Hardware part number, including revision information
- Speed capability (SDR, DDR)
- Fan Status
- Power Supply Status

3.7.2.5 hwvdpd

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

- Serial Number
- Part Number
- Model Name
- Hardware Version
- Manufacturer
- Product description



- Manufacturer ID
- Manufacture date
- Manufacture time

3.7.2.6 ping

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

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

3.7.2.7 fwverify

Verifies the integrity of the firmware images in the eeproms of the selected switches.

3.7.2.8 capture

Get switch hardware and firmware state capture of all nodes.

3.7.2.9 getconfig

Get port configurations of a externally managed switch.

3.8 Advanced Host Initialization and Verification

3.8.1 iba_host_admin

(Host) `iba_host_admin` performs a number of multi-step operations. In general operations performed by `iba_host_admin` involve a login to one or more host systems. `iba_host_admin` can perform software or firmware upgrades, reboot hosts, as well as perform a variety of host and fabric verification operations.

3.8.1.1 Usage

```
iba_host_admin [-c] [-f hostfile] [-r release] [-d dir] [-T product] [-P packages]  
[-S] operation ...
```

or

```
iba_host_admin --help
```

3.8.1.2 Options

`--help` – Produce full help text

`-c` – Clobber result files from any previous run before starting this run

`-f hostfile` – File with hosts in cluster, default is `/etc/sysconfig/iba/hosts`

`-r release` – IntelIB or InfiniServ release to load/upgrade to, default is version of Intel® True Scale Fabric Suite Software presently being run on the server running this command.

`-d dir` – Directory to get product release.tgz from for load/upgrade

`-T product` – InfiniServ product type to install, default is IntelIB-Basic.



Options include: InfiniServBasic, InfiniServPerf, InfiniServMgmt, InfiniServTools.

- P *packages* – InfiniServ packages to install; default is `iba ipoib mpi`.
The host allows: `ib_stack, oftools, ib_stack_dev, fastfabric, ofed_ipoib, ofed_srp, ofed_srpt, ofed_iser, ofed_iwarp, opensm, ofed_debug, iba, ibdev, ibboot, ifibre, inic, ipoib, mpi, mpidev, mpisrc, udapl, sdp, and rds`.
- S – Securely prompt for password for user on remote system
- operation* – Operation to perform. Can be one or more of:
 - `load` – Initial install of all hosts
 - `upgrade` – Upgrade install of all hosts
 - `configipoib` – Create `ifcfg-ib1` using host IP address from `/etc/hosts`
 - `reboot` – Reboot hosts, ensure they go down and come back
 - `sacache` – Confirm sacache has all hosts in it
 - `ipoibping` – Verify this host can ping each host through IPoIB
 - `mpiperf` – Verify latency and bandwidth for each host
 - `mpiperfdeviation` – Verify latency and bandwidth for each host against a defined threshold (or relative to average host performance)

3.8.1.3 Example

```
iba_host_admin -c reboot
```

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

- `test.res` – Appended with summary results of run
 - `test.log` – Appended with detailed results of run
 - `save_tmp/` – Contains a directory per failed test with detailed logs
 - `test_tmp*/` – Intermediate result files while test is running
- The `-c` option will remove all of the above.

Results from `iba_host_admin` are grouped into Test Suites, Test Cases and Test Items. A given run of `iba_host_admin` represents a single Test Suite. Within a Test Suite multiple Test Cases will occur, typically one Test Case per host being operated on. Some of the more complex operations (such as `ipoibping`) may have multiple Test Items per Test Case. Each such item represents a major step in the overall Test Case.

Each `iba_host_admin` run appends to `test.res` and `test.log` and creates temporary files in `test_tmp$PID` in the current directory. `test.res` will provide an overall summary of operations performed and their results. The same information will also be displayed while `iba_host_admin` is executing. `test.log` will contain detailed information about what was performed. This will include the specific commands executed and the resulting output. The `test_tmp` directories will contain temporary files which reflect tests in progress (or killed). The logs for any failures will be logged in the `save_tmp` directory with a directory per failed test case. If the same test case fails more than once, `save_tmp` will retain the information from the first failure and subsequent runs of `iba_host_admin` will only append to `test.log`. It is recommended to review failures and use the `-c` option to remove old logs before subsequent runs of `iba_host_admin`.

`iba_host_admin` implicitly performs its operations in parallel. Twenty (20) parallel operations is the default.



3.8.2 iba_host_admin Host Operations

(Host) It is recommended to set up password SSH or SCP for use during this operation. Alternatively, the `-s` option can be used to securely prompt for a password, in which case the same password is used for all hosts. Alternately, the password may be put in the `fastfabric.conf` file using `FF_PASSWORD` and `FF_ROOTPASS`.

3.8.2.1 load

This performs an initial installation of Fabric Access software on a group of hosts. Any existing Fabric Access installation will first be uninstalled and any Fabric Access configuration files will be removed. Therefore, the hosts will end up installed with a default Fabric Access configuration. The default install packages are `iba ipoib mpi` (for example, True Scale Fabric Stack, IPoIB and MPI). The default is the typical configuration for an MPI cluster compute node. The `-r` option can be used to specify a release to install other than the one that this host is presently running. The `FF_PRODUCT.FF_PRODUCT_VERSION.tgz` file (for example, `IntelIB-Basic.DISTRO.VERSION.tgz`) is expected to exist in the directory specified by `-d` (the default is the current working directory) and will be copied to all the selected hosts and installed.

Note: When using the present version of Intel® FastFabric Toolset for OFED+, Intel® FastFabric Toolset may be used to install OFED+ (`IntelIB-Basic.DISTRO.VERSION.tgz`) or the True Scale Fabric Stack Tools (`InfiniServTools.VERSION.tgz`) on the remaining hosts. When using Intel® FastFabric Toolset only to install InfiniServ Tools, OFED must be installed on each host manually.

3.8.2.2 upgrade

This is very similar to the `load` option, however all the selected hosts are upgraded without affecting existing Fabric Access configuration. This is comparable to the `-U` option when running `INSTALL` manually. The `-r` option can be used to upgrade to a release different from this host, the default will be to upgrade to the same release as the this host. The `FF_PRODUCT.FF_PRODUCT_VERSION.tgz` file (for example, `IntelIB-Basic.DISTRO.VERSION.tgz`) is expected to exist in the directory specified by `-d` (the default is the current working directory) and will be copied to all the end nodes and installed.

Note: Only those Fabric Access components that are currently installed will be upgraded. This operation will fail for hosts that do not have Fabric Access software installed.

Note: When using the present version of Intel® FastFabric Toolset for OFED+, Intel® FastFabric Toolset may be used to install OFED+ (`IntelIB-Basic.DISTRO.VERSION.tgz`) or the True Scale Fabric Stack Tools (`InfiniServTools.VERSION.tgz`) on the remaining hosts. When using Intel® FastFabric Toolset only to install InfiniServ Tools, OFED must be installed on each host manually.

3.8.2.3 configipoib

Creates a `ifcfg-ib1` configuration file (when running OFED+, this configures `ifcfg-ib0`) for each node using the IP address found through the resolver on the node (the standard Linux resolver is used by the `host` command). If the host is not found, `/etc/hosts` on the node is checked. The `-i` option can specify an IPoIB suffix to apply to the host name to create the IPoIB host name for the node (that will be looked up in `/etc/hosts`). The default suffix is `-ib`. IPoIB will be configured for a



static IP address and will be autostarted at boot. For the Intel® True Scale Fabric Stack, the default `/etc/sysconfig/ipoib.cfg` file will be used, which provides a redundant IPoIB configuration using both ports of the first HCA in the system.

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

3.8.2.4 reboot

This reboots the given hosts and ensures they go down and come back up by pinging them during the reboot process. The ping rate is slow (5 seconds), so if the servers boot faster than this, false failures may be seen.

3.8.2.5 sacache

This verifies the given hosts can properly communicate with the SA and any cached SA data that is up to date. To run this command, True Scale Fabric must be installed and running on the given hosts. The subnet manager and switches must be up. If this test fails, for QuickSilver hosts: `cmdall 'cat /proc/driver/ics_dsc/gids'` can be run against any problem hosts to see what they have cached. If this test fails, for OFED hosts: `cmdall 'iba_saquery -o desc'` can be run against any problem hosts to see what they see.

Note: This operation requires that the hosts being queried be specified by a resolvable TCP/IP host name. This operation will FAIL if the selected hosts are specified by IP address. See ["Selection of Hosts" on page 23](#) for more information.

3.8.2.6 ipoibping

This verifies IPoIB basic operation by ensuring that the host can ping all other nodes through IPoIB. To run this command True Scale Fabric must be installed, IPoIB must be configured and running on the host and the given hosts, the SM and switches must be up.

3.8.2.7 mpirperf

Verifies that MPI is operational and checks MPI end-to-end latency and bandwidth between pairs of nodes (for example, 1-2, 3-4, 5-6). This can be used to verify switch latency/hops, PCI bandwidth and overall MPI performance. The `test.res` file will have the results of each pair of nodes tested.

To obtain accurate results, this test should be run at a time when no other stressful applications (for example, MPI jobs, high stress file system operations) are running on the given hosts.

The following is a sample of expected MPI bandwidths for various server slot speeds:

- PCI-X 66 MHz (32 bit) - 250 MB/s or less
- PCI-X 66MHz - 400-450 MB/s or less
- PCI-X 100 MHz - 600-700 MB/s
- PCI-X 133 MHz - 800-900 MB/s
- PCIe x8 SDR HCA - 900+ MB/s
- PCIe x8 DDR HCA - 1300+ MB/s



- PCIe Gen2 x8 QDR HCA - 2400+ MB/s

Bandwidth issues typically indicate server configuration issues (for example, incorrect slot used, incorrect BIOS settings, incorrect HCA model), or fabric issues (for example, symbol errors, incorrect link width or speed). Assuming `iba_report` has previously been used to check for link errors and link speed issues, the server configuration should be verified.

Note that BIOS settings and differences between server models can account for 10-20% differences in bandwidth. The numbers above are conservative numbers representative of what most servers can achieve. Some server models may have 10-20% higher results. A result 5-10% below the above numbers is typically not cause for serious alarm, but may reflect limitations in the server design or the chosen BIOS settings.

For more details about BIOS settings, consult the documentation from the server supplier and/or the server PCI chipset manufacturer.

3.8.2.8 **mpiperfdeviation**

Is an upgraded version of **mpiperf**. It verifies that MPI is operational and checks MPI end-to-end latency and bandwidth between pairs of nodes. This can be used to verify switch latency/hops, PCI bandwidth and overall MPI performance. It performs assorted pairwise bandwidth and latency tests and reports pairs outside an acceptable tolerance range. The tool will identify specific nodes which have problems and provide a concise summary of results. The `test.res` file will have the results of each pair of nodes tested.

By default concurrent mode is used to quickly analyze the fabric and host performance. Pairs which have 20% less bandwidth or 50% more latency than the average pair will be reported as failures.

In concurrent mode, hosts are paired up and all pairs are run concurrently. Since there may be fabric contention during such as run, any poor performing pairs are then rerun sequentially against the reference host.

Concurrent mode will run the tests in the shortest amount of time, however the results could be slightly less accurate due to switch contention.

Note: This option is available for the IntelIB packaging of OFED, but is not presently available for other packagings of OFED.

To obtain accurate results, this test should be run at a time when no other stressful applications (for example, MPI jobs, high stress file system operations) are running on the given hosts.

Bandwidth issues typically indicate server configuration issues (for example, incorrect slot used, incorrect BIOS settings, incorrect HCA model), or fabric issues (for example, symbol errors, incorrect link width or speed). Assuming `iba_report` has previously been used to check for link errors and link speed issues, the server configuration should be verified.

Note that BIOS settings and differences between server models can account for 10-20% differences in bandwidth. A result 5-10% below the average is typically not cause for serious alarm, but may reflect limitations in the server design or the chosen BIOS settings.

For more details about BIOS settings, consult the documentation from the server supplier and/or the server PCI chipset manufacturer.



3.8.3 Interpreting the `iba_host_admin`, `iba_chassis_admin` and `iba_switch_admin` log files

Each run of `iba_host_admin`, `iba_chassis_admin` and `iba_switch_admin` will create `test.log` and `test.res` files in the current directory.

When `iba_host_admin`, `iba_chassis_admin` and `iba_switch_admin` indicates that some or all of the test cases failed, the `test.res` and `test.log` files should be reviewed. `test.res` will summarize which tests have failed. Using the `test.res` file for servers that failed can be quickly identified. If the problem is not immediately obvious, check the `test.log` file. The most recent results will be at the end of the file. The `save_tmp/*/test.log` files will be easier to read since they will represent the logs for a single test case, typically against a single chassis, switch or host.

The keyword `FAILURE` will be used to mark any failures. Typically due to the roll up of error messages, the first instance of `FAILURE` in a given sequence of failures will show what was being done. Proceeding the `FAILURE`, the log will also show the exact sequence of commands issued to the target host and/or chassis and the resulting output from that host and/or chassis.

For example, `test.log` may contain lines such as:

```
scp ./InfiniServPerf.4.1.1.0.15.tgz root@n001a:

TEST CASE FAILURE=scp ./InfiniServPerf.4.1.1.0.15.tgz root@n001a: failed: ssh:
n001a Name or service not known

lost connection
```

This indicates the `scp` command shown was executed but failed with the error message:

```
"ssh: n001a Name or service not known.

lost connection"
```

In this example, this was the exact output from SSH.

If there is a `FAILURE` message indicating time-out, it means the expected output did not occur within a reasonable time limit. The time limits used are quite generous, so such failures often indicate a host, chassis or switch is offline. It could also indicate unexpected prompts (such as a password prompt when password-less ssh is expected). Review the `test.log` first for such prompts. Also verify that the host can SSH to the target host or chassis with the expected password behavior.

Another common source of time-outs is incorrect host shell command prompts. Verify that both this host and the target host have their prompts set correctly. The command line prompt must end in `#` or `$` (make certain there is a space after either).

Yet another common source of time-outs is typographical errors in selected host or chassis names. Verify that the host, chassis or switch names in `test.log` match the intended host names. Also make sure that when IPoIB host names are used, that the correct name was formed. The default is `-ib` as specified by the `FF_IPoIB_SUFFIX` parameter in the `fastfabric.conf` file.

3.9 Health Check and Baselining Tools

(All) These tools help to rapidly identify if the fabric has a problem or if its configuration has changed since the last baseline. Analysis includes hardware, software, fabric topology and SM configuration. The tools are designed to permit easy manual execution or automated execution using `cron` or other mechanisms.

These tools consist of 5 commands:

`all_analysis` – performs selected set of the below 4 analysis commands. This command is recommended as the primary tool for general analysis.

When its desired to restrict the analysis to a specific subset of components, use one of the commands below.

`fabric_analysis` – performs fabric topology and PMA error counters analysis.

`chassis_analysis` – performs Intel® Chassis configuration and health analysis for selected chassis.

`esm_analysis` – performs embedded SM configuration and health analysis for selected chassis.

`hostsm_analysis` – performs host SM configuration and health analysis for the local host.

3.9.1 Usage Model

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

- Perform initial fabric install and verification
 - Optionally run tools in “health check only” mode
 - Performs quick health check
 - Duplicates some of steps already done during verification
- Run tools in “baseline” mode
 - Takes a baseline of present HW/SW/config
- Periodically run tools in “check” mode
 - Performs quick health check
 - Compares present HW/SW/config to baseline
 - Can be scheduled in hourly cron jobs
- As needed rerun “baseline” when expected changes occur
 - Fabric upgrades
 - Hardware replacements/changes
 - Software configuration changes
 - Etc.

3.9.2 Common Operations and Options

The Health Check and Baselining tools support the following options to select the operations to be done by the tool:

- b – perform a baseline snapshot of the configuration
- e – perform an error check/health analysis only

If neither option is specified, the tool performs a snapshot of the present configuration, compares it to the baseline and also perform an error check/health analysis.



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

The typical use of the tools is to perform an initial error check by running the `-e` option. Review the errors reported in the files indicated by the tools. Once all the errors are corrected, perform a baseline of the configuration using the `-b` option. The baseline configuration will be saved to files in `FF_ANALYSIS_DIR/baseline` (the default of `/var/opt/iba/analysis/baseline` is set through `/etc/sysconfig/fastfabric.conf`). This baseline configuration should be carefully reviewed to make sure it matches the intended configuration of the cluster. If it does not, the cluster should be corrected and a new baseline run.

3.9.2.1 For example

```
fabric_analysis -e
```

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

```
fabric_analysis -b
```

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

Note: Alternatively, the advanced topology verification capabilities of `iba_report` can be used to verify the fabric deployment against the intended design

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

3.9.2.2 For example

```
fabric_analysis
```

This command will check the present fabric links and components against the previous baseline. If there have been changes, it will report a failure and indicate which files hold the resulting snapshot and differences. It will also check the PMA error counters and link speeds for the fabric (similar to `fabric_analysis -e`). If either of these checks fail, it will return a non-zero exit status, therefore permitting higher level scripts to detect a failed condition.

The differences files are generated using the Linux command specified by `FF_DIFF_CMD` in `fastfabric.conf`. By default this is the `diff -C 1` command. It is run against the baseline and new snapshot. Therefore, lines after each `*** #, #` heading in the `diff` are from the baseline and lines after each `--- #, #` heading are from the new snapshot. If `FF_DIFF_CMD` is simply set to `diff`, lines indicated by `"<"` in the `diff` would be from the baseline and lines indicated by `">"` in the `diff` would be from the new snapshot. Another command which can be useful is the Linux `sdiff` command. For more information about the `diff` output format, consult the Linux man page for `diff`.

If the configuration is intentionally changed, a new error analysis and baseline should be obtained using the same sequence as for the initial installation (discussed above), establishing a new baseline for future comparisons.

In addition all of the tools support the following option:



`-s` – save history of failures.

When the `-s` option is used, each failed run will also create a directory (whose name is the date/time the analysis tool was started) containing the failing snapshot information and `diffs`. This will permit a history of failures to be tracked. Note that every run of the tools also creates a `latest` directory with the latest snapshot. The latest files are overwritten by each subsequent run of the tool, which means the most recent run results are always available.

Beware, frequent use of the health check tools in conjunction with `-s` can consume a large amount of disk space. The space requirements will depend greatly on the size of the cluster, for example, it could be greater than 10 megabytes per run on a 1000 node cluster.

The `FF_ANALYSIS_DIR` option in `fastfabric.conf` can be changed to provide a customer specific alternate directory which will be used to hold the baseline, snapshot, and history directory trees. Under `FF_ANALYSIS_DIR` subdirectories will be created as follows:

- `baseline` – baseline snapshot from each analysis tool.
- `latest` – latest snapshot from each analysis tool.
- `YYYY-MM-DD-HH:MM:SS` – failed analysis from analysis run with `-s`. Actual directory name will have actual date/time as the name.

3.9.3 fabric_analysis

(All) The `fabric_analysis` command performs analysis of the fabric.

3.9.3.1 Usage

```
fabric_analysis [-b|-e] [-s]
```

3.9.3.2 Options

- `-b` – Baseline mode, default is compare/check mode.
- `-e` – Evaluate health only, default is compare/check mode.
- `-s` – Save history of failures (errors/differences).

3.9.3.3 Example

```
fabric_analysis
```

The fabric analysis tool checks the following:

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

Note that the comparison includes components on the fabric. Therefore operations such as shutting down a server will cause the server to no longer appear on the fabric and will be flagged as a fabric change or failure by `fabric_analysis`.

By default the error analysis includes PMA counters and slow links (for example, links running below enabled speeds). However this can be changed through the `FF_FABRIC_HEALTH` configuration parameter in `fastfabric.conf`. See *Appendix A* in the *Intel® True Scale Fabric Suite FastFabric User Guide* for more information). This



parameter specifies the `iba_report` options and reports to be used for the health analysis. It also can specify the PMA counter clearing behavior (`-i seconds`, `-C`, or none at all).

The thresholds for PMA counter analysis default to `/etc/sysconfig/iba/iba_mon.conf`.

All files generated by `fabric_analysis` will start with `fabric` in their file name. This is followed by `0:0` identifying the use of the first active True Scale Fabric port on the local server.

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

3.9.3.4 Health Check

`latest/fabric.0:0.errors` - stdout of `iba_report` for errors encountered during fabric error analysis.

`latest/fabric.0:0.errors.stderr` - stderr of `iba_report` during fabric error analysis.

3.9.3.5 Baseline

`baseline/fabric.0:0.snapshot.xml` - `iba_report` snapshot of complete fabric components and SMA configuration.

`baseline/fabric.0:0.comps` - `iba_report` summary of fabric components and basic SMA configuration.

`baseline/fabric.0:0.links` - `iba_report` summary of internal and external links.

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

3.9.3.6 Full analysis

`latest/fabric.0:0.snapshot.xml` - `iba_report` snapshot of complete fabric components and SMA configuration.

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

`latest/fabric.0:0.errors` - stdout of `iba_report` for errors encountered during fabric error analysis.

`latest/fabric.0:0.errors.stderr` - stderr of `iba_report` during fabric error analysis.

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

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

`latest/fabric.0:0.comps.diff` - diff of baseline and latest fabric components.

`latest/fabric.0:0.links` - stdout of `iba_report` summary of internal and external links.



`latest/fabric.0:0.links.stderr` - stderr of `iba_report` summary of internal and external links.

`latest/fabric.0:0.links.diff` - diff of baseline and latest fabric internal and external links.

`latest/fabric.0:0.links.changes.stderr` - stderr of `iba_report` comparison of links.

`latest/fabric.0:0.links.changes` - `iba_report` comparison of links against baseline, this is typically easier to read than the `links.diff` file and will contain the same information.

`latest/fabric.0:0.comps.changes.stderr` - stderr of `iba_report` comparison of components.

`latest/fabric.0:0.comps.changes` - `iba_report` comparison of components against baseline, this is typically easier to read than the `comps.diff` file and will contain the same information.

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

If the `-s` option is used and failures are detected, files related to the checks that failed are also copied to the timestamped directory name under `FF_ANALYSIS_DIR`, such as:

`FF_ANALYSIS_DIR/2007-11-22-09:53:04`

3.9.3.7 True Scale Fabric items checked against the baseline

Based on `iba_report -o links`:

- Unconnected/down/missing cables
- Added/moved cables
- Changes in link width and speed
- Changes to Node GUIDs in fabric (replacement of HCA or Switch hardware)
- Adding/Removing True Scale Fabric Nodes (CA, Virtual CAs, Virtual Switches, Physical Switches, Physical Switch internal switching cards (leaf/spine))
- Changes to server or switch names

Based on `iba_report -o comps`:

- Overlap with items above from links report
- Changes in port MTU, LMC, number of VLS
- Changes in port speed/width enabled or supported
- Changes in HCA or switch device IDs/revisions/VendorID (for example, ASIC HW changes)
- Changes in port Capability mask (which True Scale Fabric features/agents run on port/server)
- Changes to ErrorLimits and PKey enforcement per port
- Changes to IOUs/IOCs/IOC Services provided

Only applicable if IOUs in fabric (9000 series Virtual IO cards, True Scale Fabric native storage, etc)

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



- Includes primary and backups
- Includes configured priority for SM

3.9.3.8 True Scale Fabric Items that are also checked during health check

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

- PMA error counters on all True Scale Fabric ports (HCA, switch external and switch internal) checked against configurable thresholds.
 - Counters are cleared each time a healthcheck is run, each healthcheck reflects a counter delta since last healthcheck.
 - Typically identifies potential fabric errors (symbol errors, etc).
 - May also identify transient congestion (depends upon counters monitored).
- Link active speed/width as compared to Enabled speed.
 - Identifies links whose active speed/width is < min (enabled speed/width on each side of link).
 - This typically reflects bad cables or bad ports or poor connections.
- Side effect is the verification of SA health.

3.9.4 chassis_analysis

(Switch) The `chassis_analysis` command performs analysis of the chassis.

3.9.4.1 Usage

```
chassis_analysis [-b|-e] [-s] [-F chassisfile]
```

3.9.4.2 Options

- b – Baseline mode. The default is the compare/check mode.
- e – Evaluate health only, default is the compare/check mode.
- s – A save history of failures (errors/differences).
- F *chassisfile* – The file with the chassis in the cluster. The default is `/etc/sysconfig/iba/chassis`.

3.9.4.3 Example

```
chassis_analysis
```

The chassis analysis tool checks the following for Intel® Chassis:

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

Setup of ssh keys for chassis (see ["setup_ssh" on page 51](#)) is recommended. If ssh keys are not setup, all chassis must be configured with the same admin password and the password must be kept in the `fastfabric.conf` configuration file.

The default set of `FF_CHASSIS_CMDS` is:



```
showInventory fwVersion showIBNodeDesc ismShowPStatThresh ismChassisSet12x  
timeZoneConf timeDSTConf snmpCommunityConf snmpTargetAddr showChassisIpAddr  
showDefaultRoute
```

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

`FF_CHASSIS_HEALTH` can specify one command (with arguments) to be used to check the chassis health. For chassis with newer firmware, the `hwCheck` command is recommended. For chassis with older firmware a benign command, such as `fruInfo`, should be used. The default is `hwCheck`. Note that only the exit status of the `FF_CHASSIS_HEALTH` command is checked. The output is not captured and compared in a snapshot. However, on failure its output is saved to aid diagnosis.

The `chassis_analysis` tool performs its analysis against one or more chassis in the fabric. As such, it permits the chassis to be specified using the `-F` option with a default specified by the `CHASSIS_FILE` parameter in `fastfabric.conf`. The handling of these options and settings is comparable to `cmdall -C` and similar Intel® FastFabric Toolset commands against a chassis.

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

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

3.9.4.4 Health Check

`latest/chassis.hwCheck` – Output of `hwCheck` command for all selected chassis

3.9.4.5 Baseline

`baseline/chassis.fwVersion` – The output of `fwVersion` command for all selected chassis.

`baseline/chassis.ismChassisSet12x` – The output of the `ismChassisSet12x` command for all selected chassis.

`baseline/chassis.ismShowPStatThresh` – The output of the `ismShowPStatThresh` command for all selected chassis.

`baseline/chassis.showChassisIpAddr` – The output of the `showChassisIpAddr` command for all selected chassis.

`baseline/chassis.showDefaultRoute` – The output of the `showDefaultRoute` command for all selected chassis.

`baseline/chassis.showIBNodeDesc` – The output of the `showIBNodeDesc` command for all selected chassis.

`baseline/chassis.showInventory` – The output of the `showInventory` command for all selected chassis.

`baseline/chassis.snmpCommunityConf` – The output of the `snmpCommunityConf` command for all selected chassis.

`baseline/chassis.snmpTargetAddr` – The output of the `snmpTargetAddr` command for all selected chassis.



`baseline/chassis.timeDSTConf` – The output of the `timeDSTConf` command for all selected chassis.

`baseline/chassis.timeZoneConf` – The output of the `timeZoneConf` command for all selected chassis.

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

3.9.4.6 Full analysis

`latest/chassis.hwCheck` – The output of the `hwCheck` command for all selected chassis.

`latest/chassis.fwVersion` – The output of the `fwVersion` command for all selected chassis.

`latest/chassis.fwVersion.diff` – The diff of the baseline and latest `fwVersion`.

`latest/chassis.ismChassisSet12x` – The output of the `ismChassisSet12x` command for all selected chassis.

`latest/chassis.ismChassisSet12x.diff` – The diff of the baseline and latest `ismChassisSet12x`.

`latest/chassis.ismShowPStatThresh` – The output of the `ismShowPStatThresh` command for all selected chassis.

`latest/chassis.ismShowPStatThresh.diff` – The diff of baseline and latest `ismShowPStatThresh`.

`latest/chassis.showChassisIpAddr` – The output of the `showChassisIpAddr` command for all selected chassis.

`latest/chassis.showChassisIpAddr.diff` – The diff of baseline and latest `showChassisIpAddr`.

`latest/chassis.showDefaultRoute` – The output of the `showDefaultRoute` command for all selected chassis.

`latest/chassis.showDefaultRoute.diff` – The diff of the baseline and the latest `showDefaultRoute`.

`latest/chassis.showIBNodeDesc` – The output of the `showIBNodeDesc` command for all selected chassis.

`latest/chassis.showIBNodeDesc.diff` – The diff of the baseline and latest `showIBNodeDesc`.

`latest/chassis.showInventory` – The output of the `showInventory` command for all selected chassis.

`latest/chassis.showInventory.diff` – The diff of the baseline and latest `showInventory`.

`latest/chassis.snmpCommunityConf` – The output of the `snmpCommunityConf` command for all selected chassis.

`latest/chassis.snmpCommunityConf.diff` – The diff of the baseline and latest `snmpCommunityConf`.



`latest/chassis.snmpTargetAddr` – The output of the `snmpTargetAddr` command for all selected chassis.

`latest/chassis.snmpTargetAddr.diff` – The diff of the baseline and latest `snmpTargetAddr`.

`latest/chassis.timeDSTConf` – The output of the `timeDSTConf` command for all selected chassis.

`latest/chassis.timeDSTConf.diff` – The diff of the baseline and latest `timeDSTConf`.

`latest/chassis.timeZoneConf` – The output of the `timeZoneConf` command for all selected chassis.

`latest/chassis.timeZoneConf.diff` – The diff of the baseline and latest `timeZoneConf`.

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

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

`FF_ANALYSIS_DIR/2007-11-22-09:53:04`

3.9.4.7 Chassis items checked against the baseline

Based upon `showInventory`:

- Addition/removal of Chassis FRUs
 - Replacement is only checked for FRUs that `showInventory` displays the serial number. For the 9000 series, the fan and power supply replacement is not checked, just present.
- Removal of redundant FRUs (spines, power supply, fan)

Based upon `fwVersion`:

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

Based upon `showIBNodeDesc`:

- Changes to configured IB Node Description for chassis. Note changes detected here would also be detected in fabric level analysis

Based upon `ismShowPStatThresh`:

- Changes to configured port thresholds for chassis port error thresholding

Based upon `ismChassisSet12x`:

- Changes to chassis link width controls. Note that changes detected here would also be detected in fabric level analysis.

Based upon `timeZoneConf` and `timeDSTConf`:

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

Based upon `snmpCommunityConf` and `snmpTargetAddr`:

- Changes to SNMP persistent configuration within the chassis

The following Chassis items will not be checked against baseline:



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

3.9.4.8 Chassis Items also checked during healthcheck

Based upon `hwCheck`:

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

3.9.5 hostsm_analysis

(All) The `hostsm_analysis` command performs analysis against the local server only.

3.9.5.1 Usage

```
hostsm_analysis [-b|-e] [-s]
```

3.9.5.2 Options

- b – Baseline mode. The default is the compare/check mode.
- e – Evaluate health only. The default is the compare/check mode.
- s – Save history of failures (for example, errors/differences).

3.9.5.3 Example

```
hostsm_analysis
```

The host SM analysis tool checks the following:

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

The `hostsm_analysis` tool performs analysis against the local server only. It is assumed that both the host SM and Intel® FastFabric Toolset are installed on the same system.

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

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



3.9.5.4 Health Check

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

3.9.5.5 Baseline

baseline/hostsm.smver – Host SM version.

baseline/hostsm.smconfig – A copy of `ifs_fm.config`.

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

3.9.5.6 Full analysis

latest/hostsm.smstatus – The output of the `sm_query smShowStatus` command.

latest/hostsm.smver – The host SM version.

latest/hostsm.smver.diff – The diff of the baseline and latest host SM version.

latest/hostsm.smconfig – A copy of `ifs_fm.config`.

latest/hostsm.smconfig.diff – The diff of the baseline and the latest `ifs_fm.config`.

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

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

`FF_ANALYSIS_DIR/2007-11-22-09:53:04`

3.9.5.7 Host SM items checked against the baseline

- SM configuration file
- The version of the SM rpm installed on the system

3.9.5.8 Host SM items also checked during healthcheck

The SM is in the running state

3.9.6 esm_analysis

(Switch) The `esm_analysis` command performs analysis of the embedded SM for configuration, and health.

3.9.6.1 Usage

```
esm_analysis [-b|-e] [-s] [-G esmchassisfile]
```

3.9.6.2 Options

- b – Baseline mode. The default is the compare/check mode.
- e – Evaluate health only. The default is the compare/check mode.
- s – Save history of failures (for example, errors/differences).



`-G esmchassisfile` – The file with SM chassis within the cluster. The default is `/etc/sysconfig/iba/esm_chassis`.

3.9.6.3 Example

`esm_analysis`

The embedded SM analysis tool checks the following:

- Embedded SM configuration (as reported by the chassis commands specified in `FF_ESM_CMDS` in `fastfabric.conf`).
- Embedded SM health (as reported by `smControl status`).
- For Intel® 12000 Chassis, the `ifs_fm.xml` file for the chassis is also checked

Setup of ssh keys for chassis (see “[setup_ssh](#)” on page 51) is recommended. If ssh keys are not setup, all chassis must be configured with the same `admin` password and the password must be kept in the `fastfabric.conf` configuration file.

The default set of `FF_ESM_CMDS` is:

```
smShowSMParms smShowDefBcGroup
```

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

The `esm_analysis` command performs analysis against one or more chassis in the fabric. As such it permits a chassis to be specified using the `-G`, with a default specified by the `ESM_CHASSIS_FILE` parameter in `fastfabric.conf`. The handling of these options and settings is comparable to `cmdall -C` and similar Intel® FastFabric Toolset commands against a chassis. The exception in this case is that the option and variable names are slightly different to distinguish the fact that they are specifying only the chassis that has an embedded SM running).

All files generated by `esm_analysis` start with `esm` within the file name.

The `esm_analysis` command generates files such as the following within `FF_ANALYSIS_DIR`. The actual file names reflect the individual chassis commands that have been configured through the `FF_ESM_CMDS` parameter:

3.9.6.4 Health Check

`latest/esm.smstatus` – The output of the `smControl status` command for all selected chassis.

3.9.6.5 Baseline

`baseline/esm.smShowDefBcGroup` – The output of the `smShowDefBcGroup` command for all selected chassis.

`baseline/esm.smShowSMParms` – The output of the `smShowSMParms` command for all selected chassis. `latest/esm.CHASSIS.ifs_fm.xml` – The `ifs_fm.xml` file for the given chassis

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



3.9.6.6 Full analysis

`latest/esm.smstatus` – The output of the `smControl status` command for all selected chassis.

`latest/esm.smShowDefBcGroup` – The output of the `smShowDefBcGroup` command for all selected chassis.

`latest/esm.smShowDefBcGroup.diff` – The diff of baseline and latest `smShowDefBcGroup`.

`latest/esm.smShowSMParms` – The output of the `smShowSMParms` command for all selected chassis

`latest/esm.smShowSMParms.diff` – The diff of the baseline and the latest `smShowSMParms`.

`latest/esm.CHASSIS.ifs_fm.xml` – The `ifs_fm.xml` file for the given chassis

`latest/esm.CHASSIS.ifs_fm.xml.diff` – The diff of the baseline and the latest `ifs_fm.xml` for the given chassis.

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

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

`FF_ANALYSIS_DIR/YYYY-MM-DD-hh:mm:ss`

3.9.6.7 Chassis SM items that are checked against the baseline

Based upon `smShowSMParms`:

- SM priority
- SM sweep rate
- SM retry and time-out
- SM fabric time-outs configured (`switchLifeTime`, `HoqLife`, `VLStall`, `PacketLifeTimes` for `PathRecords`)
- Multipath mode
 - Based on `smShowDefBcGroup`
- Default IPoIB broadcast group settings in SM (`PKey`, `MTU`, `Rate`, `SL`)
- For Intel® 12000 Chassis, the entire `ifs_fm.xml` file is also compared.

3.9.6.8 Chassis SM items also checked during healthcheck

Based upon `smControl status`:

- SM is in running state

3.9.7 all_analysis

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

3.9.7.1 Usage

```
all_analysis [-b|-e] [-s] [-F chassisfile] [-G esmchassisfile]
```



3.9.7.2 Options

- b – Baseline mode. The default is the compare/check mode.
- e – Evaluate health only. The default is the compare/check mode.
- s – A save history of failures (for example, errors and differences).
- F *chassisfile* – A file with a chassis in a cluster. The default is /etc/sysconfig/iba/chassis.
- G *esmchassisfile* – A file with the SM chassis in the cluster. The default is /etc/sysconfig/iba/esm_chassis.

3.9.7.3 Example

`all_analysis`

The `all_analysis` command will perform the set of analysis specified in `FF_ALL_ANALYSIS`. This can be provided through the environment or using `fastfabric.conf`. The set of analysis which can be specified are: `fabric`, `chassis`, `esm` or `hostsm`. `FF_ALL_ANALYSIS` must be a space-separated list of the values mentioned above. These correspond to the respective analysis commands previously discussed.

Note that the `all_analysis` command has options which are a superset of the options for all other analysis commands. The options will be passed along to the respective tools (for example, the `-F chassisfile` option will be passed on to `chassis_analysis` if it is specified in `FF_ALL_ANALYSIS`).

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

3.9.7.4 Manual and Automated Usage

There are two basic ways to use the tools:

- Manual
- Automated

In both cases the user should follow the initial setup procedure outlined above to create a good baseline of the configuration.

In the manual method, the user would run the tools manually when trying to diagnose problems, or when there is a concern or need to validate the configuration and health.

In the automated method, the user could run `all_analysis` or a specific tool in an automated script (such as a `cron` job). When run in this mode the `-s` option may prove useful (but care must be taken to avoid excessive saved failures). When run in automated mode, a frequency of no faster than hourly would be recommended. For many fabrics a run daily or perhaps every few hours would be sufficient. Since the exit code from each of the tools indicates the overall success/failure, an automated script could easily check the exit status and on failure e-mail the output from the analysis tool to the appropriate administrators for further analysis and corrective action as needed.

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

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



- Runs with the `-s` option will consume additional disk space for each run that identifies an error. The amount of disk space will vary depending on fabric size. For a larger fabric this can be on the order of 1-40 MB. Therefore, care must be taken not to run the tools too often and to visit and clean out the `FF_ANALYSIS_DIR` periodically. If the `-s` option is used during automated execution of the health check tools, it may be helpful to also schedule automated disk space checks (for example, as a cron job).
- Runs coinciding with down time for selected components (such as servers that are offline or rebooting) will be considered failures and generate the resulting failure information. If the runs are not carefully scheduled, this could be misleading and also waste disk space.

3.9.8 Re-establishing Health Check baseline

This is needed after changing the fabric configuration in any way. The following activities are examples of ways in which the fabric configuration may be changed:

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

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

1. Make sure that you have fixed all problems with the fabric, including inadvertent configuration changes, before proceeding.
2. Verify that the fabric configured is as expected. The simplest way to do this is to run `fabric_info`. This will return information for each subnet to which the fabric management server is connected. The following is an example output for a single subnet. The comments are not part of the output. They are only included to help understand the output better.

```
SM: c999f4nm02 HCA-2 Guid: 0x0008f104039908e5 State: Master
```

```
Number of CAs: 53 # one for each HCA port; including the Fabric/MS
```

```
Number of CA Ports: 53 # same as number of CAs
```

```
Number of Switch Chips: 76 # one per IBM GX HCA port + one per switch leaf + two per switch spine
```

```
Number of Links: 249 # one per HCA port + 12 per leaf
```

```
Number of 1x Ports: 0
```

3. Save the old baseline. This may be required for future debug. The old baseline is a group of files in `/var/opt/iba/analysis/baseline`.
4. Run `all_analysis -b`
5. Check the new output files in `/var/opt/iba/analysis/baseline` to verify that the configuration is as you expect it. Refer to the *Intel® True Scale Fabric Suite FastFabric User Guide* for details.



3.9.9 Interpreting the Health Check Results

When any of the health check tools are run, the overall success or failure will be indicated in the output of the tool and its exit status. The tool will also indicate which areas had problems and which files should be reviewed. The results from the latest run can be found in `FF_ANALYSIS_DIR/latest/`. Many files can be found in this directory which indicate both the latest configuration of the fabric and errors/differences found during the health check. Should the health check fail, the following paragraphs will discuss a recommended order for reviewing these files.

If the `-s` option was used when running the health check, a directory whose name is the date and time of the failing run will be created under `FF_ANALYSIS_DIR`. In which case that directory can be consulted instead of the `latest` directory shown in the examples below.

It is recommended to first review the results for any `esm` or `hostsm` health check failures. If the SM is misconfigured or not running, it can cause other health checks to fail. In which case the SM problems should be corrected first then the health check should be rerun and other problems should then be reviewed and corrected as needed.

For a `hostsm` analysis, the files should be reviewed in the following order:

1. `latest/hostsm.smstatus` – Make sure this indicates the SM is running. If no SMs are running on the fabric, that problem should be corrected before proceeding further. Once corrected the health checks should be rerun to look for further errors.
2. `latest/hostsm.smver.diff` – This indicates the SM version has changed. If this was not an expected change, the SM should be corrected before proceeding further. Once corrected the health checks should be rerun to look for further errors. If the change was expected and permanent, a baseline should be rerun once all other health check errors have been corrected.
3. `latest/hostsm.smconfig.diff` – This indicates that the SM configuration has changed. This file should be reviewed and as necessary the `latest/hostsm.smconfig` file should be compared to `baseline/hostsm.smconfig`. As necessary correct the SM configuration. Once corrected the health checks should be rerun to look for further errors. If the change was expected and permanent, a baseline should be rerun once all other health check errors have been corrected.

For an `esm` analysis, the `FF_ESM_CMDS` configuration setting will select which ESM commands are used for the analysis. When using the default setting for this parameter, the files should be reviewed in the following order:

1. `latest/esm.smstatus` – Make sure this indicates the SM is running. If no SMs are running on the fabric, that problem should be corrected before proceeding further. Once corrected the health checks should be rerun to look for further errors.
2. `latest/esm.CHASSIS.ifs_fm.xml` – The `ifs_fm.xml` file for the given chassis
3. `latest/esm.CHASSIS.ifs_fm.xml.diff` – This indicated that the SM configuration has changed. This file should be reviewed and as necessary the `latest/esm.CHASSIS.ifs_fm.xml` file should be compared to `baseline/esm.CHASSIS.ifs_fm.xml`. As necessary correct the SM configuration. Once corrected the health checks should be rerun to look for further errors. If the change was expected and permanent, a baseline should be rerun once all other health check errors have been corrected.
4. `latest/esm.smShowSMParms.diff` – This indicates that the SM configuration has changed. This file should be reviewed and as necessary the `latest/esm.smShowSMParms` file should be compared to `baseline/esm.smShowSMParms`. As necessary correct the SM configuration. Once corrected the health checks should be rerun to look for further errors. If the



change was expected and permanent, a baseline should be rerun once all other health check errors have been corrected.

5. `latest/esm.smShowDefBcGroup.diff` – This indicates that the SM broadcast group for IPoIB configuration has changed. This file should be reviewed and as necessary the `latest/esm.smShowDefBcGroup` file should be compared to `baseline/esm.smShowDefBcGroup`. As necessary correct the SM configuration. Once corrected the health checks should be rerun to look for further errors. If the change was expected and permanent, a baseline should be rerun once all other health check errors have been corrected.
6. `latest/esm.*.diff` – If `FF_ESM_CMDS` has been modified, the changes in results for those additional commands should be reviewed. As necessary correct the SM. Once corrected the health checks should be rerun to look for further errors. If the change was expected and permanent, a baseline should be rerun once all other health check errors have been corrected.

Next, it is recommended to review the results of the `fabric analysis` for each configured fabric. If nodes or links are missing, the `fabric analysis` will detect them. Missing links or nodes can cause other health checks to fail. If such failures are expected (for example, a node or switch is offline), further review of result files can be performed, but the user must beware that the loss of the node or link can cause other analysis to also fail. The discussion below presents the analysis order for `fabric.0.0`, if other or additional fabrics are configured for analysis, it is recommended to review the files in the order shown below for each fabric. There is no specific order recommended for which fabric to review first.

1. `latest/fabric.0.0.errors.stderr` – If this file is not empty, it can indicate problems with `iba_report` (such as inability to access an SM) which can result in unexpected problems or inaccuracies in the related `errors` file. If possible problems reported in this file should be corrected first. Once corrected the health checks should be rerun to look for further errors.
2. `latest/fabric.0:0.errors` – If any links with excessive error rates or incorrect link speeds are reported, they should be corrected. If there are links with errors, beware the same links may also be detected in other reports such as the `links` and `comps` files discussed below.
3. `latest/fabric.0.0.snapshot.stderr` – If this file is not empty, it can indicate problems with `iba_report` (such as inability to access an SM) which can result in unexpected problems or inaccuracies in the related `links` and `comps` files. If possible, problems reported in this file should be corrected first. Once corrected the health checks should be rerun to look for further errors.
4. `latest/fabric.0:0.links.stderr` and `latest/fabric.0:0.links.changes.stderr` – If these files are not empty, it can indicate problems with `iba_report` which can result in unexpected problems or inaccuracies in the related `links` files. If possible, problems reported in these files should be corrected first. Once corrected the health checks should be rerun to look for further errors. For more information on `.changes` files refer to [“Interpreting Health Check .changes Files” on page 131](#).
5. `latest/fabric.0:0.links.diff` and `latest/fabric.0:0.links.changes` – These indicate that the links between components in the fabric have changed, been removed/added or that components in the fabric have disappeared. If both files are available, the `fabric.0:0.links.changes` file should be used since it will have a more concise and precise description of the fabric link changes. As necessary the `latest/fabric.0:0.links` file should be compared to `baseline/fabric.0:0.links`. If components have disappeared, review of the `latest/fabric.0:0.comps.diff` and `latest/fabric.0:0.comps.changes` files may be easier for such components. As necessary correct missing nodes and links. Once corrected the health checks should be rerun to look for further errors. If the change was expected and is permanent, a baseline should be rerun once all



other health check errors have been corrected. For more information on `.changes` files refer to ["Interpreting Health Check .changes Files" on page 131](#).

6. `latest/fabric.0:0.comps.stderr` and `latest/fabric.0:0.comps.changes.stderr` – If these files are not empty, it can indicate problems with `iba_report` which can result in unexpected problems or inaccuracies in the related `comps` file. If possible, problems reported in these files should be corrected first. Once corrected the health checks should be rerun to look for further errors. For more information on `.changes` files refer to ["Interpreting Health Check .changes Files" on page 131](#).
7. `latest/fabric.0:0.comps.diff` and `latest/fabric.0:0.comps.changes` – These indicate that the components in the fabric or their SMA configuration have changed. If both files are available, the `fabric.0:0.comps.changes` file should be used since it will have a more concise and precise description of the fabric component changes. As necessary the `latest/fabric.0:0.comps` file should be compared to `baseline/fabric.0:0.comps`. As necessary correct missing nodes, missing SMs, ports which are down and port misconfigurations. Once corrected the health checks should be rerun to look for further errors. If the change was expected and permanent, a baseline should be rerun once all other health check errors have been corrected. For more information on `.changes` files refer to ["Interpreting Health Check .changes Files" on page 131](#).

Finally, it is recommended to review the results of the `chassis_analysis`. If chassis configuration has changed, the `chassis_analysis` will detect it. Previous checks should have already detected missing chassis, missing or added links and many aspects of chassis configuration. For `chassis_analysis`, the `FF_CHASSIS_CMDS` and `FF_CHASSIS_HEALTH` configuration settings will select which chassis commands are used for the analysis. When using the default setting for this parameter, the files should be reviewed in the following order:

1. `latest/chassis.hwCheck` – Make sure this indicates all chassis are operating properly with the desired power and cooling redundancy. If there are problems, they should be corrected, but other analysis files can be analyzed first. Once any problems are corrected, the health checks should be rerun to verify the correction.
2. `latest/chassis.fwVersion.diff` – This indicates the chassis firmware version has changed. If this was not an expected change, the chassis firmware should be corrected before proceeding further. Once corrected the health checks should be rerun to look for further errors. If the change was expected and permanent, a baseline should be rerun once all other health check errors have been corrected.
3. `latest/chassis.*.diff` – These files reflect other changes to chassis configuration based on checks selected through `FF_CHASSIS_CMDS`. The changes in results for these remaining commands should be reviewed. As necessary correct the chassis. Once corrected the health checks should be rerun to look for further errors. If the change was expected and permanent, a baseline should be rerun once all other health check errors have been corrected.

If any health checks failed, after correcting the related issues, another health check should be run to verify the issues were all corrected. If the failures are due to expected and permanent changes, once all other errors have been corrected, a baseline should be rerun.

3.9.10 Interpreting Health Check .changes Files

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

The format is like the following:



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

In the following example of `fabric.*:*.links.changes`, you will note that this example only shows links that were "Unexpected". That means that the link was not found in the previous baseline. The issue "Unexpected Link" is listed after the link is presented.

Links Topology Verification

Links Found with incorrect configuration:

Rate MTU NodeGUID Port Type Name

60g 4096 0x00025500105baa00 1 CA IBM G2 Logical HCA

<-> 0x00025500105baa02 2 SW IBM G2 Logical Switch 1

Unexpected Link

20g 4096 0x00025500105baa02 1 SW IBM G2 Logical Switch 1

<-> 0x00066a0007000dbb 4 SW SilverStorm 9080 c938f4ql01 Leaf 2, Chip A

Unexpected Link

60g 4096 0x00025500106cd200 1 CA IBM G2 Logical HCA

<-> 0x00025500106cd202 2 SW IBM G2 Logical Switch 1

Unexpected Link

20g 4096 0x00025500106cd202 1 SW IBM G2 Logical Switch 1

<-> 0x00066a0007000dbb 5 SW SilverStorm 9080 c938f4ql01 Leaf 2, Chip A

Unexpected Link

60g 4096 0x00025500107a7200 1 CA IBM G2 Logical HCA

<-> 0x00025500107a7202 2 SW IBM G2 Logical Switch 1

Unexpected Link

20g 4096 0x00025500107a7202 1 SW IBM G2 Logical Switch 1

<-> 0x00066a0007000dbb 3 SW SilverStorm 9080 c938f4ql01 Leaf 2, Chip A

Unexpected Link



165 of 165 Fabric Links Checked

Links Expected but Missing, Duplicate in input or Incorrect:

159 of 159 Input Links Checked

Total of 6 Incorrect Links found

0 Missing, 6 Unexpected, 0 Misconnected, 0 Duplicate, 0 Different

Table 4 summarizes possible issues found in .changes files:

Table 4. Possible issues found in health check .changes files

Issue	Description and possible actions
Missing	<p>This indicates an item that is in the baseline, is not in this instance of health check output. This may indicate a broken item or a configuration change that has removed the item from the configuration.</p> <p>If you have intentionally removed this item from the configuration, save the original baseline and re-run the baseline as instructed in "Re-establishing Health Check baseline" on page 128. For example, if you've removed an HCA connection, the HCA and the link to it will be shown as Missing in fabric.*:.links.changes and fabric.*:.comps.changes files.</p> <p>If the item should still be part of the configuration, check for faulty connections or unintended changes to configuration files on the fabric management server.</p> <p>You should also look for any "Unexpected" or "Different" items that may correspond to this item. This would be in cases where the configuration of an item has changed in a way that makes it difficult to determine precisely how it has changed.</p>
Unexpected	<p>This indicates that an item is in this instance of health check output, but it is not in the baseline. This may indicate that an item was broken when the baseline was taken or a configuration change has added the item to the configuration.</p> <p>If you have added this item to the configuration, save the original baseline and re-run the baseline as instructed in "Re-establishing Health Check baseline" on page 128. For example, if you've added an HCA connection, will be shown as Unexpected in fabric.*:.links.changes and fabric.*:.comps.changes files.</p> <p>You should also look for any "Missing" or "Different" items that may correspond to this item. This would be in cases where the configuration of an item has changed in a way that makes it difficult to determine precisely how it has changed.</p>
Misconnected	<p>This only applies to links and indicates that a link is not connected properly. This should be fixed.</p> <p>It is possible to find miswires by examining all of the Misconnected links in the fabric. However, you will have to look at all of the fabric.*:.links.changes files to find miswires between subnets.</p> <p>You should also look for any "Missing" or "Different" items that may correspond to this item. This would be in cases where the configuration of an item has changed in a way that makes it difficult to determine precisely how it has changed.</p> <p>Individual links which are Misconnected are reported as "Incorrect Link" (see "Incorrect Link" on page 135) and are added into the Misconnected summary count.</p>
Duplicate	<p>This indicates that an item has a duplicate in the fabric. This situation should be resolved such that there is only one instance of any particular item being discovered in the fabric.</p> <p>This error can occur if there are changes in the fabric such as addition of parallel links. It can also be reported when there enough changes to the fabric that it is difficult to properly resolve and report all the changes. It can also occur when iba_report is run with manually generated topology input files which may have duplicate items or incomplete specifications.</p>

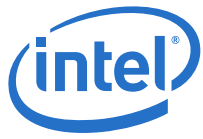
**Table 4. Possible issues found in health check .changes files**

Issue	Description and possible actions
Different	<p>This indicates that an item still exists in the current health check, but it is different from the baseline configuration.</p> <p>If the configuration has changed purposely since the most recent baseline, and the expected difference is reflected here, save the original baseline and re-run the baseline as instructed in "Re-establishing Health Check baseline" on page 128.</p> <p>If this difference was not intended, you must rectify the difference to prevent future health checks from reporting the same difference from the baseline.</p> <p>You should also look for any "Missing" or "Unexpected" items that may correspond to this item. This would be in cases where the configuration of an item has changed in a way that makes it difficult to determine precisely how it has changed.</p> <p>Individual items which are Different will be reported as "mismatched" or "Inconsistent" and are added into the Different summary count. See "X mismatch: expected ... found:" on page 135, "Node Attributes Inconsistent" on page 134, or "Port Attributes Inconsistent" on page 134, or "SM Attributes Inconsistent" on page 135.</p>
Port Attributes Inconsistent	<p>This indicates that the attributes of a port on one side of a link have changed, such as PortGuid, Port Number, Device Type, etc. The inconsistency would be caused by connecting a different type of device or a different instance of the same device type. This would also occur after replacing a faulty device.</p> <p>If the configuration has changed purposely since the most recent baseline, and this difference is reflected here, save the original baseline and re-run the baseline as instructed in "Re-establishing Health Check baseline" on page 128. If a faulty device were replaced, this would be a reason to re-establish the baseline.</p> <p>If this difference was not intended, you must rectify the difference to prevent future health checks from reporting the same difference from the baseline.</p> <p>This is a specific case of "Different". See "Different" on page 134.</p>
Node Attributes Inconsistent	<p>This indicates that the attributes of a node in the fabric have changed, such as NodeGuid, Node Description, Device Type, etc. The inconsistency would be caused by connecting a different type of device or a different instance of the same device type. This would also occur after replacing a faulty device.</p> <p>If the configuration has changed purposely since the most recent baseline, and this difference is reflected here, save the original baseline and re-run the baseline as instructed in "Re-establishing Health Check baseline" on page 128. If a faulty device were replaced, this would be a reason to re-establish the baseline.</p> <p>If this difference was not intended, you must rectify the difference to prevent future health checks from reporting the same difference from the baseline.</p> <p>This is a specific case of "Different". See "Different" on page 134.</p>

**Table 4. Possible issues found in health check .changes files**

Issue	Description and possible actions
SM Attributes Inconsistent	<p>This indicates that the attributes of the node or port running an SM in the fabric have changed, such as NodeGuid, Node Description, Port Number, Device Type, etc. The inconsistency would be caused by moving a cable, changing from host-based subnet management to embedded subnet management (or vice-versa), or by replacing the HCA in the fabric management server.</p> <p>If the configuration has changed purposely since the most recent baseline, and this difference is reflected here, save the original baseline and re-run the baseline as instructed in "Re-establishing Health Check baseline" on page 128. If the HCA in the fabric management server were replaced, this would be a reason to re-establish the baseline.</p> <p>If this difference was not intended, you must rectify the difference to prevent future health checks from reporting the same difference from the baseline.</p> <p>This is a specific case of "Different". See "Different" on page 134.</p>
X mismatch: expected ... found:	<p>This indicates an aspect of a an item has changed as compared to the baseline configuration. The aspect which changed and the expected and found values will be shown. This will typically indicate configuration differences such as MTU, Speed, Node description. It can also indicate that GUIDs have changed, such as when replacing a faulty device.(perhaps due to replacement of a device with a comparable device).</p> <p>If the configuration has changed purposely since the most recent baseline, and this difference is reflected here, save the original baseline and re-run the baseline as instructed in "Re-establishing Health Check baseline" on page 128. If a faulty device were replaced, this would be a reason to re-establish the baseline.</p> <p>If this difference was not intended, you must rectify the difference to prevent future health checks from reporting the same difference from the baseline.</p> <p>This is a specific case of "Different". See "Different" on page 134.</p>
Incorrect Link	<p>This only applies to links and indicates that a link is not connected properly. This should be fixed.</p> <p>It is possible to find miswires by examining all of the Misconnected links in the fabric. However, you will have to look at all of the fabric.*:*.links.changes files to find miswires between subnets.</p> <p>You should also look for any "Missing" or "Different" items that may correspond to this item. This would be in cases where the configuration of an item has changed in a way that makes it difficult to determine precisely how it has changed.</p> <p>This is a specific case of "Misconnected". See "Misconnected" on page 133.</p>

§ §





4.0 Complete Descriptions of Command Line Tools

4.1 Introduction

This section provides a complete description of each Intel® FastFabric Toolset command line tool and all its parameters. For new users, it is recommended to first learn the basic command line tools provided in the [Section 3.0, “Basic Command Line Tools”](#) on page 31.

4.2 Basic Single Host Operations

These tools are available on each host where the OFED+ packaging, or the Intel® FastFabric Toolset for OFED+ True Scale Fabric Stack Tools have been installed. These tools are mainly used to enable Intel® FastFabric Toolset operations against cluster nodes, however they can also be directly used on an individual host.

4.2.1 fastfabric

(Switch and Host) Takes the user to the top-level FastFabric text user interface (TUI) menu for setup and configuration.

4.2.1.1 Usage

```
fastfabric
```

4.2.1.2 Example

```
>fastfabric

Intel FastFabric IB Tools

Version: X.X.X.X.X

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

X) Exit
```

4.2.2 iba_config

(Switch and Host) Allows the user to configure FastFabric.

4.2.2.1 Usage

```
iba_config [-r root] [-v|-vv] [-F|-u|-s|-e comp] [-E comp] [-D comp] [--fwupdate
asneeded|always] [--user_queries|--no_user_queries] [--answer keyword=value]
```

or



```
iba_config -C
```

or

```
iba_config -V
```

4.2.2.2 Options

--help – Produce full help text

-F – Upgrade HCA Firmware with default options

--fwupdate *asneeded|always* – Select firmware update auto update mode

asneeded – update or downgrade to match version in this release

always – rewrite with this release version even if matches. The default is to upgrade as needed but do not downgrade. This option is ignored for interactive installs.

-u – Uninstall all ULPs and drivers with default options

-s – Enable autostart for all installed drivers

-r – Specify alternate root directory, default is /

-e *comp* – Uninstall the given component with default options. This can appear more than once on command line

-E *comp* – Enable autostart of a given component. This can appear with -D or more than once on the command line.

-D *comp* – Disable autostart of given component. This can appear with -E or more than once on command line

-v – Verbose logging.

-VV – Very verbose debug logging.

-C – Output list of supported components.

-V – Output Version.

--user_queries – Permit non-root users to query the fabric (default).

--no_user_queries – Non-root users cannot query the fabric.

--answer *keyword=value* – Provides an answer to a question that may occur during the operation. Answers to questions not asked are ignored. Invalid answers result in prompting for interactive installs, or use of the default for non-interactive.

Possible Questions

UserQueries – Permits non-root users to query the fabric.

SinglePort – Enable Intel® HCA single-port mode. The default options retain the existing configuration files.

4.2.3 p1info, p2info

(Host) Shows the port status for port 1 or port 2 respectively. On systems with more than one HCA, port 1 or port 2 status on all HCAs will be displayed. These commands are also used to show QSFP information for Intel® HCAs.



4.2.3.1 Usage

```
p1info [-q]
```

```
p2info [-q]
```

4.2.3.2 Options

```
-q -- show QSFP info if available
```

4.2.4 p1stats, p2stats

(Host) Shows the port performance counters for port 1 or port 2 respectively. On systems with more than one HCA, port 1 or port 2 counters on all HCAs will be shown.

4.2.4.1 Usage

```
p1stats
```

```
p2stats
```

4.2.5 clear_p1stats, clear_p2stats

(Host) Clears the port performance counters for port 1 or port 2 respectively. On systems with more than one HCA, port 1 or port 2 counters on all HCAs will be cleared.

4.2.5.1 Usage

```
clear_p1stats
```

```
clear_p2stats
```

4.2.6 iba_cabletest

(Switch) Used to initiate or stop Cable Bit Error Rate stress tests for HCA-to-SW links and/or ISLs.

4.2.6.1 Usage

```
iba_cabletest [-C|-A] [-c file] [-f hostfile] [-h 'hosts'] [-n numprocs]
[-t portsfile] [-p ports] [start|start_ca|start_isl|stop|stop_ca|stop_isl]
```

or

```
iba_cabletest --help
```

4.2.6.2 Options

```
--help - Produce full help text
```

```
-C - Clear error counters
```

```
-A - Force clear of hardware error counters implies -C
```

```
-c file - Error thresholds config file. The default is
/etc/sysconfig/iba/iba_mon.si.conf only used if -C or -A specified.
```

```
-f hostfile - File with hosts to include in HCA-to-SW test, The default is
/etc/sysconfig/iba/hosts
```



-h *hosts* – List of hosts to include in HCA-SW test

-n *numprocs* – Number of processes per host for HCA-SW test

-t *portsfile* – File with list of local HCA ports used to access fabric(s) when clearing counters, default is `/etc/sysconfig/iba/ports`

-p *ports* – List of local HCA ports used to access fabric(s) for counter clear default is 1st active port. This is specified as ***hca:port***

This is specified as ***hca:port***

0:0 = 1st active port in system

0:y = Port y within system

x:0 = 1st active port on HCA x

x:y = HCA x, port y

The first HCA in the system is 1. The first port on an HCA is 1.

start – Start the HCA-SW and ISL tests

start_ca – Start the HCA-SW test

start_isl – Start the ISL test

stop – Stop the HCA-SW and ISL tests

stop_ca – Stop the HCA-SW test

stop_isl – Stop the ISL test

The HCA-SW cabletest requires that `FF_MPI_APPS_DIR` be set, and contains a prebuilt copy of Intel's `mpi_apps` for an appropriate MPI.

The ISL cable test started by this tool assumes that the master Host subnet manager is running on this host. If using the Embedded subnet manager, or if a different host is the master Fabric Manager (FM), the ISL cabletest will have to be controlled by the switch CLI, or by FastFabric on the master FM. respectively.

4.2.6.3 Environment Variables

The following environment variables are also used by this command:

HOSTS – List of hosts, used if **-h** option not supplied

HOSTS_FILE – File containing list of hosts, used in absence of **-f** and **-h**

PORTS – List of ports, used in absence of **-t** and **-p**

PORTS_FILE – File containing list of ports, used in absence of **-t** and **-p**

FF_MAX_PARALLEL – Maximum concurrent operations

4.2.6.4 Example

```
iba_cabletest -A start
iba_cabletest -h 'arwen elrond' start_ca
HOSTS='arwen elrond' iba_cabletest stop
```

4.2.7 iba_capture

Captures critical system information into a zipped tar file. The resulting tar file should be sent to Customer Support along with any Intel® True Scale Fabric problem report regarding this system.



4.2.7.1 Usage

```
iba_capture [-d detail] output_tar_file
```

or

```
iba_capture --help
```

4.2.7.2 Options

--help - Show full help text

-d *detail* - Level of detail of capture

The Details levels are:

Normal - Obtains local information from each host

Fabric - In addition to Normal, also obtains basic fabric information by queries to the SM and fabric error analysis by means of the *iba_report*.

Fabric+FDB - In addition to Fabric, also obtains all the switch forwarding tables from the SM.

Analysis - In addition to Fabric+FDB, also obtains all_analysis results. If all_analysis has not yet been run, it is run as part of the capture.

output tar file - Name of a file to be created by *iba_capture*. The file name specified will be overwritten if it already exists. It is recommended to use the *.tgz* suffix in the filename supplied. If the filename given does not have a *.tgz* suffix, the *.tgz* suffix will be added.

4.2.7.3 Notes

Detail levels 2 through 4 can be used when fabric operational problems occur. If the problem is most likely node specific, detail level 1 should be sufficient. Detail levels 2 through 4 require an operational Fabric Manager. Typically your support representative will request a given detail level. If a given detail level takes excessively long or fails to be gathered, try a lower detail level.

For detail levels 2 through 4, the additional information is only available on a node with Intel® FastFabric Toolset installed. The information is gathered for every fabric specified in the */etc/sysconfig/iba/ports* file.

4.2.7.4 Example

```
iba_capture mycapture.tgz
```

```
iba_capture -d 3 030127capture.tgz
```

Note: The resulting host capture file requires a significant amount of space on the host. The actual size will vary and can be multiple megabytes. Ensure that adequate disk space is available on the host system.

4.2.8 iba_expand_file

(Linux) Expands a fastfabric hosts, chassis or ibnodes file. This tool will expand and filter out blank and comment lines. This can be useful when building other scripts that may use these files as input.

4.2.8.1 Usage

```
iba_expand_file file
```



or

```
iba_expand_file --help
```

4.2.8.2 Options

--help – Produces full help text

file – FastFabric hosts, chassis or ibnodes file to expand, and remove comment and blank lines.

4.2.8.3 Example

```
iba_expand_file allhosts
```

4.2.9 iba_linkanalysis

(Switch) Encapsulates the capabilities for link analysis from the “Check Status of IB Ports” option of the FastFabric TUI. Additionally, this tool includes cable and fabric topology verification capabilities. This tool is built on top of `iba_report` (and its analysis capabilities), and accepts the same syntax for input topology and snapshot files.

In addition to being able to run assorted `iba_report` link analysis reports, and generate human-readable output, this tool additionally analyzes the results and appends a concise summary of issues found to the `FF_RESULT_DIR/punchlist.csv` file.

4.2.9.1 Usage

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

or

```
iba_linkanalysis --help
```

4.2.9.2 Options

--help – Produces full help text

-U – Omit unexpected devices and links in punchlist from verify reports

-t portsfile – File with list of local HCA ports used to access fabric(s) for analysis, default is `/etc/sysconfig/iba/ports`

-p ports – List of local HCA ports used to access fabric(s) for analysis. The default is the 1st active port.

This is specified as **hca:port**

0:0 = 1st active port in system

0:y = Port y within system

x:0 = 1st active port on HCA x

x:y = HCA x, port y

The first HCA in the system is 1. The first port on an HCA is 1.

-T topology_input – Name of a topology input file to use. Any %P markers in this filename will be replaced with the hca:port being operated on (such as 0:0 or 1:2) default is `/etc/sysconfig/iba/topology.%P.xml` if NONE is specified, will not use any topology_input files See “[iba_report](#)” on [page 175](#) for more information on



topology_input files.

-x snapshot_input - Perform analysis using data in snapshot_input. snapshot_input must have been generated via a previous iba_report -o snapshot run. If an errors report is specified, snapshot must have been generated with the iba_report -s option. When this option is used, only one port may be specified to select a topology_input file (unless -T specified). When this option is used, clearerrors and clearhwerrors reports are not permitted.

-x snapshot_suffix - Create a snapshot file per selected port. The files will be created in FF_RESULT_DIR with names of the form: snapshotSUFFIX.HCA:PORT.xml.

-c file - Error thresholds configuration file. The default is /etc/sysconfig/iba/iba_mon.si.conf reports. The following reports are supported:

- errors** - Link error analysis
- slowlinks** - Links running slower than expected
- misconfiglinks** - Links configured to run slower than supported
- misconnlinks** - Links connected with mismatched speed potential
- all** - Includes all reports above
- verifylinks** - Verify links against topology input
- verifyextlinks** - Verify links against topology input limit analysis to links external to systems
- verifycas** - Verify CAs against topology input
- verifysws** - Verify Switches against topology input
- verifyrtrs** - Verify Routers against topology input
- verifynodes** - Verify CAs, Switches and Routers against topology input
- verifysms** - Verify SMs against topology input
- verifyall** - Verifies links, CAs, Switches, Routers and SMs against topology input
- clearerrors** - Clear error counters, uses PM if available
- clearhwerrors** - Clear HW error counters, bypasses PM
- clear** - Includes clearerrors and clearhwerrors

A punchlist of bad links is also appended to FF_RESULT_DIR/punchlist.csv.

4.2.9.3 Environment Variables

The following environment variables are also used by this command:

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

4.2.9.4 Example

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

4.2.10 iba_showmc

(Linux) Displays the Multicast groups created for the fabric along with the CA ports which are a member of each multicast group. This command can be helpful when attempting to analyze or debug multicast usage by applications or ULPs such as IPoIB.



4.2.10.1 Usage

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

or

```
iba_showmc --help
```

4.2.10.2 Options

--help – Produce full help text

-t *portsfile* – File with list of local HCA ports used to access fabric(s) for analysis, default is /etc/sysconfig/iba/ports

-p **ports** – List of local HCA ports used to access fabric(s) for analysis default is 1st active port.

This is specified as **hca:port**

0:0 = 1st active port in system

0:y = Port y within system

x:0 = 1st active port on HCA x

x:y = HCA x, port y

The first HCA in the system is 1. The first port on an HCA is 1.

4.2.10.3 Example

```
iba_showmc
```

```
iba_showmc -p '1:1 1:2 2:1 2:2'
```

4.2.10.4 Environment Variables

The following environment variables are also used by this command:

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

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

4.2.11 iba_hca_rev

(Linux) Displays information about the HCAs in the system including model numbers, serial numbers, board revisions, and other HCA model specific information.

4.2.11.1 Usage

```
iba_hca_rev [-v]
```

4.2.11.2 Options

-v – Reports additional information about Mellanox adapter firmware, including detailed output of the configuration options and verification of the firmware image.

4.2.12 iba_portenable

(Host or Switch) Enables a specified HCA port on the local host or remote switch. May also be used to change port operational parameters as part of enabling the port.

4.2.12.1 Usage

```
iba_portenable [-v] [-D] [-l lid [-m dest_port]] [-h hca] [-p port] [-w width] [-s
```




`speed] [-K mkey]`

4.2.12.2 Options

- v – Verbose output
- D – Do not cycle port through disabled physstate
- l **lid** – Destination lid, default is local port
- m **dest_port** – Destination port, default is port with given lid useful to access switch ports
- h **hca** – HCA to send by/to, default is 1st hca
- p **port** – Port to send by/to, default is 1st port
- s **speed** – New link speeds enabled (default is 0)
 - 0 – no-op
 - 1 – 2.5 Gb/s
 - 2 – 5 Gb/s
 - 4 – 10 Gb/s

To enable multiple speeds, use the sum of desired speeds.
255 – Enable all speeds supported by given HCA port
- w **width** – new link widths enabled (default is 0)
 - 0 – no-op
 - 1 – 1x
 - 2 – 4x
 - 4 – 8x
 - 8 – 12x

To enable multiple widths, use sum of desired widths
255 – Enable all widths supported by given HCA port
- K **mkey** – SM management key to access remote ports

4.2.12.3 Example

```
iba_portenable
```

```
iba_portenable -p 2 -h 2
```

The port enablement is transient in nature. If the given host/switch is rebooted or its True Scale Fabric stack is restarted, the port will revert to its default configuration and state. Typically, the default state has the port enabled with all speeds and widths supported by the given HCA/switch port enabled.

To access switch ports through this command, the -l and -m options must be given. The -l option will specify the lid of switch port 0 (the logical management port for the switch) and -m will specify the actual switch port to access. If SMA mkeys are being used, the -K option will also be needed. By default the Intel® SM does not use SMA mkeys, in which case this option does not need to be used.

Note: On many DDR HCA and switch models, in order to enable DDR (5.0Gb) operation, SDR (2.5 Gb) operation must also be enabled. For example, the -s2 option may yield a port which does not come up, in which case -s3 is preferred.



4.2.13 iba_portdisable

(Host or Switch) Disables a specified HCA port on the local host or a remote switch. May also be used to change port operational parameters as part of disabling the port.

4.2.13.1 Usage

```
iba_portdisable [-v] [-l lid [-m dest_port]] [-h hca] [-p port] [-w width] [-s speed] [-K mkey]
```

4.2.13.2 Options

- v – Verbose output
- l **lid** – Destination lid, default is local port
- m **dest_port** – Destination port, default is port with given lid useful to access switch ports
- h **hca** – HCA to send by/to, default is 1st hca
- p **port** – Port to send by/to, default is 1st port
- s **speed** – New link speeds enabled (default is 0)
 - 0 – no-op
 - 1 – 2.5 Gb/s
 - 2 – 5 Gb/s
 - 4 – 10 Gb/sTo enable multiple speeds, use the sum of desired speeds.
255 – enable all speeds supported by given HCA port
- w **width** – New link widths enabled (default is 0)
 - 0 – no-op
 - 1 – 1x
 - 2 – 4x
 - 4 – 8x
 - 8 – 12xTo enable multiple widths, use sum of desired widths
255 – Enable all widths supported by given HCA port
- K **mkey** – SM management key to access remote ports

4.2.13.3 Example

```
iba_portdisable
```

```
iba_portdisable -p 2 -h 2
```

The port disabled state is transient in nature. If the given host is rebooted or its True Scale Fabric stack is restarted, the port will revert to its default configuration and state. Typically, the default state has the port enabled with all speeds and widths supported by the given HCA port enabled.

To access switch ports through this command, the -l and -m options must be given. The -l option will specify the lid of switch port 0 (the logical management port for the switch) and -m will specify the actual switch port to access. If SMA mkeys are being used, the -K option will also be needed. By default the Intel® SM does not use SMA mkeys, in which case this option does not need to be used.



Note: On many DDR HCA and switch models, in order to enable DDR (5.0Gb) operation, SDR (2.5 Gb) operation must also be enabled. For example, the `-s2` option may yield a port which does not come up, in which case `-s3` is preferred.

Note: When using this command to disable switch ports, if the final port in the path between the Fabric Management Node and the switch is disabled, then `iba_portenable` will not be able to reen able it. In which case the switch CLI and/or a switch reboot may be needed to correct the situation.

4.2.14 iba_portconfig

(Host or Switch) Controls configuration and state of a specified HCA port on the local host or a remote switch.

4.2.14.1 Usage

```
iba_portconfig [-v] [-D] [-l lid [-m dest_port]] [-h hca] [-p port] [-z] [-S state]
[-P physstate] [-w width] [-s speed]
```

4.2.14.2 Options

- `-v` – Verbose output
- `-D` – Do not cycle port through disabled physstate
- `-l lid` – Destination lid, default is local port
- `-m dest_port` – Destination port, default is port with given lid useful to access switch ports
- `-h hca` – HCA to send by/to, default is 1st hca
- `-p port` – Port to send by/to, default is 1st port
- `-z` – Do not get port info first, clear most port attributes
- `-S state` – New State (default is 0)
 - 0 – No-op
 - 1 – Down
 - 2 – Init
 - 3 – Armed
 - 4 – Active
- `-P physstate` – New physical State (default is 0)
 - 0 – No-op
 - 1 – Sleep
 - 2 – Polling
 - 3 – Disabled
- `-s speed` – New link speeds enabled (default is 0)
 - 0 – No-op
 - 1 – 2.5 Gb/s
 - 2 – 5 Gb/s
 - 4 – 10 Gb/s

To enable multiple speeds, use the sum of the desired speeds.

 - 255 – Enable all speeds supported by given HCA port



`-w width` – New link widths enabled (default is 0)
0 – No-op
1 – 1x
2 – 4x
4 – 8x
8 – 12x
To enable multiple widths, use sum of desired widths
255 – Enable all widths supported by given HCA port

`-K mkey` – SM management key to access remote ports

4.2.14.3 Example

```
iba_portconfig -w 1  
  
iba_portconfig -p 2 -h 2 -w 3
```

The port configuration is transient in nature. If the given host is rebooted or its True Scale Fabric stack is restarted, the port will revert to its default configuration and state. Typically, the default state is to have the port enabled with all speeds and widths supported by the given HCA port enabled.

To access switch ports through this command, the `-l` and `-m` options must be given. The `-l` option will specify the lid of switch port 0 (the logical management port for the switch) and `-m` will specify the actual switch port to access. If SMA mkeys are being used, the `-K` option will also be needed. By default the Intel® SM does not use SMA mkeys, in which case this option does not need to be used.

Note: On many DDR HCA and switch models, in order to enable DDR (5.0Gb) operation, SDR (2.5 Gb) operation must also be enabled. For example, the `-s2` option may yield a port which does not come up, in which case `-s3` is preferred.

Caution: When using this command to disable or reconfigure switch ports, if the final port in the path between the Fabric Management Node and the switch is disabled or fails to come online, then `iba_portenable` will not be able to reenabling it. In which case the switch CLI and/or a switch reboot may be needed to correct the situation.

4.2.15 iba_portinfo

(Host or Switch) Displays configuration and state of a specified HCA port on the local host or a remote switch.

4.2.15.1 Usage

```
iba_portinfo [-v] [-l lid [-m dest_port]] [-h hca] [-p port] [-K mkey]
```

4.2.15.2 Options

`-v` – Verbose output

`-l lid` – Destination lid, default is local port

`-m dest_port` – Destination port, default is port with given lid useful to access switch ports

`-h hca` – HCA to send by/to, default is 1st hca

`-p port` – Port to send by/to, default is 1st port



`-K mkey` – SM management key to access remote ports

4.2.15.3 Example

```
iba_portinfo -p 1
iba_portinfo -p 2 -h 2 -l 5 -m 18
```

To access switch ports through this command, the `-l` and `-m` options must be given. The `-l` option will specify the lid of switch port 0 (the logical management port for the switch) and `-m` will specify the actual switch port to access. If SMA mkeys are being used, the `-K` option will also be needed. By default the Intel® SM does not use SMA mkeys, in which case this option does not need to be used.

4.2.16 iba_portstats

(Host or Switch) Displays port performance counters of a specified HCA port on the local host or a remote switch.

4.2.16.1 Usage

```
iba_portstats [-v] [-l lid [-m dest_port]] [-h hca] [-p port]
```

4.2.16.2 Options

`-v` – Verbose output

`-l lid` – Destination lid, default is local port

`-m dest_port` – Destination port, default is port with given lid useful to access switch ports

`-h hca` – HCA to send by/to, default is 1st hca

`-p port` – Port to send by/to, default is 1st port

4.2.16.3 Example

```
iba_portstats -p 2 -h 2 -l 5 -m 18
```

To access switch ports through this command, the `-l` and `-m` options must be given. The `-l` option will specify the lid of switch port 0 (the logical management port for the switch) and `-m` will specify the actual switch port to access.

4.2.17 iba_portclear

(Host or Switch) Clears port performance counters of a specified HCA port on the local host or a remote switch.

4.2.17.1 Usage

```
iba_portclear [-v] [-l lid [-m dest_port]] [-h hca] [-p port]
```

4.2.17.2 Options

`-v` – Verbose output

`-l lid` – Destination lid, default is local port



`-m dest_port` – Destination port, default is port with given lid useful to access switch ports

`-h hca` – HCA to send by/to, default is 1st hca

`-p port` – Port to send by/to, default is 1st port

4.2.17.3 Example

```
iba_portclear -p 2 -h 2 -l 5 -m 18
```

To access switch ports through this command, the `-l` and `-m` options must be given. The `-l` option will specify the lid of switch port 0 (the logical management port for the switch) and `-m` will specify the actual switch port to access.

4.2.18 iba_resolve_hca_port

(Host) `iba_resolve_hca_port` permits the OFED+ style HCA number and port number arguments to be converted to an OFED style HCA name and physical port number. This can be useful when writing scripts that can accept FastFabric-style arguments, and interact directly with OFED comments.

4.2.18.1 Usage

```
iba_resolve_hca_port hca port
```

4.2.18.2 Options

`-hca` – HCA to send via, default is 1st hca

`-port` – Port to send via, default is 1st active port

4.2.19 iba_sorthosts

The `iba_sorthosts` command will sort its stdin in a typical host name order. Hosts are stored alphabetically by any alpha numeric prefix and then sorted numerically by any numeric suffix. Leading zeros in the numeric suffix are optional. This command does not remove duplicates, any duplicates are listed in adjacent lines.

This command can be useful to build `mpi_hosts` input files for applications or `cabletest` which places hosts in order by name.

4.2.19.1 Usage

```
iba_sorthosts < hostlist > mpi_hosts
```

or

```
iba_sorthosts --help
```

4.2.19.2 Options

`--help` – produce full help text

Sort the `hostlist` alphabetically (case insensitively) then numerically hostnames may end in a numeric field which may optionally have leading zeros.



4.2.19.3 Example input

```
osd04
osd1
compute20
compute3
mgmt1
mgmt2
login
```

4.2.19.4 Resulting output

```
compute3
compute20
login
mgmt1
mgmt2
osd1
osd04
```

4.2.20 iba_verifyhosts

`iba_verifyhosts` is a tool to help perform single node verification. The actual verification is performed using `/root/hostverify.sh`. A sample of `hostverify.sh` is provided in `/opt/iba/samples/hostverify.sh` and should be reviewed and edited to set appropriate configuration and performance expectations and select which tests to run by default. See `/opt/iba/samples/hostverify.sh` and `/sbin/iba_verifyhost.sh` for more information.

Note: `iba_verifyhosts` now supports systems with multiple HCAs. To configure for multiple HCAs, the variables at the start of the script `hostverify.sh` located at `/opt/iba/samples/` must be edited:

- **HCA_COUNT** - number of Intel HCAs expected in system. If 0 then tests like `pcispeed`, `pcicfg` and `ipath_pkt_test` merely verify there are no Intel HCAs.
- **HCA_CPU_CORE[0]** - CPU core to run `ipath_pkt_test` on when testing HCA 0.
- Additional variables, such as **HCA_CPU_CORE[1]**, must be specified when there is more than 1 HCA. It is recommended to select a CPU core (other than 0) that is in a CPU chip closest to the HCA within the server. For example, Core systems have a PCI bus on each CPU chip and, it is recommended to evaluate HCA performance by using the PCI bus of the CPU chip the HCA is connected to.

4.2.20.1 Usage

```
iba_verifyhosts [-kc] [-f hostfile] [-u upload_file] [-d upload_dir] [-h 'hosts']
[-T timelimit] [test ...]
```

or



```
iba_verifyhosts --help
```

4.2.20.2 Options

--help – Produce full help text.

-k – At start and end of verification, kill any existing hostverify or xhpl jobs on the hosts

-c – Copy hostverify.sh to hosts first, useful if you have edited it

-f *hostfile* – File with hosts in cluster, default is /etc/sysconfig/iba/hosts

-h *hosts* – List of hosts to ping

-u *upload_file* – Filename to upload hostverify.res to after verification to allow backup and review of the detailed results for each node. The default upload destination file is hostverify.res. If **-u ''** is specified, no upload will occur.

-d *upload_dir* – Directory to upload result from each host. Default is uploads

-T *timelimit* – Time limit (in seconds) for host to complete tests. Default of 300 seconds (5 minutes)

test – One or more specific tests to run. This verifies basic node configuration and performance by running /root/hostverify.sh on all specified hosts. These tests are:

- **pcicfg** - verify PCI max payload and max read request size settings
- **pcispeed** - verify PCI bus negotiated to PCIe Gen2 x8 speed
- **cstates** - verify CPU cstates are disabled
- **initscripts** - verify irqbalance, irq_balancer, powerd and cpuspeed init.d scripts are disabled
- **hyperthreading** - verify hyperthreading is disabled
- **ipath_pkt** - check PCI-HCA bus performance. Requires HCA port is Active
- **memsize** - check total size of memory in system
- **hpl** - perform a single node HPL test, useful to determine if all hosts perform consistently
- **default** - run all tests selected in TESTS

Prior to using this, edit /opt/iba/samples/hostverify.sh to set proper expectations for node configuration and performance. Then be sure to use the **-c** option on first run for a given node so that /opt/iba/samples/hostverify.sh gets copied to each node as /root/hostverify.sh.

A summary of results is appended to **FF_RESULT_DIR/verifyhosts.res**. A punchlist of failures is also appended to **FF_RESULT_DIR/punchlist.csv**. Only failures are shown on stdout.

4.2.20.3 Environment

HOSTS – List of hosts, used if **-h** option not supplied

HOSTS_FILE – File containing list of hosts, used in absence of **-f** and **-h**

UPLOADS_DIR – Directory to upload to, used in absence of **-d**

FF_MAX_PARALLEL – Maximum concurrent operations



4.2.20.4 Example

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

4.2.21 iba_xlat_topology

`iba_xlat_topology`, accompanied by `topology.xlsx`, `linksum_180.csv` and `linksum_360.csv` provide the capability to document the topology of a customer cluster, and generate a topology XML file based on that topology ("translate" the spread sheet to a topology file). The topology file can be used to bring up and verify the cluster.

`topology.xlsx` provides a standard format for representing each external link in a cluster. Each link contains **Source**, **Destination**, and **Cable** fields with one link per row of the spread sheet. The cells cannot contain commas. **Source** and **Destination** fields each have the following columns:

- **Rack Group**
- **Rack**
- **Name** (primary name)
- **Name-2** (secondary name)
- **Port** number
- Port **Type**
- The **Cable** fields have the following columns:
 - **Label**
 - **Length**
 - **Details**

The **Rack Group** and **Rack** names are individually optional. If either column is completely empty it will be ignored. If the **Rack Group** or **Rack** field is empty on a particular row, the script will default the value in that field to the closest previous value (Defaulting the field to a non-empty value. The first row must have a value.). **Name** and **Name-2** provide the name of the node which is output as the NodeDesc using the following information:

- NodeType = Name or Name-2
- Host = hostname or hostdetails
- Edge Switch = switchname
- Core Leaf = corename or Lnnn
- Core Spine = corename or Snnn (used only in internal core switch links)

For hosts **Name-2** is optional and is output as NodeDetails in the topology XML file; also `HCA-1` is appended to Name (see `-c` option). For core leaves (and spines) **Name** and **Name-2** are concatenated (see `-c` option).

Port contains the port number. If the **Port** field is empty on a host node, the script will default to 1.

Type contains the node type. If the **Type** field is empty on a particular row, the script will default the value to the closest previous value (at least the first row must have a value). The type values are:



- NodeType = Type
- Host = CA
- Edge Switch = SW
- Core Leaf = CL
- Core Spine = CS (used only in internal core switch links)

Cable values are optional and have no special syntax. If the cable information is present it will appear in the topology XML file as **CableLabel**, **CableLength** and **CableDetails** respectively.

The `iba_xlat_topology` script reads the topology, linksum_180 and linksum_360 CSV (Comma-Separated-Values) files. The `topology.csv` file is created from the `topology.xlsx` spread sheet by saving the **Fabric Links** tab as a CSV file to `topology.csv`. The `topology.csv` file should be inspected to ensure that each row contains the correct and same number of comma separators. Any extraneous entries on the excel spread sheet can cause the CSV output to have extra fields.

The script produces as an output one or more topology files starting with `topology.0:0.xml`. Output at the top level as well as Group, Rack, and Switch level can be produced. Input files must be present in the same directory from which the script operates.

4.2.21.1 Usage

```
iba_xlat_topology [-d level -v level -i level -c char -K -?]
```

4.2.21.2 Options

- d *level* – Output detail level (default 0), values are additive
 - 1 – Edge switch topology files
 - 2 – Rack topology files
 - 4 – Rack group topology files
- v *level* – Verbose level (0-8, default 2)
 - 0 – No output
 - 1 – Progress output
 - 2 – Reserved
 - 4 – Time stamps
 - 8 – Reserved
- i *level* – Output indent level (0-15, default 0)
- c *char* – Nodedesc concatenation char (default SPACE)
- K – Do not clean temporary files
- ? – Print this output

The output detail level specifies the level to which the script will produce topology XML files. By default the top level is always produced, but edge switch, rack and rack group topology files can also be produced. If the output at the group or rack level is specified, then group or rack names must be provided in the spread sheet. Detailed output can be specified in any combination. A directory for each topology XML file will be created hierarchically, with group directories (if specified) at the highest level, followed by rack and edge switch directories (if specified).



The concatenation character (`-c char`) is used when creating NodeDesc values (Name to Name-2, Name to HCA-1, and so on). A space is used by default, but another character (ex. underscore) can be specified.

The `-K` option is used to prevent temporary files (in each topology directory) from being removed. Temporary files contain lists (CSV) of links, CAs, and switches used to create a topology XML file. They are not normally needed after a topology file is created, but they are used in the creation of `linksum_180.csv` and `linksum_360.csv`, or can be retained for subsequent inspection or processing.

The `linksum_180.csv` and `linksum_360.csv` are provided as stand-alone source files. However, they can be recreated (or modified) from the spread sheet, if needed, by performing the following steps:

1. Save each of the following from the `topology.xlsx` Excel file as individual as CSV files
 - **Internal 180 Links** tab as `linksum_180.csv`
 - **Internal 360 Links** tab as `linksum_360.csv`
 - **Fabric Links** tab as `topology.csv`
2. For each saved `topology.csv` file, run the script with the `-K` option. Upon completion of the script, save the top level `linksum.csv` file as `linksum_180.csv` or `linksum_360.csv` as appropriate.

4.2.22 iba_xlat_topology_cust

The script `iba_xlat_topology_cust` has been added, accompanied by `topology_cust.xlsx`. The script and spread sheet provide a sample alternative to the standard-format topology capability to document the topology of a customer cluster (see `iba_xlat_topology`). The alternative is provided for situations in which a customer chooses not to define a fabric topology using the standard-format spread sheet and `iba_xlat_topology.topology_cust.xlsx` provides an alternative for representing each external link in a cluster. `iba_xlat_topology_cust` translates the CSV form of the alternate spread sheet cluster tab(s) to the standard CSV form used by `iba_xlat_topology`. In using the alternative, a user would modify the sample spread sheet as needed to fit specific needs, then modify the script as needed to translate the spread sheet CSV output to the standard-format CSV output.

Like the standard format, each link contains source, destination and cable fields with one link per line (row) of the spread sheet. Link fields must not contain commas. Source and Destination fields are each a concatenation of name and port information as shown in Table 5 (N/n is a host/switch/port number; names not of the form 'ib' or 'C' are taken to be host names):

Table 5. iba_xlat_topology_cust

NodeType	Source/Destination
Host	hostN
Edge Switch	ibNpN
Core Leaf	CnLnnnpN

Cable values `CableLength` and `CableDetails` are optional and have no special syntax. If present, they are placed in the standard-format CSV file exactly as they appear. `CableLabel` is created automatically by `iba_xlat_topology_cust` as the concatenation (see `-c` option below) of `Source` and `Destination`. Rack Group and Rack are not supported in `topology_cust.xlsx`. `iba_xlat_topology_cust` leaves these fields empty in the standard-format CSV file.



4.2.22.1 Usage

```
iba_xlat_topology_cust -t topology_prime [-s topology_second] -T topology_out [-v level] [-i level] [-c char] [-K] [-?]
```

4.2.22.2 Options

- t *topology_prime* – Primary topology CSV input file
- s *topology_second* – Secondary topology CSV input file
- T *topology_out* – Topology CSV output file
- v *level* – Verbose level (0-8, default 2)
 - 0 – No output
 - 1 – Progress output
 - 2 – Reserved
 - 4 – Time stamps
 - 8 – Reserved
- i *level* – Screen output indent level (0-15, default 0)
- c *char* – Concatenation char (default SPACE)
- K – DO NOT clean temporary files
- ? – Print this output

-t *topology_prime* specifies the primary CSV input file and must be present. If needed -s *topology_second* can specify a secondary CSV input file. It will be appended to the primary for processing. -T *topology_out* specifies the CSV output file name and must be specified. The concatenation character (*c char*) is used when creating Cable Label values. A space is used by default, but another character (e.g., underscore) can be specified. -K is used to prevent temporary files from being removed. Temporary files contain CSV data used during processing. They are not needed after the standard-format CSV file is created, but they can be retained for subsequent inspection or processing.

4.2.23 HCA port thresholding using iba_mon

(Host) *iba_mon* is a daemon process which can be started on individual hosts to monitor the port state and statistics of all HCAs on the given host.

4.2.23.1 Usage

```
iba_mon [-v|-q] [-d] [-c file] [-f facility]
```

4.2.23.2 Options

- v – Verbose output
- q – No output
- d – Daemon (detach from terminal)
- c *file* – Configuration file, default is /etc/sysconfig/iba/iba_mon.conf
- f *facility* – Syslog facility, default is local6



Normally, `iba_mon` is run as a background process started by the `/etc/init.d/iba_mon` initialization script. The `iba_config` or `INSTALL` commands may be used to configure `iba_mon` to be started automatically at system boot time.

The `iba_mon.conf` file (see *Appendix A* in the *Intel® True Scale Fabric Suite FastFabric User Guide* for more information) defines the statistics that `iba_mon` will monitor and how often it will clear them for threshold analysis.

When `iba_mon` detects a port state change (for example, a port going down or becoming active) it will log output to syslog at the syslog facility level specified. Similarly, when `iba_mon` detects a configured threshold has been exceeded for a statistic over the specified interval, it will log the affected statistic and its value over the interval.

Note: It is recommended to not run `iba_mon` when fabric level tools (such as `all_analysis`, `fabric_analysis`, `iba_report`, `iba_reports` or a centralized Performance Manager) are being used for port statistics analysis. Alternatively, if desired, `iba_mon` can be run provided the statistics being centrally-monitored are configured with a threshold of 0 in `iba_mon.conf`, such that `iba_mon` will not monitor or clear the given statistic.

4.2.24 s20tune

(Host) `s20tune` is a daemon process which can be started on individual hosts to monitor the port state and speed of all HCAs on the given host. This process must be run on any hosts which have HCAs which do not support an IBTA compliant DDR or QDR link speed negotiation. This tool monitors for the link to stay down for 10 seconds or more and then restores the enabled speed to match the supported speed, therefore restarting the speed negotiation process. For more information on FM based link speed negotiation, see the *Intel® True Scale Fabric Suite Fabric Manager User Guide*.

4.2.24.1 Usage

```
s20tune -F -C [-v|-q] [-D] [-h hca]
```

4.2.24.2 Options

`-F` – Force speed. For a port which is down for 10 seconds or more, force the enabled speed to match the supported speeds for the HCA port. This helps to recover from cable pulls when doing auto negotiation using the `LinkSpeedOverride` option in the Intel® FM.

`-C` – Do not check link performance

`-v` – Verbose output

`-q` – Quiet mode

`-D` – Daemon (detach from terminal)

`-h hca` – HCA to monitor, default is 1st HCA

Normally, `s20tune` is run as a background process started by the `/etc/init.d/s20tune` initialization script. The `iba_config` or `INSTALL` commands may be used to configure `s20tune` to be started automatically at system boot time.

If it is desired to manually force the link speed, `s20tune` should not be run.

Note: `s20tune` has additional arguments which are not documented above. It is recommended not to use such arguments unless directed to do so by Intel Support.



4.3 Basic Setup and Administration Tools

4.3.1 pingall

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

4.3.1.1 Usage

```
pingall [-Cp] [-f hostfile] [-F chassisfile] [-h HOSTS] [-H chassis]
```

or

```
pingall --help
```

4.3.1.2 Options

--help – Produce full help text

-C – Performs a ping against a chassis. The default is hosts

-p – Ping all hosts/chassis in parallel

-f *hostfile* – File with hosts in cluster, default is /etc/sysconfig/iba/hosts

-F *chassisfile* – File with chassis in cluster default is /etc/sysconfig/iba/chassis

-h *HOSTS* – List of hosts to ping

-H *chassis* – List of chassis to ping

4.3.1.3 Example

```
pingall
```

```
pingall -h 'arwen elrond'
```

```
HOSTS='arwen elrond' pingall
```

```
pingall -C
```

```
pingall -C -H 'chassis1 chassis2'
```

```
CHASSIS='chassis1 chassis2' pingall -C
```

4.3.1.4 Environment Variables

The following environment variables are also used by this command:

HOSTS, *HOSTS_FILE* – See discussion on [“Selection of Hosts” on page 23](#).

CHASSIS, *CHASSIS_FILE* – See discussion on [“Selection of Chassis” on page 24](#).

FF_MAX_PARALLEL – When -p option is used, maximum concurrent operations will be performed.

Note:

This command pings all hosts/chassis found in the specified host/chassis file. The use of -C option merely selects the default file and/or environment variable to use. For this command it is valid to use a file which lists both hosts and chassis.



4.3.2 check_rsh

(Linux) Verifies that `rsh` is set up to allow passwordless file copies (RCP) and commands (`rsh`) to be run from this host to all the other hosts (and to itself using `localhost`) as a specific user (default is `root`). Additionally, this command can be used to verify `rsh` is setup to allow MPI to use `rsh` for job startup.

Note: For security reasons, configuration and use of `rsh/rcp/rlogin` is no longer recommended. Instead `ssh` is recommended. `SSH` may be used by MPI as well as `setup_ssh`.

Note: This command is deprecated and will be removed in a future release.

4.3.2.1 Usage

```
check_rsh [-i ipoib_suffix] [-f hostfile] [-h 'HOSTS'] [-u user]
```

or

```
check_rsh --help
```

4.3.2.2 Options

`--help` - Produce full help text

`-i ipoib_suffix` - Suffix to apply to host names to create IPoIB host names. The default is `-ib`. Use `-i ''` to indicate no suffix.

`-h HOSTS` - List of hosts to setup.

`-f hostfile` - File with hosts in cluster, default is `/etc/sysconfig/iba/hosts`.

`-u user` - User on remote system to verify this user can `rsh` to. The default is current user code.

4.3.2.3 Example

```
check_rsh
check_rsh -h 'arwen elrond'
HOSTS='arwen elrond' check_rsh
```

4.3.2.4 Environment Variables

The following environment variables are also used by this command:

`HOSTS`, `HOSTS_FILE` - See ["Selection of Hosts" on page 23](#).

4.3.3 setup_ssh

(Linux or Switch) creates `ssh` keys and configures them on all hosts or chassis so the system can `ssh` and `scp` into all other hosts or chassis without a password prompt. Typically, during cluster setup this tool is used to enable the root user on the Fabric Management node to login to the other hosts as root or chassis as admin using password-less `ssh`. However, if desired, this tool can also aid the setup of password-less `ssh` login for other user codes as well.



4.3.3.1 Usage

```
setup_ssh [-C] [-U] [-s] [-f hostfile] [-F chassisfile] [-h 'HOSTS'] [-H 'chassis']  
[-i ipoib_suffix] [-u user] [-S] [-R] [-p] [-P]
```

or

```
setup_ssh --help
```

4.3.3.2 Options

- help – Produce full help text
- C – Perform operation against chassis, default is hosts.
- U – Only perform connect (to enter in local hosts, known hosts). When run in this mode, -S and -s options are ignored).
- s – Use ssh/scp to transfer files, default is rsh/rcp.
- f *hostfile* – File with hosts in cluster, default is /etc/sysconfig/iba/hosts.
- F *chassisfile* – File with chassis in cluster default is /etc/sysconfig/iba/chassis
- h *HOSTS* – List of hosts to setup.
- H *chassis* – List of chassis to ping
- i *ipoib_suffix* – Suffix to apply to host names to create IPoIB host names. The default is -ib.
- u *user* – User on remote system to allow this user to ssh to, default is current user code.
- S – Securely prompt for password for user on remote system.
- R – Skip setup of ssh to localhost.
- p – Perform operation against all chassis or hosts in parallel.
- P – Skip ping of host (for ssh to devices on internet with ping firewalled).

4.3.3.3 Example

Operations on hosts:

```
setup_ssh -s -S -i ''  
setup_ssh -U  
setup_ssh -h 'arwen elrond' -U  
HOSTS='arwen elrond' setup_ssh -U
```

Operations on chassis:

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




4.3.3.4 Environment Variables

The following environment variables are also used by this command:

`HOSTS`, `HOSTS_FILE` – See discussion on “Selection of Hosts” on page 23.

`CHASSIS`, `CHASSIS_FILE` – See discussion on “Selection of Chassis” on page 24.

`FF_IPOIB_SUFFIX` – Suffix to append to hostname to create IPoIB hostname. Used in absence of `-i`.

`FF_CHASSIS_LOGIN_METHOD` – How to login to chassis. Can be ssh or telnet

`FF_CHASSIS_ADMIN_PASSWORD` – Password for admin on all chassis. Used in absence of `-S` option.

Intel® FastFabric Toolset provides additional flexibility in the translation between IPoIB and management network hostnames. Refer to “Configuration of IPoIB Name Mapping” in the *Intel® True Scale Fabric Suite FastFabric User Guide* for more information.

`setup_ssh` provides an easy way to create ssh keys and distribute them to the hosts or chassis in the cluster. Many of the FastFabric tools (as well as many versions of MPI) require ssh be set up for password-less operation. Therefore, `setup_ssh` is an important setup step.

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

The tool will also configure the local host ssh on all servers. All servers need to be configured with the latest OFED+ Host Software release, otherwise the localhost step will be skipped.

Note: The `-R` option disables the local host configuration.

Note: The `-i` option and environment variables control whether local IPoIB loopback ssh is also configured.

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

Note: The meaning of the `-C` option has changed, in previous releases `-C` selected the mode of operation (`-U` now serves that purpose).

4.3.3.5 Host Initial Key Exchange

When run without the `-U` option, `setup_ssh` will perform the initial key exchange and enable password-less ssh and scp. The key exchange can be accomplished for hosts using ssh and scp (in a password prompting manner) using the `-s` option or using password-less rsh and rcp (omitting the `-s` option).

The preferred way to use `setup_ssh` for initial key exchange is with the `-s` and `-S` options. This requires all hosts have been configured with the same password for the specified user (typically `root`). In this mode the password will be prompted for once and then ssh and scp are used in conjunction with that password to complete the set up for the hosts. Use in this manner also avoids the need to setup rsh/rcp/rlogin (which can be a security risk).



If `-s` is used without the `-S` option, the user will be prompted by ssh and scp for each host as they are set up. There will be multiple prompts per host. For a handful of hosts this is manageable, however for a significant number of hosts this can become cumbersome. Therefore, the `-S` option is recommended in this case.

If the `-s` option is not specified, `rsh` and `rcp` will be used to perform the ssh key exchange. This requires password-less `rcp` and `rlogin` to be enabled on each host (`check_rsh` can perform verification).

`setup_ssh` will configure password-less ssh/scp for both the management network and IPoIB. Typically, the management network will be used for Intel® FastFabric Toolset operations while IPoIB will be used for MPI and other applications. If IPoIB is not yet running (for example, during initial cluster installation True Scale Fabric software will not yet be installed on all the hosts), the `-i` option can be specified with an empty string:

```
setup_ssh -i ''
```

This will cause the last part of the setup of ssh for IPoIB to be skipped.

4.3.3.6 Host Refreshing Local Systems Known Hosts

If hosts have IP addresses added (for example, by installing True Scale Fabric software and enabling IPoIB), IP address changes, MAC addresses changed or other aspects have changed (such as server OS reinstallation), the local hosts `ssh known_hosts` file can be refreshed by running `setup_ssh` with the `-U` option. This option will not transfer the keys, but rather will connect to each host (management network and IPoIB) in order to refresh the ssh keys. Existing entries for the specified hosts are replaced within the local `known_hosts` file. When run in this mode the `-S` and `-s` options are ignored. This mode assumes ssh has previously been setup for the hosts, as such no files are transferred to the specified hosts and no passwords should be required.

Typically after completing the installation and booting of True Scale Fabric software, `setup_ssh` will need to be rerun with the `-U` option to update the `known_hosts` file

4.3.3.7 Chassis Initial Key Exchange

When run without the `-U` option, `setup_ssh` will perform the initial key exchange and enable password-less ssh and scp. For chassis, the key exchange uses scp and the chassis CLI. Login to the chassis during this command will be through the configured mechanism for chassis login. Typically the `-S` option should be used when doing initial setup of ssh keys for a chassis. For chassis the `-s` option is ignored

The preferred way to use `setup_ssh` for initial key exchange is with the `-S` option. This requires all chassis have been configured with the same password for the specified user (typically `admin`). In this mode the password will be prompted for once and then the `FF_CHASSIS_LOGIN_METHOD` and `scp` are used in conjunction with that password to complete the setup for the chassis. Use in this manner also avoids the need to setup the chassis password in `fastfabric.conf` (which can be a security risk).

For chassis the `-i` option is ignored.

4.3.3.8 Chassis Refreshing Local Systems Known Hosts

If chassis have IP addresses changed, MAC addresses changed or other aspects have changed, the local hosts `ssh known_hosts` file can be refreshed by running `setup_ssh` with the `-U` option. This option will not transfer the keys, but rather will connect to each chassis in order to refresh the ssh keys. Existing entries for the specified chassis are replaced within the local `known_hosts` file. When run in this



mode the `-S` options is ignored. This mode assumes ssh has previously been setup for the chassis, as such no files are transferred to the specified hosts and no passwords should be required.

4.3.4 cmdall

(Linux and Switch) Executes a command on all hosts or Intel® Chassis. This is very powerful and can be used for everything from configuring servers or chassis, verifying that they are running, starting and stopping host processes, etc.

4.3.4.1 Usage

```
cmdall [-Cpq] [-f hostfile] [-F chassisfile] [-h 'hosts'] [-H 'chassis'] [-u user]
[-S] [-m 'marker'] [-T timelimit] [-P] 'cmd'
```

or

```
cmdall --help
```

4.3.4.2 Options

- `--help` - Produce full help text
- `-C` - Perform command against chassis, default is hosts
- `-p` - Run command in parallel on all hosts
- `-q` - Quiet mode, do not show command to execute
- `-f hostfile` - File with hosts in cluster, default is `/etc/sysconfig/iba/hosts`
- `-F chassisfile` - File with chassis in cluster default is `/etc/sysconfig/iba/chassis`
- `-u user` - The user to perform the command as. For hosts, the default is current user code. For chassis, the default is `admin` (this argument is ignored)
- `-S` - Securely prompt for password for admin on chassis
- `-m 'marker'` - Marker for end of chassis command output if omitted defaults to chassis command prompt this may be a regular expression
- `-T timelimit` - Time limit in seconds when running host commands default is -1 (infinite)
- `-P` - Output hostname/chassis name as prefix to each output line this can make script processing of output easier

4.3.4.3 Host Examples

```
cmdall date
cmdall 'uname -a'
cmdall -h 'elrond arwen' date
HOSTS='elrond arwen' cmdall date
```

4.3.4.4 Chassis Examples

```
cmdall -C 'ismPortStats -noprompt'
```



```
cmdall -C -H 'chassis1 chassis2' ismPortStats -noprompt'  
CHASSIS='chassis1 chassis2' cmdall ismPortStats -noprompt'
```

4.3.4.5 Environment Variables

The following environment variables are also used by this command:

HOSTS, *HOSTS_FILE* – See discussion on “Selection of Hosts” on page 23.

CHASSIS, *CHASSIS_FILE* – See discussion on “Selection of Chassis” on page 24.

FF_MAX_PARALLEL – When *-p* option is used, maximum concurrent operations will be performed.

FF_CHASSIS_LOGIN_METHOD – How to login to chassis. Can be ssh or telnet

FF_CHASSIS_ADMIN_PASSWORD – Password for admin on all chassis. Used in absence of *-S* option.

Note:

All commands performed with *cmdall* must be non-interactive in nature. *cmdall* will wait for the command to complete before proceeding. For example, when running host commands such as *rm*, the *-i* option (interactively prompt before removal) should not be used (Note that this option is sometimes part of a standard bash alias list). Similarly, when running chassis commands such as *fwUpdateChassis*, the *-reboot* option should not be used (this option causes an immediate reboot therefore, the command never returns). Similarly, the chassis command *reboot* should not be executed using *cmdall*. Instead use the *iba_chassis_admin reboot* FastFabric tool command to reboot one or more chassis. For further information about individual chassis CLI commands consult the *Intel® True Scale Fabric Switches 12000 Series CLI Reference Guide* or *SilverStorm 9000 CLI Reference Guide*. For further information about Linux OS commands, consult the Linux man pages and any other documentation supplied with the OS by the OS supplier.

When performing *cmdall* against hosts, internally ssh is used. The command *cmdall* requires that password-less ssh be setup between the host running Intel® FastFabric Toolset and the hosts *cmdall* is operating against. The *setup_ssh* FastFabric tool can aid in setting up password-less ssh.

When performing *cmdall* against a set of chassis, all chassis must be configured with the same admin password or the *setup_ssh* FastFabric tool can be used to setup password-less ssh to the chassis.

When performing operations against chassis, set up of ssh keys is recommended (see “*setup_ssh*” on page 159). If ssh keys are not setup, use of the *-S* option is recommended. This avoids the need to keep the password in configuration files.

4.3.5 captureall

(Switch and Host) Captures supporting information for a problem report from all hosts or Intel® Chassis and uploads to this system

4.3.5.1 Usage

```
captureall [-Cnp] [-f hostfile] [-F chassisfile] [-L nodefile] [-h 'HOSTS'] [-H  
'chassis'] [-N 'nodes'] [-t portsfile] [-d upload_dir] [-S] [-D detail_level] [-n  
hosts] [file]
```

or

```
captureall --help
```



4.3.5.2 Options

- help – Produce full help text
- C – Perform capture against chassis, default is hosts
- n – Perform capture against nodes, default is hosts
- p – Perform capture in parallel (for a host capture this only affects the upload phase)
- f *hostfile* – File with hosts in cluster, default is /etc/sysconfig/iba/hosts
- F *chassisfile* – File with chassis in cluster, default is /etc/sysconfig/iba/chassis
- L *nodefile* – File containing a list of the nodes in the cluster. The default file is /etc/sysconfig/iba/ibnodes
- h *host* – List of hosts from which to perform a capture
- H *chassis* – List of chassis to perform a capture of
- N *nodes* – List of nodes to execute operation against.
- t *portsfile* – File with list of local HCA ports used to access fabric(s) for switch access, default is /etc/sysconfig/iba/ports
- d *upload_dir* – Directory to upload to, default is uploads. If not specified, the environment variable UPLOADS_DIR is used. If that is not exported, the default (./uploads) will be used.
- S – Securely prompt for password for administrator on a chassis
- D *detail_level* – Level of detail of capture (only used for host captures, ignored for chassis capture)
 - The Details levels are:
 - 1 – Default – Obtains local information from each host
 - 2 – Fabric – In addition to “Default”, also obtains basic fabric information by queries to the SM and fabric error analysis using *iba_report*.
 - 3 – Fabric+FDB – In addition to “Fabric”, also obtains all the switch forwarding tables from the SM.
 - 4 – Analysis – In addition to “Fabric+FDB”, also obtains all_analysis results. If all_analysis has not yet been run, it is run as part of the capture.
- file* – Name for capture file (if the filename given does not have a .tgz suffix, .tgz will be appended)

When a host captureall is performed, *iba_capture* will be run to create the specified capture file within *~root* on each host (with the .tgz suffix added as needed). The files will be uploaded and unpacked into a matching directory name within *upload_dir/hostname/* on the local system. The default file name is *hostcapture*.

Note: Detail levels 2-4 can be used when fabric operational problems occur. If the problem is most likely node specific, detail level 1 should be sufficient. Detail levels 2-4 require an operational Fabric Manager. Typically your support representative will request a given detail level. If a given detail level takes excessively long or fails to be gathered, try a lower detail level.

For detail levels 2-4, the additional information is only gathered on the node running the captureall command. The information is gathered for every fabric specified in the */etc/sysconfig/iba/ports* file.



When a chassis capture all is performed, the `chassis capture` CLI command will be run on each chassis and its output will be saved to `upload_dir/chassisname/file` on the local system. The default file name is `chassiscapture`.

For both host and chassis capture, the uploaded captures will be combined into a `tgz` file with the file name specified and the suffix `.all.tgz` added

4.3.5.3 Host Capture Examples

```
captureall
```

The above example creates a `hostcapture` directory in `/uploads/hostname/` for each host in `/etc/sysconfig/iba/hosts` then creates `hostcapture.all.tgz`.

```
captureall mycapture
```

The above example creates a `mycapture` directory in `./uploads/hostname/` for each host in `/etc/sysconfig/iba/hosts` then creates `mycapture.all.tgz`.

```
captureall -h 'arwen elrond' 030127capture
```

4.3.5.4 Chassis Capture Examples

```
captureall -C
```

The above example creates a `chassiscapture` file in `./uploads/chassisname/` for each chassis in `/etc/sysconfig/iba/chassis` then creates `chassiscapture.all.tgz`.

```
captureall -C mycapture
```

The above example creates a `mycapture.tgz` file in `./uploads/chassisname/` for each chassis in `/etc/sysconfig/iba/chassis` then creates `mycapture.all.tgz`.

```
captureall -C -H 'chassis1 chassis2' 030127capture
```

4.3.5.5 Environment Variables

The following environment variables are also used by this command:

HOSTS, HOSTS_FILE – See discussion on “[Selection of Hosts](#)” on page 23.

CHASSIS, CHASSIS_FILE – See discussion on “[Selection of Chassis](#)” on page 24.

IBNODES – List of nodes, used if `-n` used and `-N` and `-L` option not supplied

UPLOADS_DIR – Directory to upload to, used in absence of `-d`.

FF_MAX_PARALLEL – When `-p` option is used, maximum concurrent operations will be performed.

FF_CHASSIS_LOGIN_METHOD – How to login to chassis. Can be SSH or telnet.

FF_CHASSIS_ADMIN_PASSWORD – Password for administrator on all chassis. Used in absence of `-S` option.

When performing `captureall` against hosts, internally SSH is used. The command `captureall` requires that password-less SSH be setup between the host running Intel® FastFabric Toolset and the hosts `captureall` is operating against. The `setup_ssh` FastFabric tool can aid in setting up password-less SSH.



When performing operations against chassis, set up of ssh keys is recommended (see “[setup_ssh](#)” on page 159). If ssh keys are not setup, all chassis must be configured with the same admin password and use of the `-S` option is recommended. The `-S` option avoids the need to keep the password in configuration files.

Note: The resulting host capture files can require significant amounts of space on the Intel® FastFabric Toolset host. Actual size will vary, but sizes can be multiple megabytes per host. As such it is recommended to ensure adequate space is available on the Intel® FastFabric Toolset system. In many cases it may not be necessary to run `captureall` against all hosts or chassis, but rather a representative subset may be sufficient. Consult with your support representative for further information.

4.4 File Management Tools

The following tools aid in copying files to and from large groups of nodes in the fabric.

Internally, these tools make use of SCP and require that password-less SSH/SCP be setup between the host running Intel® FastFabric Toolset and the hosts files that are being transferred to and from. The `setup_ssh` FastFabric tool can aid in setting up password-less SSH/SCP.

4.4.1 `scpall`

(Linux) The `scpall` tool permits efficient copying of files or directories from the current system to multiple hosts in the fabric. When copying large directory trees, performance can be improved by using the `-t` option. This will tar and compress the tree, then transfer the resulting compressed tarball to each node (and untar it on each node).

This can provide a powerful facility for copying data files, operating system files or even applications to all the hosts (or a subset of hosts) within the fabric.

4.4.1.1 Usage

```
scpall [-p] [-r] [-f hostfile] [-h 'HOSTS'] [-u user] source_file ... dest_file
```

```
scpall -t [-p] [-f hostfile] [-h 'HOSTS'] [-u user] [source_dir [dest_dir]]
```

or

```
scpall --help
```

4.4.1.2 Options

`--help` – Produce full help text

`-r` – Recursive copy of directories

`-p` – Perform copy in parallel

`-t` – Optimized recursive copy of directories using tar

`-h HOSTS` – List of hosts to copy to

`-f hostfile` – File with hosts in cluster, default is `/etc/sysconfig/iba/hosts`.

`-u user` – User to perform copy to, default is current user code

`source_file` – Name of files to copy from this system, relative to the current directory. Multiple files may be listed.



source_dir – Name of directory to copy from this system, relative to the current directory.

dest_file or *dest_dir* – Name of the file or directory on the destination system to copy to. It is relative to the home directory of the specified user code (an absolute path name may be specified if desired).

When performing directory copies using the `-t` option, the destination directory is optional. If not specified it defaults to the present directory name. If both the source and destination directory names are omitted, they both default to the current directory name.

4.4.1.3 Example

```
# copy a single file
scpall MPI-PMB /root/MPI-PMB

# efficiently copy an entire directory tree
scpall -p -t /opt/iba/src/mpi_apps /opt/iba/src/mpi_apps

# copy a group of files
scpall a b c /root/tools/

# copy to an explicitly specified set of hosts
scpall -h 'arwen elrond' a b c /root/tools

HOSTS='arwen elrond' scpall a b c /root/tools
```

Note: The tool `scpall` can only copy from this system to a group of systems in the cluster. The `user@` style syntax cannot be used in the arguments to `scpall`.

4.4.1.4 Environment Variables

The following environment variables are also used by this command:

HOSTS, *HOSTS_FILE* – See discussion on “[Selection of Hosts](#)” on page 23.

FF_MAX_PARALLEL – When `-p` option is used maximum concurrent operations will be performed.

To copy from hosts in the cluster to this host, use `uploadall`.

4.4.2 uploadall

(Linux) Copies one or more files from a group of hosts to this system. Since the file name will be the same on each host, a separate directory on this system is created for each host and the file is copied to it. This is a convenient way to upload log files or configuration files for review. It can also be used in conjunction with `downloadall` to upload a host specific configuration file, edit it for each host and download the new version to all the hosts.

4.4.2.1 Usage

```
uploadall [-rp] [-f hostfile] [-d upload_dir] [-h 'HOSTS'] [-u user] source_file
... dest_file
```

or



```
uploadall --help
```

4.4.2.2 Options

--help – Produce full help text

-p – Perform copy in parallel on all hosts

-r – Recursive upload of directories

-f *host_file* – File with hosts in cluster, default is /etc/sysconfig/iba/hosts

-h *HOSTS* – List of hosts to upload from

-u *user* – User to perform copy to, default is current user code

-d *upload_dir* – Directory to upload to, default is uploads. If not specified the environment variable UPLOADS_DIR will be used, if that is not exported the default, /uploads, will be used.

source_file – Name of files to copy to this system, relative to the current directory. Multiple files may be listed.

dest_file – Name of the file or directory on this system to copy to. It is relative to upload_dir/*hostname*.

A local directory within upload_dir/ will be created for each host being uploaded from. Each uploaded file will be copied to upload_dir/*hostname*/*dest_file*. If more than one source file is specified, *dest_file* will be treated as a directory name and the directories upload_dir/*hostname*/*dest_file*/ will be created for each host and the *source_files* will be uploaded to those directories.

4.4.2.3 Example

```
# upload two files from 2 hosts
uploadall -h 'arwen elrond' capture.tgz /etc/init.d/ipoib.cfg

# upload two files from all hosts
uploadall capture.tgz /etc/sysconfig/ifs_fm.xml

# upload network config files from all hosts
uploadall -r -p /etc/sysconfig/network-scripts network-scripts

# upload two files to a specific subdirectory of upload_dir
uploadall capture.tgz /etc/sysconfig/ifs_fm.xml pre-install
```

The above example copies capture.tgz and /etc/sysconfig/ifs_fm.xml to /uploads/*hostname*/preinstall/ where a *hostname* directory is created for each host in /etc/sysconfig/iba/hosts.

Note: The uploadall tool can only copy from a group of systems in a cluster to this system. The user@ style syntax cannot be used in the arguments to uploadall.

To copy files from this host to hosts in the cluster use scpall or downloadall.

4.4.2.4 Environment Variables

The following environment variables are also used by this command:



HOSTS, *HOSTS_FILE* – See discussion on “Selection of Hosts” on page 23.

FF_MAX_PARALLEL – When *-p* option is used maximum concurrent operations will be performed.

UPLOADS_DIR – Directory to upload to, used in absence of *-d*.

4.4.3 downloadall

(Linux) Copies one or more files to a group of hosts from a system. Since the file contents to copy may be different for each host, a separate directory on this system is used for the source files for each host. This can also be used in conjunction with *uploadall* to upload a host-specific configuration file, edit it for each host and download the new version to all the hosts.

4.4.3.1 Usage

```
downloadall [-rp] [-f hostfile] [-d download_dir] [-h 'HOSTS'] [-u user]  
source_file ... dest_file
```

or

```
downloadall --help
```

4.4.3.2 Options

--help – Produce full help text

-p – Perform copy in parallel on all hosts

-r – Recursive download of directories

-f hostfile – File with hosts in cluster. The default is */etc/sysconfig/iba/hosts*.

-h HOSTS – List of hosts to download files to

-u user – User to perform the copy. The default is current user code

-d download_dir – Directory to download files to. The default is *./downloads*. If not specified, the environment variable *DOWNLOADS_DIR* will be used. If that is not exported the default (*./downloads*) will be used.

source_file – Name of files to copy from the system. Multiple files may be listed. The option *source_file* is relative to *download_dir/hostname*.

A local directory within *download_dir/* must exist for each host being downloaded to each downloaded file will be copied from *download_dir/hostname/source_file*.

dest_file – Name of the file or directory on the destination hosts to copy to.

If more than one source file is specified, *dest_file* will be treated as a directory name. The given directory must already exist on the destination hosts (the copy will fail for hosts where the directory does not exist).

4.4.3.3 Example

```
# copy two files to 2 hosts
```

```
downloadall -h 'arwen elrond' ics_srp.cfg ics_inic.cfg /etc/sysconfig
```



```
# copy two files to all hosts
downloadall ics_srp.cfg ics_inic.cfg /etc/sysconfig
```

Note: The tool `downloadall` can only copy from this system to a group of hosts in the cluster. The `user@` style syntax cannot be used in the arguments to `downloadall`.

To copy files from hosts in the cluster to this host use `uploadall`.

4.4.3.4 Environment Variables

The following environment variables are also used by this command:

`HOSTS`, `HOSTS_FILE` – See discussion on “Selection of Hosts” on page 23.

`FF_MAX_PARALLEL` – When `-p` option is used maximum concurrent operations will be performed.

`DOWNLOADS_DIR` – Directory to download from, used in absence of `-d`.

4.4.4 Simplified Editing of Node-Specific Files

(Linux) The combination of `uploadall` and `downloadall` provide a powerful yet simple to use mechanism for reviewing and/or editing node-specific files without the need to login to each node.

This is best explained with an example.

Assume the file `/etc/sysconfig/network-scripts/ifcfg-ib1` needs to be reviewed and possibly edited for each host. This file would typically contain the IP configuration information for IPoIB and may contain a unique IP address per host.

To upload the file from all the hosts:

```
uploadall /etc/sysconfig/network-scripts/ifcfg-ib1 ifcfg-ib1
```

Now edit the uploaded files with an editor, such as `vi`:

```
vi uploads/*/ifcfg-ib1
```

If through the editor, the file was changed for some or all of the hosts, it can then be downloaded to all the hosts:

```
downloadall -d uploads ifcfg-ib1 /etc/sysconfig/network-scripts/ifcfg-ib1
```

Alternatively, if there was no need to download the file to all hosts, a subset of hosts can be specified using the `-h` option or by creating an alternate host list file:

```
downloadall -d uploads -h 'host1 host32' ifcfg-ib1
/etc/sysconfig/network-scripts/ifcfg-ib1
```

Note: When downloading to a subset of hosts, make sure that only the hosts uploaded from are specified.

4.4.5 Simplified Setup of Node-Generic Files

(Linux) In contrast `scpall` can provide a powerful yet simple to use mechanism for transferring files to all nodes that are generic (for example, not node-specific).

For example, if all nodes in the cluster will use the same DNS server and TCP/IP name resolution, they may be quickly set as follows:



Create an appropriate local file with the desired information. For example:

```
vi resolv.conf
```

Now copy the file to all hosts:

```
scpall resolv.conf /etc/resolv.conf
```

4.5 Fabric Analysis Tools

4.5.1 fabric_info

`fabric_info` provides a brief summary of the components in the fabric. `fabric_info` uses the first active True Scale Fabric port on the given local host to perform its analysis. `fabric_info` is supplied as part of both Intel® FastFabric Toolset and the Intel® True Scale Fabric Tools. When provided as part of Intel® FastFabric Toolset it can support management of more than 1 fabric (subnet). When provided as part of the Intel® True Scale Fabric Tools it will only perform analysis against the 1st active port on the system.

4.5.1.1 Usage

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

or

```
fabric_info --help
```

`fabric_info` has no options and uses no environment variables when run as part of the Intel® True Scale Fabric Tools.

4.5.1.2 Options

`--help` – Produce full help text

`-t portsfile` – File with list of local HCA ports used to access fabric(s) for analysis. The default is `/etc/sysconfig/iba/ports`.

`-p ports` – List of local HCA ports used to access fabric(s) for analysis. The default is the first active port.

This is specified as **HCA : port**:

0:0 – 1st active port in system

0:y – Port y within system

x:0 – 1st active port on HCA x

x:y – HCA x, port y

for example:

```
fabric_info
```

```
fabric_info -p '1:1 1:2 2:1 2:2'
```

4.5.1.3 Environment Variables

The following environment variables are also used by this command:

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

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



For simple fabrics, the Intel® FastFabric Toolset host would be connected to a single fabric. By default the first active port on the Intel® FastFabric Toolset host will be used to analyze the fabric.

However, in more complex fabrics, the Intel® FastFabric Toolset host may be connected to more than one fabric (or subnet). In this case the specific ports and/or HCAs to use for fabric analysis may be specified.

Specification of the ports to be used can be performed on the command line using the `-p` option, in a file specified using the `-t` option, through the environment variables `PORTS` or `PORTS_FILE`, or using the `ports_file` configuration option in `fastfabric.conf`. If the specified file does not exist or is empty, the first active port on the local system will be used. In more complex configurations (such as where the Intel® FastFabric Toolset host is connected to multiple True Scale Fabrics or subnets), the user will need to specify the exact ports to use such that all fabrics are analyzed. For more information, refer to ["Selection of local Ports \(subnets\)" on page 28](#).

4.5.1.4 Example output

```
# fabric_info

Fabric Information:

SM: i9k229 Guid: 0x00066a00d8000229 State: Master
SM: i9k3ff Guid: 0x00066a00d90003ff State: Standby

Number of CAs: 17

Number of CA Ports: 22

Number of Switch Chips: 6

Number of Links: 29

Number of 1x Ports: 2
```

4.5.1.5 The output is as follows

SM – Each subnet manager (SM) running in the fabric is listed along with its node name, port GUID and present SM state (Master, Standby, etc).

Number of CA – Number of unique channel adapters (CA) in the fabric. A CA with two-connected ports is counted as a single CA.

Note: Channel adapters include both HCAs in servers as well as TCAs within IO Modules, Native Storage, etc.

Number of CA ports – Number of connected CA ports in the fabric.

Number of Switch chips – Number of unique switches in the fabric.

Note: A large switch may be composed of many unique switch chips.

Number of Links – Number of links in the fabric. Note that a large switch may have internal links.

Number of 1x Ports – Number of ports in the fabric running at 1x speed. Typically such ports represent a bad cable connection, a bad cable, too long a cable or perhaps faulty hardware on one side of the link.



`fabric_info` can be very useful as a quick assessment of the fabric state. `fabric_info` can be run against a known good fabric to identify its components and then later run to see if anything has changed about the fabric configuration or state. When used in this manner it can be used to quickly identify if CAs are down, links are missing, SMs are missing, etc.

For more extensive fabric analysis, see [“iba_report” on page 175](#) and [“iba_reports” on page 212](#).

4.5.2 showallports

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

Note: `iba_report` and `iba_reports` are newer and more powerful Intel® FastFabric Toolset commands. For general fabric analysis, use `iba_report` or `iba_reports` with options such as `-o errors` and/or `-o slowlinks` to perform a more efficient analysis of link speeds and errors.

4.5.2.1 Usage

```
showallports [-C|-I] [-f hostfile] [-F chassisfile] [-L ibnodefile] [-h 'HOSTS']  
[-H 'chassis'] [-N 'ibnodes'] [-M 'host'] [-S]
```

or

```
showallports --help
```

4.5.2.2 Options

- `--help` – Produce full help text
- `-C` – Perform operation against chassis; the default is `hosts`
- `-I` – Perform operation against nodes; the default is `hosts`
- `-f hostfile` – File with hosts in cluster; the default is `/etc/sysconfig/iba/hosts`
- `-F chassisfile` – File with chassis in cluster; the default is `/etc/sysconfig/iba/chassis`
- `-L ibnodefile` – File with ib nodes in the cluster; the default is `/etc/sysconfig/iba/ibnodes`
- `-h HOSTS` – List of hosts to show port information
- `-H chassis` – List of chassis to show port information
- `-N ibnodes` – List of nodes to show port information
- `-M HOST` – Management host. This is the remote host from which to run node queries; the default is `localhost`
- `-S` – Securely prompt for password for administrator on chassis

4.5.2.3 Example

```
showallports  
  
showallports -h 'elrond arwen'
```



```
HOSTS='elrond arwen' showallports
showallports -C
showallports -H 'chassis1 chassis2'
CHASSIS='chassis1 chassis2' showallports
showallports -I
showallports -I -N '0x00066a0005000105 0x00066a0005000110'
IBNODES='0x00066a0005000105 0x00066a0005000110' showallports -I
```

4.5.2.4 Environment Variables

The following environment variables are also used by this command:

HOSTS, *HOSTS_FILE* – See discussion on “[Selection of Hosts](#)” on page 23.

CHASSIS, *CHASSIS_FILE* – See discussion on “[Selection of Chassis](#)” on page 24.

IBNODES, *IBNODES_FILE* – See discussion on “[Selection of Switches](#)” on page 26.

MGMT_HOST – Host to use to perform node queries, used in absence of *-M*

FF_CHASSIS_LOGIN_METHOD – How to login to chassis. Can be SSH or Telnet

FF_CHASSIS_ADMIN_PASSWORD – Password for the administrator on all chassis. Used in absence of *-S* option.

When performing *showallports* against hosts, internally SSH is used. *showallports* requires that password-less SSH be setup between the host running Intel® FastFabric Toolset and the hosts *showallports* is operating against. The *setup_ssh* FastFabric tool can aid in setting up password-less SSH.

For operations against chassis, setup of ssh keys (see “[setup_ssh](#)” on page 159) is recommended. If ssh keys are not setup, all chassis must be configured with the same admin password and use of the *-S* option is recommended. The *-S* option avoids the need to keep the password in configuration files.

When performing *showallports* against externally-managed switches it requires an Fabric Management Node with Intel® FastFabric Toolset installed. Typically this will be the Intel® FastFabric Toolset node from which *showallports* is being run. However, if desired an alternate node may be specified by the *-M* option or *MGMT_HOST* environment variable.

4.5.3 iba_report

(All) *iba_report* provides powerful fabric analysis and reporting capabilities. It must be run on a host connected to the True Scale Fabric with Intel® FastFabric Toolset installed.

iba_report obtains all its data in an IBTA-compliant manner. Therefore, it will interoperate with both Intel® and 3rd party components with Infiniband* Technology, provided those components are IBTA compliant and implement the IBTA optional features required by *iba_report*.

iba_report requires that the subnet manager implement all the IBTA SA queries defined in the standard (such as SM Info records, Link Records, Trace Routes, Port Records, Node Records, etc). As such, it is recommended that the Intel® True Scale Fabric Suite Fabric Manager (FM) version 4.0 or later be used. *iba_report* requires all



end nodes to implement the PMA PortCounters (IBTA mandatory counters). Also any end nodes which report support of a IBTA device management agent must implement the IOU Info, IOC Profile and Service Entry queries as outlined in the IBTA 1.1 standard.

`iba_report` also supports operation with the Intel® FM with the PM/PA. When `iba_report` detects the presence of a PA, it automatically issues any required PortCounter queries and clears to the PA to access the PMs running totals. If a PA is not detected, then `iba_report` will directly access the PMAs on all the nodes. The `-M` option can force access to the PMA even if a PA is present.

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

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

Each run of `iba_report` obtains up to date information from the fabric. At the start of the run `iba_report` will take a few seconds to obtain all the fabric data, then it will output it to `stdout`. The reports are sorted by GUIDs and other permanent information such that they can be rerun in the future and produce output in the same order even if components have been rebooted. This is useful for comparison using simple tools like `diff`. `iba_report` permits multiple reports to be requested for a single run (for example, 1 of each report type).

By default `iba_report` uses the first active port on the local system. However, if the Fabric Management Node is connected to more than one fabric (for example, a subnet), the HCA and port may be specified to select the fabric to analyze.

4.5.3.1 Usage

```
iba_report [-v][-q] [-h hca] [-p port] [-o report] [-d detail] [-P|-H] [-N] [-x]
[-X snapshot_input] [-T topology_input] [-s] [-r] [-V] [-i seconds] [-C] [-a] [-M]
[-c file] [-L] [-F point] [-S point] [-D point] [-Q] [-e]
```

or

```
iba_report --help
```

4.5.3.2 Options

- `--help` - Produce full help text
- `-v/--verbose` - Verbose output
- `-q/--quiet` - Disable progress reports
- `-h/--hca hca` - HCA to send by, default is first HCA
- `-p/--port port` - Port to send by, default is first active port
- `-o/--output report` - Report type for output



-d/--detail *level* - Level of detail 0-n for output, default is 2

-P/--persist - Only include data persistent across reboots

-H/--hard - Only include permanent hardware data

-N/--noname - Omit node and IOC names

-x/--xml - Output in XML

-X/--infile *snapshot_input* - Generates a report using the data within the *snapshot_input*. The *snapshot_input* must have been generated during a previous -o *snapshot* run. When -X *snapshot_input* is used, the -s, -r, -i, -C, and -a options are ignored. The -X *snapshot_input* option can be used with -o *route* (when run against a snapshot generated with the -r option) and -F **route**. '-' may be used as the *snapshot_input* to specify stdin.

-T/--topology *topology_input* - Use *topology_input* file to augment and verify fabric information. When used various reports can be augmented with information not available electronically (such as cable labels and lengths). '-' may be used to specify stdin

-s/--stats - Get performance statistics for all ports

-r/--routes - Get routing tables for all switches

-V/--vltables - Get QOS VL-related tables for all ports

-i/--interval *seconds* - Obtain performance statistics over interval seconds, clears all statistics, waits interval seconds, then generates report. Implies -s

-C/--clear - Clear performance stats for all ports. Only stats with error thresholds are cleared. A clear occurs after generating the report.

-a/--clearall - Clear all performance stats for all ports

-M/--pmdirect - Access performance stats via direct PMA

-c/--config *file* - Error thresholds configuration file. The default is /etc/sysconfig/iba/iba_mon.conf

-L/--limit - For port error counters check (-o *errors*) and port counters clear (-C or -i) with -F limit operation to exact specified focus. Normally the neighbor of each selected port would also be checked/cleared does not affect other reports

-F/--focus *point* - Focus area for report used for all reports except route to limit scope of report

-S/--src *point* - Source for trace route, default is local port

-D/--dest *point* - Destination for trace route

-Q/--quietfocus - Do not include focus description in report

-e/--ehcapma - Enable PMA query of eHCA logical CAs

4.5.3.3 Report Types

comps - Summary of all systems and SMs in fabric

brcomps - Brief summary of all systems and SMs in fabric



`nodes` – Summary of all node types and SMs in fabric

`brnodes` – Brief summary of all node types and SMs in fabric

`ious` – Summary of all IO units in the fabric

`lids` – Summary of all lids in the fabric

`links` – Summary of all links

`extlinks` – Summary of links external to systems

`slowlinks` – Summary of links running slower than expected

`slowconfiglinks` – Summary of links configured to run slower than supported, includes `slowlinks`

`slowconnlinks` – Summary of links connected with mismatched speed potential, includes `slowconfiglinks`

`misconfiglinks` – Summary of links configured to run slower than supported

`misconnlinks` – Summary of links connected with mismatched speed potential

`errors` – Summary of links whose errors exceed counts in the configuration file (implies `-s`)

`otherports` – Summary of ports not connected to the fabric

`linear` – Summary of linear forwarding tables in all switches in the fabric (implies `-r`)

`mcast` – Summary of multicast forwarding tables in all switches in the fabric (implies `-r`)

`portusage` – Summary of ports referenced in linear forwarding tables broken down by type of DLID (implies `-r`)

`pathusage` – Summary of number of CA to CA paths routed through each switch port

`treepathusage` – Analysis of number of CA to CA paths routed through each switch port for a fat tree

`validateroutes` – Validate all routes in the fabric

`vfinfo` – Summary of vFabric information

- `verifycas`** – Compare fabric (or snapshot) CAs to supplied topology and identify differences and omissions
- `verifysws`** – Compare fabric (or snapshot) Switches to supplied topology and identify differences and omissions
- `verifyrtrs`** – Compare fabric (or snapshot) Routers to supplied topology and identify differences and omissions
- `verifynodes`** – `verifycas`, `verifysws` and `verifyrtrs` reports
- `verifysms`** – Compare fabric (or snapshot) SMs to supplied topology and identify differences and omissions
- `verifylinks`** – Compare fabric (or snapshot) links to supplied topology and identify differences and omissions
- `verifyextlinks`** – Compare fabric (or snapshot) links to supplied topology and identify differences and omissions limit analysis to links external to systems
- `verifyall`** – `verifycas`, `verifysws`, `verifyrtrs`, `verifysms` and `verifylinks` reports



all - comp, nodes, ious, links, extlinks, slowconnlinks, and errors reports

route - Trace route between -S and -D points

snapshot - Output snapshot of the fabric state for later use as snapshot_input. This implies -x. May not be combined with other reports.

none - No report, useful if just want to clear statistics

4.5.3.4 Point Syntax

gid:value - value is numeric port gid of form: subnet:guid

lid:value - value is numeric lid

lid:value:node - value is numeric lid, selects entire node with given lid

lid:value:port:value2 - value is numeric lid of node, value2 is port number

portguid:value - value is numeric port GUID

nodeguid:value - value is numeric node GUID

nodeguid:value1:port:value2 - value1 is numeric node GUID, value2 is port number

iocguid:value - value is numeric IOC GUID

iocguid:value1:port:value2 - value1 is numeric IOC GUID, value2 is port number

systemguid:value - value is numeric system image GUID

systemguid:value1:port:value2 - value1 is the numeric system image GUID; value2 is port number

ioc:value - value is IOC Profile ID String (IOC Name)

ioc:value1:port:value2 - value1 is IOC Profile ID String (IOC Name); value2 is port number

iocpat:value - value is global pattern for IOC Profile ID String (IOC Name)

iocpat:value1:port:value2 - value1 is global pattern for IOC Profile ID String; (IOC Name), value2 is port number

ioctype:value - value is IOC type

ioctype:value1:port:value2 - value1 is IOC type ; value2 is port number

node:value - value is node description (node name)

node:value1:port:value2 - value1 is node description (node name); value2 is port number

nodepat:value - value is glob pattern for node description (node name)

nodepat:value1:port:value2 - value1 is the global pattern for the node description (node name); value2 is port number

nodedetpat:value - value is glob pattern for node details



nodedetpat: *value1:port:value2* – *value1* is the global pattern for the node details; *value2* is port number

nodetype: *value* – *value* is node type (SW, CA or RT)

nodetype: *value1:port:value2* – *value1* is node type (SW, CA or RT); *value2* is port number

rate: *value* – *value* is string for rate (2.5g, 5g, 10g, 20g, etc), omits switch mgmt port 0

portstate: *value* – *value* is a string for state (init, armed, active)

mtu: *value* – *value* is MTU size (256, 512, 1024, 2048 or 4096), omits switch mgmt port 0

labelpat: *value* – *value* is glob pattern for cable label

lengthpat: *value* – *value* is glob pattern for cable length

cabledetpat: *value* – *value* is glob pattern for cable details

linkdetpat: *value* – *value* is glob pattern for link details

portdetpat: *value* – *value* is glob pattern for port details

sm – Master subnet manager

smdetpat: *value* – *value* is glob pattern for SM details

route: *point1:point2* – all ports along the routes between the 2 given points

4.5.3.5 Examples

iba_report can generate hundreds of different reports. Following is a list of some commonly generated reports:

Analyze a fabric for bad cables:

```
iba_report -o slowlinks -o errors
```

Analyze a fabric for bad cables or misconfigured ports:

```
iba_report -o slowconfiglinks -o errors
```

Analyze a fabric for bad cables or misconfigured ports or misconnected ports:

```
iba_report -o slowconnlinks -o errors
```

Reverse lookup a CA lid:

```
iba_report -o brnodes -F lid:5
```

Reverse lookup a Switch lid:

```
iba_report -o brnodes -F lid:7:node
```

Reverse lookup a nodeguid:

```
iba_report -o brnodes -F nodeguid: 0x00066a0098000380
```

Reverse lookup a portguid:



```
iba_report -o brnodes -F portguid: 0x00066a00a0000380
```

Find all the connections to a server:

```
iba_report -o links -F node:duster
```

Find all the connections to a switch chip:

```
iba_report -o links -F 'node:i9k156'
```

Find all the connections to a multi-node system:

```
iba_report -o links -F systemguid:0x00066a0098000380
```

Report on all the components in a multi-node system:

```
iba_report -o comp -F node:goblin
```

Identify the routes between 2 servers:

```
iba_report -o route -S node:duster -D node:goblin
```

Identify the route between a server and a specific lid:

```
iba_report -o route -S node:duster -D lid:5
```

Identify the route between a server and the master SM:

```
iba_report -o route -S node:duster -D sm
```

Analyze the route between 2 nodes for bad cables or misconfigured ports or misconnected ports:

```
iba_report -o slowconnlunks -o errors -F route:node:cuda:node:duster
```

Identify the routes between this server and another server:

```
iba_report -o route -D node:goblin
```

Analyze a single switch and its neighbors for any high error counts:

```
iba_report -o errors -F 'node:i9k156'
```

Identify the routes between a server and an IOC:

```
iba_report -o route -S node:duster -D 'ioc:Chassis 0x00066A005000010C, Slot 2, IOC 2'
```

Clear all the port counters in the fabric:

```
iba_report -C -o none
```

Clear all the port counters on a multi-HCA server and its neighbor switch port(s):

```
iba_report -C -F node:goblin -o none
```

Clear all the port counters on a multi-HCA server but not its neighbor switch port(s):

```
iba_report -C -L -F node:goblin -o none
```

Check all port counters, clear them, then recheck:

```
iba_report -o errors -C; sleep 10; iba_report -o errors
```

Clear all port counters, wait 10 seconds, then check



```
Iba_report -i 10 -o errors
```

Check all port counters on a server and its neighbor switch port(s):

```
iba_report -o errors -F node:goblin
```

Check all port counters on a specific port on a server and its neighbor switch port:

```
iba_report -o errors -F node:goblin:port:2
```

Check all port counters on a specific port on a server but not its neighbor switch port:

```
iba_report -o errors -L -F node:goblin:port:2
```

Get all the detailed information for a server including port counters:

```
iba_report -o nodes -F node:goblin -d 5 -s
```

Get all the detailed information for an IOU including port counters:

```
iba_report -o nodes -F 'ioc:Chassis 0x00066A005000010C, Slot 2, IOC 2' -d 5 -s
```

4.5.3.6 Basics of Using iba_report

`iba_report` can be run with no options at all. In this mode it provides a brief list of the nodes in the fabric (the `brnodes` report). The report organizes nodes as CAs, Switches and Routers. It also includes a summary of all the SMs in the fabric.

Here is a sample of `iba_report` for a small fabric:

```
[root@duster root]# iba_report

Node Type Brief Summary

14 Connected CAs in Fabric:

NodeGUID          Type Name
-----
Port LID  PortGUID          Width Speed
0x0002c9020020e0d4 CA coyote1
      1 0x000d 0x0002c9020020e0d5  4x   2.5Gb
0x00066a00580001e0 CA VEx in Chassis 0x00066a005000010c, Slot 2
      2 0x0014 0x00066a02580001e0  4x   2.5Gb
0x00066a0098000001 CA julio
      1 0x000c 0x00066a00a0000001  4x   2.5Gb
0x00066a00980001b8 CA orc
      1 0x000b 0x00066a00a00001b8  4x   2.5Gb
0x00066a0098000380 CA goblin
      1 0x000a 0x00066a00a0000380  4x   2.5Gb
0x00066a0098000384 CA cuda
      1 0x0005 0x00066a00a0000384  1x   2.5Gb
      2 0x0006 0x00066a01a0000384  4x   2.5Gb
```



```

0x00066a00980003a6 CA erik
    1 0x0015 0x00066a00a00003a6 4x 2.5Gb
    2 0x0016 0x00066a01a00003a6 4x 2.5Gb
0x00066a00980006a2 CA goblin
    1 0x000f 0x00066a00a00006a2 4x 2.5Gb
0x00066a0098000849 CA rockaway
    2 0x000e 0x00066a01a0000849 4x 2.5Gb
0x00066a0098002813 CA brady
    1 0x0002 0x00066a00a0002813 4x 2.5Gb
    2 0x0003 0x00066a01a0002813 4x 2.5Gb
0x00066a0098002854 CA brady
    1 0x0004 0x00066a00a0002854 4x 2.5Gb
    2 0x0008 0x00066a01a0002854 4x 2.5Gb
0x00066a0098003f81 CA ibm345
    1 0x0007 0x00066a00a0003f81 4x 2.5Gb
0x00066a009800447b CA duster
    1 0x0011 0x00066a00a000447b 4x 2.5Gb
    2 0x0012 0x00066a01a000447b 4x 2.5Gb
0x00066a0098004a73 CA erik
    1 0x0009 0x00066a00a0004a73 4x 2.5Gb

```

3 Connected Switches in Fabric:

NodeGUID	Type	Name
0x00066a00280002cd	SW	InfiniCon Systems InfiniFabric (Sw A Dev A)
0 0x0013	0x00066a00280002cd	Noop Noop
3		4x 2.5Gb
5		4x 2.5Gb
0x00066a00d8000123	SW	InfiniCon Systems InfinIO9024
0 0x0001	0x00066a00d8000123	4x 2.5Gb
1		4x 2.5Gb
2		1x 2.5Gb
3		4x 2.5Gb
4		4x 2.5Gb



```
5          4x  2.5Gb
6          4x  2.5Gb
7          4x  2.5Gb
8          4x  2.5Gb
9          4x  2.5Gb
10         4x  2.5Gb
11         4x  2.5Gb
12         4x  2.5Gb
14         4x  2.5Gb
15         4x  2.5Gb
16         4x  2.5Gb
17         4x  2.5Gb
18         4x  2.5Gb
19         4x  2.5Gb
20         4x  2.5Gb

0x00066a10280002cd SW InfiniCon Systems InfiniFabric (Sw A Dev B)

  0 0x0010 0x00066a10280002cd Noop    Noop
  2          4x  2.5Gb
  4          4x  2.5Gb
```

1 Connected SMs in Fabric:

State	GUID	Name
Master	0x00066a00d8000123	InfiniCon Systems InfinIO9024

Each `iba_report` allows for various levels of detail. Increasing detail is shown as further indentation of the additional information. The `-d` option to `iba_report` controls the detail level. The default is 2. Values from 0-n are permitted. The maximum detail per report varies, but most have less than 5 detail levels.

For example, the above report when run at detail level 0 outputs:

```
[root@duster root]# iba_report -d 0
```

Node Type Brief Summary

14 Connected CAs in Fabric:

3 Connected Switches in Fabric:

1 Connected SMs in Fabric:

This is a summary of fabric components and is very similar to `fabric_info`.

At the next level of detail the report has more detail:



```
[root@duster root]# iba_report -d 1

Node Type Brief Summary

14 Connected CAs in Fabric:

NodeGUID          Type Name
0x0002c9020020e0d4 CA coyote1
0x00066a00580001e0 CA VEx in Chassis 0x00066a005000010c, Slot 2
0x00066a0098000001 CA julio
0x00066a00980001b8 CA orc
0x00066a0098000380 CA goblin
0x00066a0098000384 CA cuda
0x00066a00980003a6 CA erik
0x00066a00980006a2 CA goblin
0x00066a0098000849 CA rockaway
0x00066a0098002813 CA brady
0x00066a0098002854 CA brady
0x00066a0098003f81 CA ibm345
0x00066a009800447b CA duster
0x00066a0098004a73 CA erik

3 Connected Switches in Fabric:

NodeGUID          Type Name
0x00066a00280002cd SW InfiniCon Systems InfiniFabric (Sw A Dev A)
0x00066a00d8000123 SW InfiniCon Systems InfinIO9024
0x00066a10280002cd SW InfiniCon Systems InfiniFabric (Sw A Dev B)

1 Connected SMS in Fabric:

State      GUID          Name
Master     0x00066a00d8000123 InfiniCon Systems InfinIO9024
```

The above examples were all performed with a single report, the brnodes (Brief Nodes) report. However this is just one of the many topology reports which `iba_report` can generate. The others include:

- `nodes` – a more verbose form of brnode which can provide much greater levels of detail to drill down into all the details of every node, even down to all the port state, IOUs/IOCs/Services, Port counters.
- `comps` and `brcomps` are very similar to brnodes and nodes, except the reports are organized around systems. The grouping into systems is based on system image



guids for each node. This report will help to present more complex systems (such as servers with multiple HCAs or large switches composed of multiple switch chips).

Note:

All Intel® Switches implement a system image GUID and will therefore be properly grouped. However, some third-party devices do not implement the system image GUID and may report a value of 0. In such a case `iba_report` will treat each component as an independent system.

- `links` – This report presents all the links in the fabric. The output is very concise and helps to identify the connectivity between nodes in the fabric. This includes both internal (inside a large switch or system) and external ports (cables).
- `extlinks` – All of the external links in the fabric (for example, those between different systems). This omits links internal to a single system. Identification of a system is through SystemImageGuid.
- `lids` – This report is somewhat similar to `brnodes`, however it is organized and sorted by LID. The output is very concise and helps to provide a simple cross reference of LIDs assigned to each HCA and Switch in the fabric. This information can be useful in interpreting the output from the `linear`, `mcast` and `portusage` reports.
- `iious` – This is somewhat similar to the nodes reports, however the focus is around IOUs/IOCs and IO Services in the fabric. This report can be used to identify various IO devices in the fabric and their capabilities (such as the IBTA compliant direct-attach storage).
- `otherports` – All the ports which are not connected to this fabric. This report will identify additional ports on CAs or Switches which are not connected to this fabric. For switches these represent unused ports. For CAs these may be ports connected to other fabrics or unused ports.

The above reports are all summaries of the present state of the fabric. These reports can be very helpful to analyze the configuration of the fabric and or verify it was installed consistent with the desired design and configuration.

However, `iba_report` does not stop there. Additionally, `iba_report` has reports that will help to analyze the operational characteristics of the fabric and help to identify bottlenecks and faulty components in the fabric.

To assist in this area, `iba_report` also supports the following reports:

- `slowlinks` – Identifies links which are running slower than expected. This helps to pinpoint bad cables or components in the fabric, such as a 4x cable that is poorly-connected and therefore only runs at 1x link width. The analysis includes both link speed and width.
- `slowconfiglinks` – This extends on the `slowlinks` report to also report links which have been configured (most likely by software) to run at a width or speed below their potential. Such as DDR capable links which have been forced to run at SDR rates.
- `slowconnlinks` – This further extends on the `slowconfiglinks` report to also report links which are cabled such that one of the ends of the link will never run to its potential. Such as a DDR capable HCA connected to an SDR switch.
- `misconfiglinks` – This is similar to `slowconfiglinks` in that it reports links which have been configured to run below their potential. However it does not include links which are running slower than expected.
- `misconnlinks` – This is similar to `slowconnlinks` in that it reports links which have been connected between ports of different speed potential. However it does not include links which are running slower than expected, nor links which have been configured to run slower than their potential.



- **errors** – This performs a single point in time analysis of the PMA port counters for every node and port in the fabric. All the counters are compared against configured thresholds (defaults are those in the `iba_mon.conf` file). Any link whose counters exceed these thresholds are listed (and depending on the detail level the exact counter and threshold will be reported). This is a powerful way to identify marginal links in the fabric such as bad or loose cables or damaged components. The `iba_mon.si.conf` file can also be used to check for any non-zero values for signal integrity (SI) counters.
- **route** – This permits the user to identify two end points in the fabric (by node name, node GUID, port name, port GUID, system image GUID, LID, port GUID, IOC GUID or IOC name) and obtain a list of all the links and components used when these two end points communicate. If there are multiple paths between the end points (such as a CA with 2 connected ports or a system with 2 CAs), the route for every available path (based on presently configured routing tables) will be reported.
- **linear** – This shows the linear forwarding table for each switch in the fabric. This may be used to manually review the routing of unicast traffic in the fabric. For each switch every unicast LID is shown along with the port it will be routed out (egress port) and the neighboring Node and Port. For large fabrics this report can be quite large.
- **mcast** – This shows the multicast forwarding table for each switch in the fabric. This may be used to manually review the routing of multicast traffic in the fabric. For each switch every multicast LID is shown along with the list of ports it will be routed out. For large fabrics this report can be quite large.
- **portusage** – This provides a summary analysis of the unicast routing in the fabric in terms of how many LIDs of each node type are routed out a given port. This can be useful when doing analysis of how balanced the routes in the fabric are, especially for ISLs and core switches. For each switch, all the ports are shown along with the counts of how many unicast LIDs are routed out each port. The Total is shown along with: HCA-All, HCA-Base, Switch and Router. HCA-All includes all LIDs which correspond to an HCA (regardless of whether the LID is the base LID of the HCA or whether it maps to the HCA through LMC masking). HCA-Base includes LIDs which correspond to the base LID (only) of an HCA. HCA-Base will always be a subset of (less than or equal to) HCA-All. Switch includes all LIDs which correspond to a Switch. Router includes all LIDs which correspond to a Router. Only Ports with a non-zero Total are shown.
- **pathusage** – This reports computes all the CA to CA dLID paths through the fabric and reports on the usage of each ISL Port (SW to SW link). The `-F` option indicates which switches to analyze and which ports on those switches to analyze. Switch Port 0 is always omitted from the analysis. These reports can also be run against snapshots which were performed with the `-r` option.
- **treepathusage** – This report is similar to **pathusage** with the exception that **treepathusage** is applicable only to Fat Tree topologies and provides specific analysis of uplink and downlink paths, indicating what tier each switch is in within the fabric.

The above set of reports can therefore be very powerful ways to obtain point in time status and problem analysis for the fabric.

4.5.3.7 Simple Topology Verification

`iba_report` provides a flexible way to identify changes to the fabric or the appropriate reassembly of the fabric after a move (for example, after staging and testing the fabric in a remote location before final installation at a customer site).



In this mode of operation, all the above reports are available, however the types of information output can be filtered. For example, using the `-P` option, information which would not persist across a fabric reboot (such as LIDs and error counters) will be omitted from the report (and marked out with `xxx`). Such a report can be saved for later comparison to a future report. Since `iba_report` produces simple text reports, standard tools such as `sdiff` (for example, side by side diff) can be used for easy comparison and analysis of what changed.

Given the wealth of reports available, the user can select the information they want to save. For ease of use an `all` report is available which includes all the reports of general interest.

If software configuration changes are anticipated (such as adjusting the time-outs the SM configures in the fabric), the `iba_report -H` option can be used. This will further limit the report to only include hardware information. This is a superset of `-P` and omits more information.

A related but independent option is `-N`. This will omit all the node and IOC names from the report. If changes are anticipated in this area, this option can be used so future diffs will not report changes in names.

4.5.3.8 Advanced Topology Verification

`iba_report` also provides a powerful way to compare the state of the fabric against a previous state or a user generated configuration for the fabric. This is accomplished by using the `-T` option to `iba_report` to supply an XML description of the fabric configuration.

The XML format used by the `-T` option is the same as the XML format generated by the `-o links` or `-o extlinks` and/or `-o brnodes` reports when they are run with the `-x` option.

A simple way to perform topology verification against a previous configuration is to generate the previous topology using a command such as:

```
iba_report -o links -o brnodes -x > topology.xml
```

Then at a point in the future the fabric can be compared against that topology using a command such as:

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

Unlike simple diff comparisons discussed in Simple Topology Verification, this form of topology verification will perform a more context sensitive comparison and can present information in terms of links, nodes and/or SMs which are missing, unexpected or incorrectly configured.

All the other capabilities of `iba_report` are fully available when using a `topology_input` file. For example, `snapshot_input` files can also be used to generate or compare topologies based on previous fabric snapshots. In addition the `-F` option may be used to focus the analysis. Note: `verify*` reports may still report missing links, nodes or SMs outside the scope of the desired focus.

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

`verifylinks` and `verifyextlinks` both perform the same analysis, however they differ in the scope of the analysis. `verifylinks` will check all links in the fabric. In contrast `verifyextlinks` will limit its verification to links outside of a system. In `verifyextlinks` links which are between nodes with the same SystemImageGuid,



such as within a large Intel® Switching Chassis, will not be analyzed. Also `verifyextlinks` will ignore links from the `topology_input` file which specify a non-zero value for the XML tag `<Internal>` within the `<Link>` tag.

The XML format of topology input can appear as follows (the example below is purposely brief and omits many links, nodes, and SMs):

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

<Report>

<LinkSummary>

<Link>

<Rate>20g</Rate>

<MTU>2048</MTU>

<Internal>0</Internal>

<LinkDetails>Bender to Switch</LinkDetails>

<Cable>

<CableLength>11m</CableLength>

<CableLabel>S4567</CableLabel>

<CableDetails>gore cable model 456</CableDetails>

</Cable>

<Port>

<NodeGUID>0x0002c9020020e004</NodeGUID>

<PortGUID>0x0002c9020020e005</PortGUID>

<PortNum>1</PortNum>

<NodeType>CA</NodeType>

<NodeDesc>bender HCA-1</NodeDesc>

<PortDetails>bender primary port</PortDetails>

</Port>

<Port>

<NodeGUID>0x00066a0007000df6</NodeGUID>

<PortNum>1</PortNum>

<NodeType>SW</NodeType>

<NodeDesc>i9k159 Leaf 4, Chip A</NodeDesc>

</Port>

</Link>

<Link>

<Rate>20g</Rate>
```



```
<MTU>2048</MTU>

<Internal>0</Internal>

<Port>

<NodeGUID>0x0002c9020025a678</NodeGUID>
<PortGUID>0x0002c9020025a679</PortGUID>
<PortNum>1</PortNum>
<NodeType>CA</NodeType>
<NodeDesc>mindy2 HCA-1</NodeDesc>
</Port>

<Port>

<NodeGUID>0x00066a0007000e6d</NodeGUID>
<PortNum>4</PortNum>
<NodeType>SW</NodeType>
<NodeDesc>i9k159 Leaf 5, Chip A</NodeDesc>
</Port>
</Link>
</LinkSummary>
<Nodes>
<CAs>
<Node id="0x0002c9020025a678">
<NodeGUID>0x0002c9020025a678</NodeGUID>
<NodeDesc>mindy2 HCA-1</NodeDesc>
<NodeDetails>mindy2 only HCA</NodeDetails>
</Node>
</CAs>
<Switches>
<Node id="0x00066a000600025a">
<NodeGUID>0x00066a000600025a</NodeGUID>
<NodeDesc>i9k159 Spine 1, Chip A</NodeDesc>
<NodeDetails>core switch</NodeDetails>
</Node>
</Switches>
<SMs>
<SM id="0x0002c9020025a678:1">
```



```

<NodeGUID>0x0002c9020025a678</NodeGUID>

<NodeDesc>mindy2 HCA-1</NodeDesc>

<PortNum>1</PortNum>

<PortGUID>0x0002c9020025a679</PortGUID>

<NodeType>CA</NodeType>

<NodeType_Int>1</NodeType_Int>

<SMDetails>mindy2 SM</SMDetails>

</SM>

</SMs>

</Nodes>

</Report>

```

The meaning of the various XML tags are as follows:

<Report> – Primary top level tag. Exactly one such tag is permitted per file. Alternatively this may be **<Topology>**

<LinkSummary> – Container tag describing all the links expected in the fabric. Alternatively **<ExternalLinkSummary>** may be used. **<ExternalLinkSummary>** should be used if the file only describes external links. If both external and internal links are described, **<LinkSummary>** should be used. Only one of these two choices is permitted per file

<Link> – Container tag describing a single link. Many instances of this tag can occur per **<LinkSummary>** or **<ExternalLinkSummary>**

The following tags are permitted per **<Link>**:

<Rate> – String describing the expected rate of the link. Valid values are 2.5g, 5g, 10g, 20g, 30g, 40g, 60g, 80g, or 120g. The value is case insensitive but must contain no extra whitespace. This describes the expected rate of the link. Alternatively an integer value **<Rate_Int>** may be provided based on the IBTA defined values for Rate from the SMA packets. If both **<Rate>** and **<Rate_Int>** are specified, which ever appears later within the given Link will be used. If neither is specified, the rate of the link will not be verified

<MTU> – An integer describing the expected MTU of the link. Valid values are 256, 512, 1024, 2048 and 4096. If not specified, the MTU of the link will not be verified

<Internal> – A flag indicating if the link is internal or external. A value of 0 indicates external links which will be processed by both **verifylinks** and **verifyextlinks**. A value of 1 indicates an internal link which will only be processed by **verifylinks**. If omitted the actual fabric link attributes or the attributes of the port are used to determine if the link should be processed. The value for this field is not verified against the actual fabric.

<LinkDetails> – A free form text field of up to 64 characters. This field is optional and may be supplied by the user to augment the reports. When provided this will be output as a link attribute in all reports which show link details, such as links, extlinks, route, **verifylinks** and **verifyextlinks** reports. It is recommended to use this field to describe the purpose of the link. This field can also be used by the **linkdetpat** focus option to select the link

<Cable> – A container tag providing additional information about the cable



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

The following are permitted per <Cable>:

<CableLength> – A free form text field up to 10 characters. This field is optional and may be supplied by the user to augment the reports. When provided this will be output as a link cable attribute in all reports which show link details, such as links, extlinks, route, verifylinks and verifyextlinks reports. It is recommended to use this field to describe the length of the cable using text such as 11m. This field can also be used by the lengthpat focus option to select the link

<CableLabel> – A free form text field up to 20 characters. This field is optional and may be supplied by the user to augment the reports. When provided this will be output as a link cable attribute in all reports which show link details, such as links, extlinks, route, verifylinks and verifyextlinks reports. It is recommended to use this field to describe the identifying label attached to the cable using text such as S4576. This field can also be used by the labelpat focus option to select the link. Use of this field to match the actual unique physical labels placed on the cables during installation can greatly help cross referencing the reports to the physical cluster, such as when needing to identify or replace cables.

<CableDetails> – A free form text field of up to 64 characters. This field is optional and may be supplied by the user to augment the reports. When provided this will be output as a link attribute in all reports which show link details, such as links, extlinks, route, verifylinks and verifyextlinks reports. It is recommended to use this field to describe the type, model and/or manufacturer of the cable. This field can also be used by the cabledetpat focus option to select the link

The following are permitted per <Port>:

<NodeGUID> – The Node GUID reported by the SMA for the given CA, Switch or Router

<PortGUID> – The Port GUID reported by the SMA for the given CA, switch or router. Note that switches only have PortGuids for port 0 (the internal management port) while CAs and routers have a unique GUID for every port.

<PortNum> – The Port Number within the CA, switch or router

<NodeDesc> – The Node Description reported by the CA, Switch or router. Where possible its recommended the user configure a unique value for this field in each node in your fabric. For example Intel® True Scale OFED+ Linux hosts will use the combination of Linux hostname and HCA number to create a unique NodeDesc.

<NodeType> – The node type reported by the node. This can be: CA, SW or RT. Alternatively an integer value <NodeType_Int> may be provided based on the IBTA defined values for NodeType from the SMA packets. If both <NodeType> and <NodeType_Int> are specified, which ever appears later within the given Port will be used. If neither is specified, the node type of the port will not be verified

<PortDetails> – A free form text field of up to 64 characters. This field is optional and may be supplied by the user to augment the reports. When provided this will be output as a port attribute in all reports which show port details, such as links, extlinks, route, comps, verifylinks and verifyextlinks reports. It is recommended to use this field to describe the purpose of the port. This field can also be used by the portdetpat focus option to select the port



The above fields are used to associate a Port in the `topology_input` file with an actual port in the fabric (referred to below as resolving the port). As such its not necessary to provide all of the above information. Association to an actual port in the fabric is performed using the following order of checks based on which tags are specified:

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

If NodeDesc is used to specify ports, its important that the fabric be configured such that NodeDesc is unique. Otherwise the `<Port>` may resolve to a different port than desired which could result in incorrect results or errors during topology verification.

When redundant information is provided, the extra information is ignored while resolving the port. However during `verifylinks` or `verifyextlinks` all the input provided is verified against the actual fabric and any discrepancies will be reported.

Some examples of redundant information:

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

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

`<Nodes>` – Container tag describing all the nodes expected in the fabric.

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

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

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

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

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

The following tags are permitted per `<Node>`:

`<NodeGUID>` – The Node GUID reported by the SMA for the given CA, Switch or Router

`<NodeDesc>` – The Node Description reported by the CA, Switch or router. Where possible its recommended the user configure a unique value for this field in each node in your fabric. For example Intel® True Scale OFED+ Linux hosts will use the combination of Linux hostname and HCA number to create a unique NodeDesc.



`<NodeDetails>` – A free form text field of up to 64 characters. This field is optional and may be supplied by the user to augment the reports. When provided this will be output as a node attribute in all reports which show node details, such as links, extlinks, route, comps, verifycas, verifysws, verifyrts, verifylinks and verifyextlinks reports. It is recommended to use this field to describe the purpose and/or model of the node. This field can also be used by the `nodedetpat` focus option to select the node

The above fields are used to associate a Node (CA, Switch or Router) in the `topology_input` file with an actual node in the fabric (referred to below as resolving the node). As such its not necessary to provide all of the above information. Association to an actual node in the fabric is performed using the following order of checks based on which tags are specified:

- NodeGUID
- NodeDesc

If NodeDesc is used to specify nodes, its important that the fabric be configured such that NodeDesc is unique. Otherwise the `<Node>` may resolve to a different node than desired which could result in incorrect results or errors during topology verification.

When redundant information is provided, the extra information is ignored while resolving the node. However during verifycas, verifysws or verifyrts all the input provided is verified against the actual fabric and any discrepancies will be reported.

Some examples of redundant information:

NodeGuid, NodeDesc - NodeDesc is not used to resolve node

The node type (as implied by the container tag for the `<Node>`) is never used during resolution, it is only used during verification.

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

The following tags are permitted per `<SM>`:

`<NodeGUID>` – The Node GUID reported by the SMA for the given CA, Switch or Router which is running the SM

`<NodeDesc>` – The Node Description reported by the CA, Switch or router which is running the SM. Where possible its recommended the user configure a unique value for this field in each node in your fabric. For example Intel® True Scale OFED+ Linux hosts will use the combination of Linux hostname and HCA number to create a unique NodeDesc.

`<PortGUID>` – The Port GUID reported by the SMA for the given CA, switch or router which is running the SM. Note that switches only have PortGuids for port 0 (the internal management port) while CAs and routers have a unique GUID for every port.

`<PortNum>` – The Port Number within the CA, switch or router which is running the SM

`<NodeType>` – The node type reported by the node which is running the SM. This can be: CA, SW or RT. Alternatively an integer value `<NodeType_Int>` may be provided based on the IBTA defined values for NodeType from the SMA packets. If both `<NodeType>` and `<NodeType_Int>` are specified, which ever appears later within the given Port will be used. If neither is specified, the node type of the SM will not be verified



`<SMDetails>` – A free form text field of up to 64 characters. This field is optional and may be supplied by the user to augment the reports. When provided this will be output as a SM attribute in all reports which show SM details, such as comps and verifysms reports. It is recommended to use this field to describe the purpose of the SM. This field can also be used by the `smdetpat` focus option to select the SM

The above fields are used to associate a Port running an SM in the `topology_input` file with an actual port in the fabric (referred to below as resolving the SM). As such its not necessary to provide all of the above information. Association to an actual port in the fabric is performed using the following order of checks based on which tags are specified:

- NodeGUID, PortNum
- NodeGUID, PortGUID
- NodeGUID – If given CA has exactly 1 active port or is a switch
- NodeDesc, PortNum
- NodeDesc, PortGUID
- NodeDesc – If given CA has exactly 1 active port or is a switch
- PortGUID, PortNum – limited usefulness
- PortGUID

If NodeDesc is used to specify SM ports, its important that the fabric be configured such that NodeDesc is unique. Otherwise the `<SM>` may resolve to a different port than desired which could result in incorrect results or errors during topology verification.

When redundant information is provided, the extra information is ignored while resolving the port for an SM. However during verifysms all the input provided is verified against the actual fabric and any discrepancies will be reported.

Some examples of redundant information:

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

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

4.5.3.9 Augmented Report Information

As discussed in Advanced Topology Verification, a `topology_input` file may include additional information about the cable (length, label, details), links (details), ports (details), nodes (details) and SMs (details). As such, a `topology_input` file can be used during any report to provide information about the fabric which is not electronically available. This can be very useful to help cross reference the output of the report against the physical fabric. For example if the cable length field is supplied, reports can be focused on all cables of a given length. Similarly if cable labels are supplied, the output from reports will include the labels, making it much easier to locate the actual cables for rerouting, replacement, etc.

4.5.3.10 Focused Reports

One of the more powerful features of `iba_report` is the ability to focus a report on a subset of the fabric. Using the `-F` option the user can specify a node name, node name pattern, node guid, node type, port guid, IOC name, IOC name pattern, ioc guid, ioc type, system image guid, port gid, port rate, mtu, lid or SM. The subsequent report will indicate the total components in the fabric but will only report on those which relate to



the focus area. For example in a nodes report, if a port is specified for focus, only the node containing that port will be reported on. In a links report, only the link using that port will be reported.

When a focus is used for fabric analysis, `-o errors`, `-C` or `-i`, the analysis will include all the ports selected by the focus as well as their neighbors. If desired the `-L` option may be used to limit the operation to exactly the selected ports.

Notice that a focus level that is different from the orientation of the report may be chosen. For example if a node name is specified as the focus for the links report, a report of all the links to that node will be provided. This could include multiple switch ports or CA ports.

By carefully using this feature of report focus, reverse lookups can be done. For example, doing a `brnodes` report with a focus on a LID will reverse lookup the LID and indicate what node it is for.

When focusing a report, it can sometimes be helpful to also use a detail level of 0 or 1. In this case the report will show only a count of number of matches (for detail 0) and just the highest level of the entity which matches (for detail 1).

4.5.3.11 Advanced Focus

The node name, node name pattern, node guid, node type, IOC name, IOC name pattern, IOC GUID, IOC type and system image GUID also allow for a port number specifier. This permits the focus to be limited to the given port number. If the selection resolves to multiple switches or CAs (for example, a system composed of multiple nodes), all ports on the present fabric matching the given port number will be selected.

An even more advanced form of focus is to focus on the route between any two points. This will focus on all the ports involved in that route and can be an excellent way to focus in quickly on a performance or error situation which is being reported between two specific points in the fabric (Such as a `StatusTimeoutRetry` that MPI may be reporting between two processes in its run).

Focus can use glob style patterns. This permits a wild carded focus by node name or IOC name. If a naming convention is used for fabric components, this can provide a powerful way to focus reports on nodes. For example, if the host names are prefixed with an indication of their purpose, searches can be performed based on the purpose of the node. For example if the following naming convention is used: `l###` = login node `###`, `n###` = compute node `###`, `s###` = storage node `###`, etc. Node purposes can be focused by using patterns such as `'l*'`, `'n*'` or `'s*'`

Note:

A glob style pattern is a shell style wildcard pattern as used by `bash` and many other tools. When using such patterns they should be single quoted so that the shell will not try to expand them to match local file names.

Typically a focused report will include a summary at its start of the items focused on. When the focus has a large scope, this list can be quite long. In this case the `-Q` option can be used to omit this section from the report.

4.5.3.12 Focus Examples

Below are some examples of using the focus options:

```
iba_report -o nodes -F portguid:0x00066a00a000447b
```

```
iba_report -o nodes -F nodeguid:0x00066a009800447b:port:1
```

```
iba_report -o nodes -F nodeguid:0x00066a009800447b
```



```

iba_report -o nodes -F node:duster
iba_report -o nodes -F node:duster:port:1
iba_report -o nodes -F 'nodepat:d*'
iba_report -o nodes -F 'nodepat:d*:port:1'
iba_report -o nodes -F nodetype:CA
iba_report -o nodes -F nodetype:CA:port:1
iba_report -o nodes -F lid:1
iba_report -o nodes -F lid:1:node
iba_report -o nodes -F gid:0xfe80000000000000:0x00066a00a000447b
iba_report -o nodes -F systemguid:0x00066a009800447b
iba_report -o nodes -F systemguid:0x00066a009800447b:port:1
iba_report -o nodes -F iocguid:0x00066a01300001e0
iba_report -o nodes -F iocguid:0x00066a01300001e0:port:2
iba_report -o nodes -F 'ioc:Chassis 0x00066A005000010C, Slot 2, IOC 1'
iba_report -o nodes -F 'ioc:Chassis 0x00066A005000010C, Slot 2, IOC 1:port:2'
iba_report -o nodes -F 'iocpat:*Slot 2*'
iba_report -o nodes -F 'iocpat:*Slot 2*:port:2'
iba_report -o nodes -F ioctype:XXXX
iba_report -o nodes -F ioctype:XXXX:port:2
iba_report -o nodes -F sm
iba_report -o nodes -F route:node:duster:node:cuda
iba_report -o nodes -F route:node:duster:port:1:node:cuda:port:2

```

4.5.3.13 Scriptable output

`iba_report` permits custom scripting. As previously mentioned, options like `-H`, `-P` and `-N` can aid the generation of reports that can be compared to each other.

The `-x` option permits output reports to be generated in XML format. The XML hierarchy is similar to the textual reports. Use of XML permits other XML tools (such as PERL XML extensions) to easily parse `iba_report` output such that scripts can be created to further search and refine report output formats.

The `xml_extract` FastFabric tool can easily convert between XML files and delimited text files. See the [“Converting iba_report output to excel importable files - xml_extract” on page 213](#) for more information.

This allows `iba_report` to be integrated into custom scripts. It can also be used to generate customer-specific new report formats, cross reference `iba_report` with other site-specific information, etc.



4.5.3.14 Using iba_report to monitor for fabric changes

`iba_report` can easily be used in other scripts. For example the following simple script could be run as a cron job to identify if the fabric has changed as compared to the initial design:

```
#!/bin/bash

# specify some filenames to use
expected_config=/usr/local/report.master # master copy of config previously created
config=/tmp/report$$ # where we will generate new report
diffs=/tmp/report.diff$$ # where we will generate diffs

iba_report -o all -d 5 -P > $config 2>/dev/null
if ! diff $config $expected_config > $diffs 2>/dev/null
then
# notify admin, for example mail the new report to the admin
    cat $diffs $expected_config $config |
mail -s "fabric change detected" admin@somewhere
fi

rm -f $config $diffs
```

4.5.3.15 Sample Output

4.5.3.15.1 Analysis of all ports in fabric for errors, inconsistent connections, bad cables

```
[root@duster root]# iba_report -o errors -o slowconlinks
```

Links running slower than faster port Summary

Links running slower than expected:

20 of 20 Links Checked, 0 Errors found

Links configured to run slower than supported:

Rate	MTU	NodeGUID	Port	Type	Name
		Enabled	Supported		
2.5g	2048	0x00066a0098000384	1	CA	cuda
		1x 2.5Gb	1-4x		2.5Gb
<->		0x00066a00d8000123	2	SW	InfiniCon Systems InfinIO9024
		1-4x 2.5Gb	1-4x		2.5Gb

20 of 20 Links Checked, 1 Errors found



Links connected with mismatched speed potential:

20 of 20 Links Checked, 0 Errors found

Links with errors > threshold Summary

Configured Error Thresholds:

SymbolErrorCounter	100
LinkErrorRecoveryCounter	3
LinkDownedCounter	3
PortRcvErrors	100
PortRcvRemotePhysicalErrors	100
PortXmitDiscards	100
PortXmitConstraintErrors	10
PortRcvConstraintErrors	10
LocalLinkIntegrityErrors	3
ExcessiveBufferOverrunErrors	3
VL15Dropped	100

Rate	MTU	NodeGUID	Port	Type	Name
10g	2048	0x00066a0098000001	1	CA	julio
<->		0x00066a00d8000123	8	SW	InfiniCon Systems InfinIO9024

LinkDownedCounter: 5 Exceeds Threshold: 3

10g	2048	0x00066a00980001b8	1	CA	orc
<->		0x00066a00d8000123	10	SW	InfiniCon Systems InfinIO9024

LinkDownedCounter: 5 Exceeds Threshold: 3

10g	2048	0x00066a0098000380	1	CA	goblin
<->		0x00066a00d8000123	15	SW	InfiniCon Systems InfinIO9024

SymbolErrorCounter: 65535 Exceeds Threshold: 100

LinkErrorRecoveryCounter: 255 Exceeds Threshold: 3

PortRcvErrors: 65535 Exceeds Threshold: 100

SymbolErrorCounter: 41079 Exceeds Threshold: 100

LinkErrorRecoveryCounter: 188 Exceeds Threshold: 3



```
10g 2048 0x00066a0098003f81 1 CA ibm345
<->      0x00066a00d8000123 12 SW InfiniCon Systems InfinIO9024
SymbolErrorCounter: 9533 Exceeds Threshold: 100
LinkErrorRecoveryCounter: 46 Exceeds Threshold: 3
PortRcvErrors: 617 Exceeds Threshold: 100
20 of 20 Links Checked, 4 Errors found
```

4.5.3.15.2 Identification of the route between 2 nodes in the fabric

```
[root@duster root]# ./iba_report -o route -S node:orc -D node:julio
Routes Summary Between:
Node: 0x00066a00980001b8 CA orc
and Node: 0x00066a0098000001 CA julio
```

Routes between ports:

```
0x00066a00980001b8 1 CA orc
and 0x00066a0098000001 1 CA julio
1 Paths
SGID: 0xfe80000000000000:00066a00a00001b8
DGID: 0xfe80000000000000:00066a00a0000001
SLID: 0x000b DLID: 0x000c Reversible: Y PKey: 0xffff
Raw: N FlowLabel: 0x00000 HopLimit: 0x00 TClass: 0x00
SL: 0 Mtu: 2048 Rate: 10g PktLifeTime: 67 ms Pref: 0
```

	Rate	MTU	NodeGUID	Port	Type	Name
	10g	2048	0x00066a00980001b8	1	CA	orc
->			0x00066a00d8000123	10	SW	InfiniCon Systems InfinIO9024
	10g	2048	0x00066a00d8000123	8	SW	InfiniCon Systems InfinIO9024
->			0x00066a0098000001	1	CA	julio

2 Links Traversed

4.5.3.15.3 Analysis of the route between 2 nodes for errors, inconsistent connections, etc

```
[root@duster root]# ./iba_report -o route -S node:orc -D node:julio
Routes Summary Between:
Node: 0x00066a00980001b8 CA orc
and Node: 0x00066a0098000001 CA julio
```




```

Routes between ports:

    0x00066a00980001b8    1 CA orc
and  0x00066a0098000001    1 CA julio

1 Paths

    SGID: 0xfe80000000000000:00066a00a00001b8
    DGID: 0xfe80000000000000:00066a00a0000001
    SLID: 0x000b DLID: 0x000c Reversible: Y PKey: 0xffff
    Raw: N FlowLabel: 0x00000 HopLimit: 0x00 TClass: 0x00
    SL:  0 Mtu: 2048 Rate:  10g PktLifeTime:  67 ms Pref: 0

    Rate MTU  NodeGUID          Port Type Name
    10g 2048 0x00066a00980001b8    1 CA orc
    ->          0x00066a00d8000123  10 SW InfiniCon Systems InfinIO9024
    10g 2048 0x00066a00d8000123    8 SW InfiniCon Systems InfinIO9024
    ->          0x00066a0098000001    1 CA julio

2 Links Traversed

```

4.5.3.15.4 Obtain very detailed information about nodes

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

```

[root@duster root]# iba_report -o nodes -F node:erik -d 5 -s

Node Type Summary Focused on:

    System: 0x00066a0098004a73

    Node: 0x00066a00980003a6 CA erik
    Node: 0x00066a0098004a73 CA erik

13 Connected CAs in Fabric:

    Name: erik

    NodeGUID: 0x00066a00980003a6 Type: CA
    Ports: 2 PartitionCap: 64 SystemImageGuid: 0x00066a0098004a73
    BaseVer: 1 SmaVer: 1 VendorID: 0x66a DeviceID: 0x5a44 Rev: 0xa1

2 Connected Ports:

    PortNum:  1 LID: 0x0015 GUID: 0x00066a00a00003a6
    Neighbor:  0x00066a00d8000123    9 SW InfiniCon Systems InfinIO9024
    PortState: Active                PhysState: LinkUp    DownDefault: Pollg

```



```
LID:      0x0015          LMC: 0          Subnet: 0xfe800000000000
SMLID:    0x0001    SMLS: 0    RespTimeout: 33 ms    SubnetTimeout: 6 ms
M_KEY:    0x0000000000000000    Lease:      0 s          Protect: Readonly
MTU:      Active:      2048    Supported:      2048    VL Stall: 0
LinkWidth: Active:      4x    Supported:      1-4x    Enabled:      1-4x
LinkSpeed: Active:      2.5Gb    Supported:      2.5Gb    Enabled:      2.5Gb
VLs:      Active:      4+1    Supported:      4+1    HOQLife: 4096 ns
Capability 0x02010048: CR CM SL Trap
Violations: M_Key:      0    P_Key:      0    Q_Key:      0
ErrorLimits: Overrun: 15    LocalPhys: 15    DiagCode: 0x0000
P_Key Enforcement: In: Off Out: Off    FilterRaw: In: Off Out: Off

Performance: Transmit

Xmit Data      3452 MiB (905003712 Quads)
Xmit Pkts      12569496

Performance: Receive

Rcv Data      6569 MiB (1722249442 Quads)
Rcv Pkts      23920772

Errors:

Symbol Errors      0
Link Error Recovery      0
Link Downed      0
Port Rcv Errors      0
Port Rcv Rmt Phys Err      0
Port Rcv Sw Relay Err      0
Port Xmit Discards      0
Port Xmit Constraint      0
Port Rcv Constraint      0
Local Link Integrity      0
Exc. Buffer Overrun      0
VL15 Dropped      0

PortNum: 2 LID: 0x0016 GUID: 0x00066a01a00003a6
Neighbor: 0x00066a00d8000123 7 SW InfiniCon Systems InfinIO9024
PortState: Active      PhysState: LinkUp    DownDefault: Pollg
LID:      0x0016          LMC: 0          Subnet: 0xfe800000000000
```



```

SMLID: 0x0001   SMLS: 0   RespTimeout: 33 ms   SubnetTimeout:6 ms
M_KEY: 0x0000000000000000 Lease: 0 s   Protect: Readonly
MTU:      Active: 2048 Supported: 2048 VL Stall: 0
LinkWidth: Active: 4x Supported: 1-4x Enabled: 1-4x
LinkSpeed: Active: 2.5Gb Supported: 2.5Gb Enabled: 2.5Gb
VLs:      Active: 4+1 Supported: 4+1 HOQLife: 4096 ns
Capability 0x02010048: CR CM SL Trap
Violations: M_Key: 0 P_Key: 0 Q_Key: 0
ErrorLimits: Overrun: 15 LocalPhys: 15 DiagCode: 0x0000
P_Key Enforcement: In: Off Out: Off FilterRaw: In: Off Out: Off

Performance: Transmit

Xmit Data 0 MiB (0 Quads)
Xmit Pkts 0

Performance: Receive

Rcv Data 0 MiB (0 Quads)
Rcv Pkts 0

Errors:

Symbol Errors 0
Link Error Recovery 0
Link Downed 0
Port Rcv Errors 0
Port Rcv Rmt Phys Err 0
Port Rcv Sw Relay Err 0
Port Xmit Discards 0
Port Xmit Constraint 0
Port Rcv Constraint 0
Local Link Integrity 0
Exc. Buffer Overrun 0
VL15 Dropped 0

```

Name: erik

NodeGUID: 0x00066a0098004a73 Type: CA

Ports: 2 PartitionCap: 64 SystemImageGuid: 0x00066a0098004a73

BaseVer: 1 SmaVer: 1 VendorID: 0x66a DeviceID: 0x5a44 Rev: 0xa1

1 Connected Ports:



```
PortNum: 1 LID: 0x0009 GUID: 0x00066a00a0004a73

Neighbor: 0x00066a00d8000123 18 SW InfiniCon Systems InfinIO9024

PortState: Active PhysState: LinkUp DownDefault: Pollg

LID: 0x0009 LMC: 0 Subnet: 0xfe80000000000000

SMLID: 0x0001 SMSL: 0 RespTimeout: 33 ms SubnetTimeout: 6 ms

M_KEY: 0x0000000000000000 Lease: 0 s Protect: Readonly

MTU: Active: 2048 Supported: 2048 VL Stall: 0

LinkWidth: Active: 4x Supported: 1-4x Enabled: 1-4x

LinkSpeed: Active: 2.5Gb Supported: 2.5Gb Enabled: 2.5Gb

VLs: Active: 4+1 Supported: 4+1 HOQLife: 4096 ns

Capability 0x02010048: CR CM SL Trap

Violations: M_Key: 0 P_Key: 0 Q_Key: 0

ErrorLimits: Overrun: 15 LocalPhys: 15 DiagCode: 0x0000

P_Key Enforcement: In: Off Out: Off FilterRaw: In: Off Out: Off

Performance: Transmit

Xmit Data 0 MiB (238320 Quads)

Xmit Pkts 3310

Performance: Receive

Rcv Data 0 MiB (238320 Quads)

Rcv Pkts 3310

Errors:

Symbol Errors 0

Link Error Recovery 0

Link Downed 0

Port Rcv Errors 0

Port Rcv Rmt Phys Err 0

Port Rcv Sw Relay Err 0

Port Xmit Discards 0

Port Xmit Constraint 0

Port Rcv Constraint 0

Local Link Integrity 0

Exc. Buffer Overrun 0

VL15 Dropped 0
```

2 Matching CAs Found



3 Connected Switches in Fabric:

0 Matching Switches Found

1 Connected SMS in Fabric:

0 Matching SMS Found

4.5.3.15.5 Obtain very detailed information about IOUs

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

```
[root@duster root]# iba_report -o ious -F ioc:'Chassis 0x00066a005000010c, Slot 2,
IOC 2' -d 5
```

IOU Summary Focused on:

```
Ioc: 2 0x00066a02300001e0 Chassis 0x00066a005000010c, Slot 2, IOC 2
in Node: 0x00066a00580001e0 CA VEx in Chassis 0x00066a005000010c, Slot
```

1 IOUs in Fabric:

```
Name: VEx in Chassis 0x00066a005000010c, Slot 2
```

```
NodeGUID: 0x00066a00580001e0 Type: CA
```

```
Ports: 2 PartitionCap: 64 SystemImageGuid: 0x00066a00580001e0
```

```
BaseVer: 1 SmaVer: 1 VendorID: 0x66a DeviceID: 0x5a44 Rev: 0xa1
```

1 Connected Ports:

```
PortNum: 2 LID: 0x0013 GUID: 0x00066a02580001e0
```

```
Neighbor: 0x00066a00280002cd 3 SW InfiniCon Systems InfiniFabric
```

(Sw A Dev A)

```
PortState: Active PhysState: LinkUp DownDefault: Pollig
LID: 0x0013 LMC: 0 Subnet: 0xfe80000000000000
SMLID: 0x0001 SML: 0 RespTimeout: 33 ms SubnetTimeout: 56 ms
M_KEY: 0x0000000000000000 Lease: 0 s Protect: Readonly
MTU: Active: 2048 Supported: 2048 VL Stall: 0
LinkWidth: Active: 4x Supported: 1-4x Enabled: 1-4x
LinkSpeed: Active: 2.5Gb Supported: 2.5Gb Enabled: 2.5Gb
VLs: Active: 1+1 Supported: 4+1 HOQLife: 4096 ns
Capability 0x02090048: CR DM CM SL Trap
Violations: M_Key: 0 P_Key: 0 Q_Key: 0
ErrorLimits: Overrun: 15 LocalPhys: 15 DiagCode: 0x0000
```



```
P_Key Enforcement: In: Off Out: Off FilterRaw: In: Off Out: Off
Max IOCs:      3 Change ID:      9 DiagDeviceId: 0 Rom: 0
IocSlot:      2 GUID: 0x00066a02300001e0
ID String: Chassis 0x00066A005000010C, Slot 2, IOC 2
IO Class: 2000 SubClass: 66a Protocol: 0 Protocol Ver: 1
VendorID: 0x66a DeviceID: 0x30 Rev: 0x1
Subsystem: VendorID: 0x66a DeviceID: 0x30
Capability: 0x33: ST SF WT WF
Send Depth: 2 Size: 256; RDMA Read Depth: 0 RDMA Size: 4294967295
2 Services:
Name: InfiniNIC.InfiniConSys.Control:02
Id: 0x1000066a00000002
Name: InfiniNIC.InfiniConSys.Data:02
Id: 0x1000066a00000102
1 Matching IOUs Found
```

4.5.3.15.6 Identify connections and links composing the fabric

```
[root@duster root]# iba_report -o links
```

Link Summary

20 Links in Fabric:

Rate	MTU	NodeGUID	Port	Type	Name
10g	2048	0x00066a00280002cd	3	SW	InfiniCon Systems InfiniFabric (Sw A Dev A)
<->		0x00066a00580001e0	2	CA	VEx in Chassis 0x00066a005000010c, Slot 2
10g	2048	0x00066a00280002cd	5	SW	InfiniCon Systems InfiniFabric (Sw A Dev A)
<->		0x00066a10280002cd	4	SW	InfiniCon Systems InfiniFabric (Sw A Dev B)
10g	2048	0x00066a0098000001	1	CA	julio
<->		0x00066a00d8000123	8	SW	InfiniCon Systems InfinIO9024
10g	2048	0x00066a00980001b8	1	CA	orc
<->		0x00066a00d8000123	10	SW	InfiniCon Systems InfinIO9024
10g	2048	0x00066a0098000380	1	CA	goblin
<->		0x00066a00d8000123	15	SW	InfiniCon Systems InfinIO9024
2.5g	2048	0x00066a0098000384	1	CA	cuda
<->		0x00066a00d8000123	2	SW	InfiniCon Systems InfinIO9024



```

10g 2048 0x00066a0098000384 2 CA cuda
<->      0x00066a00d8000123 1 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a00980003a6 1 CA erik
<->      0x00066a00d8000123 9 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a00980003a6 2 CA erik
<->      0x00066a00d8000123 7 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a00980006a2 1 CA goblin
<->      0x00066a00d8000123 20 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a0098000849 2 CA rockaway
<->      0x00066a00d8000123 3 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a0098002813 1 CA brady
<->      0x00066a00d8000123 19 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a0098002813 2 CA brady
<->      0x00066a00d8000123 5 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a0098002854 1 CA brady
<->      0x00066a00d8000123 11 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a0098002854 2 CA brady
<->      0x00066a00d8000123 6 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a0098003f81 1 CA ibm345
<->      0x00066a00d8000123 12 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a009800447b 1 CA duster
<->      0x00066a00d8000123 4 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a009800447b 2 CA duster
<->      0x00066a00d8000123 16 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a0098004a73 1 CA erik
<->      0x00066a00d8000123 18 SW InfiniCon Systems InfinIO9024
10g 2048 0x00066a00d8000123 14 SW InfiniCon Systems InfinIO9024
<->      0x00066a10280002cd 2 SW InfiniCon Systems InfiniFabric (Sw A Dev B)

```

4.5.3.15.7 Reverse lookups, translate a LID or GUID into the information about the node or port represented

```
[root@duster root]# iba_report -o nodes -F lid:5
```

Node Type Summary Focused on:

```

Port:      1 0x00066a00a0000384
          in Node: 0x00066a0098000384 CA cuda

```



13 Connected CAs in Fabric:

Name: cuda

NodeGUID: 0x00066a0098000384 Type: CA

Ports: 2 PartitionCap: 64 SystemImageGuid: 0x00066a0098000384

BaseVer: 1 SmaVer: 1 VendorID: 0x66a DeviceID: 0x5a44 Rev: 0xa1

2 Connected Ports:

PortNum: 1 LID: 0x0005 GUID: 0x00066a00a0000384

Neighbor: 0x00066a00d8000123 2 SW InfiniCon Systems InfinIO9024

Width: 1x Speed: 2.5Gb

1 Matching CAs Found

3 Connected Switches in Fabric:

0 Matching Switches Found

1 Connected SMs in Fabric:

0 Matching SMs Found

4.5.3.15.8 Forward lookups - lookup nodes or IOCs by name

```
[root@duster root]# iba_report -o nodes -F node:erik
```

Node Type Summary Focused on:

System: 0x00066a0098004a73

Node: 0x00066a00980003a6 CA erik

Node: 0x00066a0098004a73 CA erik

13 Connected CAs in Fabric:

Name: erik

NodeGUID: 0x00066a00980003a6 Type: CA

Ports: 2 PartitionCap: 64 SystemImageGuid: 0x00066a0098004a73

BaseVer: 1 SmaVer: 1 VendorID: 0x66a DeviceID: 0x5a44 Rev: 0xa1

2 Connected Ports:

PortNum: 1 LID: 0x0015 GUID: 0x00066a00a00003a6

Neighbor: 0x00066a00d8000123 9 SW InfiniCon Systems InfinIO9024

Width: 4x Speed: 2.5Gb



```

PortNum: 2 LID: 0x0016 GUID: 0x00066a01a00003a6
Neighbor: 0x00066a00d8000123 7 SW InfiniCon Systems InfinIO9024
Width: 4x Speed: 2.5Gb

Name: erik

NodeGUID: 0x00066a0098004a73 Type: CA

Ports: 2 PartitionCap: 64 SystemImageGuid: 0x00066a0098004a73

BaseVer: 1 SmaVer: 1 VendorID: 0x66a DeviceID: 0x5a44 Rev: 0xa1

1 Connected Ports:

PortNum: 1 LID: 0x0009 GUID: 0x00066a00a0004a73
Neighbor: 0x00066a00d8000123 18 SW InfiniCon Systems InfinIO9024
Width: 4x Speed: 2.5Gb

2 Matching CAs Found

3 Connected Switches in Fabric:

0 Matching Switches Found

1 Connected SMS in Fabric:

0 Matching SMS Found

```

4.5.3.15.9 Generate reports in a “comparable manner” so topology verification can be performed against a known good configuration

Note: To shorten the length of the output, the following example focuses on only 1 node.

```

[root@duster root]# iba_report -o nodes -F node:erik -d 5 -P

Node Type Summary Focused on:

System: 0x00066a0098004a73

Node: 0x00066a00980003a6 CA erik

Node: 0x00066a0098004a73 CA erik

13 Connected CAs in Fabric:

Name: erik

NodeGUID: 0x00066a00980003a6 Type: CA

Ports: 2 PartitionCap: 64 SystemImageGuid: 0x00066a0098004a73

BaseVer: 1 SmaVer: 1 VendorID: 0x66a DeviceID: 0x5a44 Rev: 0xa1

2 Connected Ports:

PortNum: 1 LID: xxxxxx GUID: 0x00066a00a00003a6

```



```
Neighbor: 0x00066a00d8000123 9 SW InfiniCon Systems InfinIO9024
PortState: Active PhysState: LinkUp DownDefault: Pollig
LID: xxxxxx LMC: 0 Subnet: 0xfe80000000000000
SMLID: xxxxxx SMSL: 0 RespTimeout: 33 ms SubnetTimeout: 56 ms
M_KEY: 0x0000000000000000 Lease: 0 s Protect: Readonly
MTU: Active: 2048 Supported: 2048 VL Stall: 0
LinkWidth: Active: 4x Supported: 1-4x Enabled: 1-4x
LinkSpeed: Active: 2.5Gb Supported: 2.5Gb Enabled: 2.5Gb
VLs: Active: 4+1 Supported: 4+1 HOQLife: 4096 ns
Capability 0x02010048: CR CM SL Trap
Violations: M_Key: xxxxx P_Key: xxxxx Q_Key: xxxxx
ErrorLimits: Overrun: 15 LocalPhys: 15 DiagCode: 0x0000
P_Key Enforcement: In: Off Out: Off FilterRaw: In: Off Out: Off
PortNum: 2 LID: xxxxxx GUID: 0x00066a01a00003a6
Neighbor: 0x00066a00d8000123 7 SW InfiniCon Systems InfinIO9024
PortState: Active PhysState: LinkUp DownDefault: Pollig
LID: xxxxxx LMC: 0 Subnet: 0xfe80000000000000
SMLID: xxxxxx SMSL: 0 RespTimeout: 33 ms SubnetTimeout: 56 ms
M_KEY: 0x0000000000000000 Lease: 0 s Protect: Readonly
MTU: Active: 2048 Supported: 2048 VL Stall: 0
LinkWidth: Active: 4x Supported: 1-4x Enabled: 1-4x
LinkSpeed: Active: 2.5Gb Supported: 2.5Gb Enabled: 2.5Gb
VLs: Active: 4+1 Supported: 4+1 HOQLife: 4096 ns
Capability 0x02010048: CR CM SL Trap
Violations: M_Key: xxxxx P_Key: xxxxx Q_Key: xxxxx
ErrorLimits: Overrun: 15 LocalPhys: 15 DiagCode: 0x0000
P_Key Enforcement: In: Off Out: Off FilterRaw: In: Off Out: Off
Name: erik
NodeGUID: 0x00066a0098004a73 Type: CA
Ports: 2 PartitionCap: 64 SystemImageGuid: 0x00066a0098004a73
BaseVer: 1 SmaVer: 1 VendorID: 0x66a DeviceID: 0x5a44 Rev: 0xa1
1 Connected Ports:
PortNum: 1 LID: xxxxxx GUID: 0x00066a00a0004a73
Neighbor: 0x00066a00d8000123 18 SW InfiniCon Systems InfinIO9024
```



```

PortState: Active          PhysState: LinkUp    DownDefault: Polli
g
LID:      xxxxxx          LMC: 0          Subnet: 0xfe8000000000000
SMLID:    xxxxxx    SMLS: 0    RespTimeout: 33 ms    SubnetTimeout: 56 ms
M_KEY:    0x0000000000000000    Lease: 0 s          Protect: Readonly
MTU:      Active: 2048    Supported: 2048    VL Stall: 0
LinkWidth: Active: 4x    Supported: 1-4x    Enabled: 1-4x
LinkSpeed: Active: 2.5Gb    Supported: 2.5Gb    Enabled: 2.5Gb
VLs:      Active: 4+1    Supported: 4+1    HOQLife: 4096 ns
Capability 0x02010048: CR CM SL Trap
Violations: M_Key: xxxxx P_Key: xxxxx Q_Key: xxxxx
ErrorLimits: Overrun: 15 LocalPhys: 15 DiagCode: 0x0000
P_Key Enforcement: In: Off Out: Off FilterRaw: In: Off Out: Off
2 Matching CAs Found

3 Connected Switches in Fabric:
0 Matching Switches Found

1 Connected SMs in Fabric:
0 Matching SMs Found

```

4.5.3.16 Snapshots

The ability to snapshot the state of the fabric for later offline analysis is also provided by `iba_report`. This is accomplished using the `-o snapshot` report. This report generates an XML snapshot of the present fabric status in a format that `iba_report` can parse. It is recommended that users not develop their own tools against this format as it may change in future versions of `iba_report`.

When a snapshot is being generated, various `iba_report` options are ignored (such as `-F`, `-P`, `-H` and `-N`). However, it is valid to use options such as `-s` (to include port counters in the snapshot), `-r` (to include switch routing tables in the snapshot), `-V` (to include QOS VL-related tables in the snapshot) and `-i`, `-L`, `-a` and `-C` to control the port counters. When a snapshot is being generated, no other reports may be performed during the given run (for example, no additional `-o` options).

Once a snapshot has been generated, it may then be used as input to generate the majority of `iba_report` reports. This is accomplished using the `-X snapshot input` option, where the file is the output from a previous `-o snapshot` run. When using a snapshot as input, the fabric will not be accessed and the node running `iba_report` need not be attached to the fabric. As such, selected options are not available (such as `-i`, `-a`, `-C`, `-h hca`, `-p port`). Similarly the reports generated from the snapshot will only include port counters if the original snapshot was run with the `-s` option (if not, reports such as `-o errors` are not permitted against the snapshot). Similarly only if the original snapshot was run with the `-r` option, will reports such as `-o linear`, `-o`



mcast and -o portusage, -o pathusage, -o treepathusage, -o route be permitted. If it is desired to use standard input (stdin) for the snapshot file, -X - may be specified. This can be helpful if snapshots are piped through gzip/gunzip to conserve disk space.

Note: Limitations of -o route:

- The Path Records reported may not be complete. Fields such as SLID, SL, PKey, MTU, Rate, PktLifeTime are not available, so iba_report simply shows the minimum valid value or an invalid value. These values do not impact the actual route shown.
- Some routes reported may not be incomplete or not available to applications (for example, when disrupted Torus fabrics are being analyzed).

The snapshot capability can be used to provide powerful analysis capabilities. Not only can multiple reports be run against the exact same fabric snapshot (also saving time by not requiring the subsequent reports to query the fabric), but also historic snapshots can be retained for later offline analysis or historical tracking of the fabric.

4.5.4 iba_reports

(All) iba_reports is a front end to iba_report. It provides many of the same options and capabilities of iba_report. However in addition it can run a report against multiple fabrics/subnets (for example, local host HCA ports). It also can automatically use a topology_input file such that the reports will be augmented using additional details from the topology_input file.

4.5.4.1 Usage

```
iba_reports [-t portsfile] [-p ports] [iba_report arguments]
```

or

```
iba_reports --help
```

4.5.4.2 Options

--help - produce full help text

-t *portsfile* - file with list of local HCA ports used to access fabric(s) for analysis. The default is /etc/sysconfig/iba/ports.

-p *ports* - list of local HCA ports used to access fabric(s) for analysis. The default is the first active port.

This is specified as **HCA:port**:

0:0 - 1st active port in system

0:y - Port y within system

x:0 - 1st active port on HCA x

x:y - HCA x, port y

iba_report arguments - any of the other iba_report arguments. The -h and -X options are not available. Note that the meaning of -p is different for iba_reports than iba_report. Also when run against multiple fabrics, the -x and -o snapshot options are also not available.

Note: When run against multiple fabrics the -F option will be applied to all fabrics. See ["iba_report" on page 175](#) for more information.



`-T topology_input` – As for `iba_report`, this option permits a topology input file to be specified. However it also permits the filename to have `%P` as a marker which will be replaced with the `hca_port` being operated on (such as `0:0` or `1:2`). The default is `/etc/sysconfig/iba/topology.%P.xml`. If `-T NONE` is specified, no topology input file will be used.

4.5.4.3 Example

```
iba_reports

iba_reports -p '1:1 1:2 2:1 2:2'
```

4.5.4.4 Environment Variables

The following environment variables are also used by this command:

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

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

`FF_TOPOLOGY_FILE` – File containing `topology_input` (may have `%P` marker in filename), used in absence of `-T`.

For simple fabrics, the Intel® FastFabric Toolset host would be connected to a single fabric. By default the first active port on the Intel® FastFabric Toolset host will be used to analyze the fabric.

However, in more complex fabrics, the Intel® FastFabric Toolset host may be connected to more than one fabric (or subnet). In this case the specific ports and/or HCAs to use for fabric analysis may be specified.

Specification of the ports to be used can be performed on the command line using the `-p` option, in a file specified using the `-t` option, through the environment variables `PORTS` or `PORTS_FILE`, or using the `ports_file` configuration option in `fastfabric.conf`. If the specified file does not exist or is empty, the first active port on the local system will be used. In more complex configurations (such as where the Intel® FastFabric Toolset host is connected to multiple True Scale Fabrics or subnets), the user will need to specify the exact ports to use such that all fabrics are analyzed. For more information, refer to [“Selection of local Ports \(subnets\)” on page 28](#).

Specification of the `topology_input` file to be used can be performed on the command line using the `-T` option, in a file specified through the environment variable `FF_TOPOLOGY_FILE`, or using the `ff_topology_file` configuration option in `fastfabric.conf`. If the specified file does not exist, no `topology_input` file will be used. Alternately the filename can be specified as `NONE` to prevent use of a `topology_input` file. For more information, refer to [“iba_report” on page 175](#).

4.5.5 Converting iba_report output to excel importable files - xml_extract

(Linux) `xml_extract` takes well-formed XML as input, extracts element values as specified by command line options, and outputs the data as lines (records) of data in delimited format (commonly referred to as comma-separated-values (CSV) format). `xml_extract` is intended to be used with `iba_report`, to parse and filter its XML output, and to allow the filtered output to be imported into other tools such as excel spread sheets and customer written scripts. `xml_extract` can also be used with any well-formed XML stream to extract element values into a delimited format.



4.5.5.1 Usage

```
xml_extract [-v] [-H] [-d delimiter] [-e extract_element]  
            [-s suppress_element] [-X input_file]  
            [-P param_file]
```

4.5.5.2 Options

-e/--extract *extract_element* - Name of the XML element to extract. Elements can be used multiple times; elements can be nested in any order, but are output in the order specified; an optional attribute (or attribute and value) can also be specified with elements:

```
-e element  
-e element:attrName  
-e element:attrName:attrValue
```

Elements can be specified multiple times, with a different attribute name or attribute value.

-s/--suppress *suppress_element* - Name of the XML element to suppress; can be used multiple times (in any order); supports the same syntax as -e.

-d/--delimit *delimiter* - Use delimiter (single character or string) as the delimiter between element names and element values; default is semicolon;

-X/--infile *input_file* - Input XML from input_file instead of stdin;

-P/--pfile *param_file* - Input command line options (parameters) from param_file;

-H/--noheader - Do not output element name header record;

-v/--verbose - Verbose output: 1) output progress reports during extraction; and 2) output prepended wildcard characters on element names in output header record.

4.5.5.3 Details

xml_extract is a flexible and powerful tool to process an XML stream; it:

- Requires no specific element names to be present in the XML;
- Assumes no hierarchical relationship between elements;
- Allows extracted element values to be output in any order;
- Allows an element's value to be extracted only in the context (scope) of another (specified) element;
- Allows extraction to be suppressed during the scope of specified elements.

xml_extract takes the XML input stream from either stdin or a specified input file. xml_extract does not use nor require a connection to an True Scale Fabric.

xml_extract works from two lists of elements supplied as command line or input parameters. The first is a list of elements whose values are to be extracted ("extraction elements"). The second is a list of elements for which extraction is to be suppressed ("suppression elements"). When an extraction element is encountered (and extraction is not suppressed), the value of the element is extracted for later output in an "extraction record". An extraction record contains a value for all extraction elements (including those which have a null value).



When a suppression element is encountered, then no extraction will be performed during the extent of that element (start through end). Suppression is maintained for elements specified inside the suppression element, including elements which may happen to match extraction elements. Suppression can be used to prevent extraction in sections of XML which are present, but not of current interest (for example, NodeDesc or NodeGUID inside a Neighbor specification of `iba_report`).

During operation, `xml_extract` outputs an extraction record under the following conditions:

- One or more extraction elements containing a non-null value go out of scope (the element containing the extraction elements is ended) and a record containing the element values has not already been output;
- A new and different value is specified for an extraction element and an extraction record containing the previous value has not already been output.

Element names (extraction or suppression) can be made context sensitive with an enclosing element name using the syntax `element1.element2`. In which case, `element2` will be extracted (or extraction will be suppressed) only when `element2` is enclosed by `element1`. The syntax also allows `'*'` to be specified as a wildcard. `'*.element3'` specifies `element3` enclosed by any element or sequence of elements (ex. `element1.element3` or `element1.element2.element3`). `'element1.*.element3'` specifies `element3` enclosed by `element1` with any number of (but at least 1) intermediate elements. `xml_extract` prepends any entered element name not containing a `'*'` (anywhere) with `'*.'`, matching the element regardless of the enclosing elements.

Note: Any element names which include a wildcard should be quoted to the shell attempting to wildcard match against filenames.

At the beginning of operation `xml_extract`, by default, outputs a delimited "header record" containing the names of the extraction elements. The order of the names is the same as specified on the command line and is the same order as that of the extraction record. Output of the header record can be disabled with the `-H` option. By default, element names are shown as they were entered on the command line. The `-v` option causes element names to be output as they are used during extraction, with any prepended wildcard characters.

Options (parameters) to `xml_extract` can be specified on the command line, with a parameter file, or using both methods. A parameter file is specified with `-P param_file`. When a parameter file specification is encountered on the command line, option processing on the command line is suspended, the parameter file is read and processed entirely, and then command line processing is resumed. Option syntax within a parameter file is the same as on the command line. Multiple parameter file specifications can be made, on the command line or within other parameter files. At each point that a parameter file is specified, current option processing is suspended while the parameter file is processed, then resumed. Options are processed in the order they are encountered on the command line or in parameter files. A parameter file can be up to 8192 bytes in size and may contain up to 512 parameters.

4.5.5.4 Sample Use and Output

The following shows a simple example of `iba_report` output filtered by `xml_extract`.

```
>iba_report -o comps -s -x | xml_extract -d \; -e NodeDesc -e SystemImageGUID -e
NumPorts -s Neighbor

NodeDesc;SystemImageGUID;NumPorts

mindy2 HCA-1;0x0002c9020025a67b;2
```



```
MT25408 ConnectX Mellanox Technologies;0x0002c9030000079b;2
cuda;0x00066a009800413e;2
duster;0x00066a009800447b;2
stewie HCA-1;0x00066a0098007b70;2
InfiniCon System InfinIO 9024 Lite;0x00066a00d900045f;24
InfiniCon System InfinIO 9024 Lite;0x00066a00d9000479;24
InfiniCon System InfinIO 9024 Lite;0x00066a00d90004e9;24
i9k159 Spine 1, Chip A;0x00066a00da000159;24
i9k159 Spine 2, Chip A;0x00066a00da000159;24
i9k159 Leaf 2, Chip A;0x00066a00da000159;24
i9k159 Leaf 3, Chip A;0x00066a00da000159;24
i9k159 Leaf 1, Chip A;0x00066a00da000159;24
i9k159 Leaf 4, Chip A;0x00066a00da000159;24
i9k159 Leaf 6, Chip A;0x00066a00da000159;24
i9k159 Leaf 5, Chip A;0x00066a00da000159;24
i9k159 Leaf 8, Chip A;0x00066a00da000159;24
i9k159 Leaf 7, Chip A;0x00066a00da000159;24
i9k159 Spine 1, Chip B;0x00066a00da000159;24
i9k159 Spine 2, Chip B;0x00066a00da000159;24

i9k159 Spine 1, Chip A;;
i9k159 Spine 2, Chip A;;
```

4.5.5.5 Sample Scripts

Five sample scripts are available as examples of how to use `xml_extract` and as prototypes for customized scripts. They combine various calls to `iba_report` with a call to `xml_extract` with commonly used parameters.

4.5.6 iba_extract_perf

`iba_extract_perf` provides a report of all the performance counters in a format easily imported to excel for further analysis.

It generates a detailed `iba_report` component summary report and pipes the result to `xml_extract`, extracting element values for `NodeDesc`, `SystemImageGUID`, `PortNum`, and all the performance counters. Extraction is performed only from the Systems portion of the report which does not contain Neighbor information (the Neighbor and SMs portions are suppressed).

The implementation of the script is as follows:

```
iba_report -o comps -s -x -d 10 | xml_extract -d \; -e NodeDesc -e SystemImageGUID
```




```
-e PortNum -e XmitDataMB -e XmitData -e XmitPkts -e RcvDataMB -e RcvData -e RcvPkts
-e SymbolErrors -e LinkErrorRecovery -e LinkDowned -e PortRcvErrors -e
PortRcvRemotePhysicalErrors -e PortRcvSwitchRelayErrors -e PortXmitDiscards -e
PortXmitConstraintErrors -e PortRcvConstraintErrors -e LocalLinkIntegrityErrors -e
ExcessiveBufferOverrunErrors -e VL15Dropped -s Neighbor -s SMs
```

4.5.7 iba_extract_error

`iba_extract_error` is very similar to `iba_extract_perf`, however it only reports error counters. Its output is easily imported into excel for further analysis of fabric errors.

It generates the same `iba_report` as `iba_extract_perf` but extracts error counters (a subset of the performance counters). Extraction from the Neighbor and SMs portions of the report is suppressed.

The implementation of the script is as follows:

```
iba_report -o comps -s -x -d 10 | xml_extract -d \; -e NodeDesc -e
SystemImageGUID -e PortNum -e SymbolErrors -e LinkErrorRecovery -e LinkDowned -e
PortRcvErrors -e PortRcvRemotePhysicalErrors -e PortRcvSwitchRelayErrors -e
PortXmitConstraintErrors -e PortRcvConstraintErrors -e LocalLinkIntegrityErrors -e
ExcessiveBufferOverrunErrors -s Neighbor -s SMs
```

4.5.8 iba_extract_stat

`iba_extract_stat` is a sample excel importable report of cable symbol errors. It performs an error analysis of a fabric and provides augmented information from a topology_input file. Therefore the report can provide cable information as well as symbol error counts.

`iba_extract_stat` generates a detailed `iba_report` errors report which also has a topology input file (see "[iba_report](#)" on page 175 for more information about topology files). The report is piped to `xml_extract` which extracts values for Link, Cable and Port (the port element names are context-sensitive). Note that `xml_extract` generates 2 extraction records for each link (one for each port on the link); therefore, `iba_extract_stat` merges the 2 records into a single record and removes redundant link and cable information. `iba_extract_stat` contains a 'while read' loop which reads the CSV line-by-line, uses 'cut' to remove redundant information, and outputs the data on a common line.

The portion of the script which calls `iba_report` and `xml_extract` follows:

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

4.5.9 iba_extract_stat2

`iba_extract_stat2` is similar to `iba_extract_stat` except that it also extracts all error counters in addition to SymbolErrors (error counter names are context-sensitive).

The portion of the script which calls `iba_report` and `xml_extract` follows:

```
iba_report -x -d 10 -s -o errors -T $@ | xml_extract -d \; -e Rate -e MTU -e
Internal -e LinkDetails -e CableLength -e CableLabel -e CableDetails -e
Port.NodeGUID -e Port.PortGUID -e Port.PortNum -e Port.PortType -e Port.NodeDesc -e
Port.PortDetails -e PortXmitData.Value -e PortXmitPkts.Value -e PortRcvData.Value
-e PortRcvPkts.Value -e SymbolErrors.Value -e LinkErrorRecovery.Value -e
LinkDowned.Value -e PortRcvErrors.Value -e PortRcvRemotePhysicalErrors.Value -e
PortRcvSwitchRelayErrors.Value -e PortXmitConstraintErrors.Value -e
PortRcvConstraintErrors.Value -e LocalLinkIntegrityErrors.Value -e
```



ExcessiveBufferOverrunErrors.Value

4.5.10 iba_extract_link

`iba_extract_link` produces an excel importable summary of the fabric topology.

`iba_extract_link` generates an `iba_report` links report and pipes the result to `xml_extract`, extracting element values for Link, Cable and Port (the port element names are context-sensitive). `iba_extract_link` uses the same logic as `iba_extract_stat` to merge the 2 link records into a single record and remove redundant information.

The portion of the script which calls `iba_report` and `xml_extract` follows:

```
iba_report -x -o links | xml_extract -d \; -e Rate -e MTU -e LinkDetails -e  
CableLength -e CableLabel -e CableDetails -e Port.NodeDesc -e Port.PortNum
```

Additional commands that also use `iba_report` and `xml_extract` include:

- `iba_extract_bad_links`
- `iba_extract_lids`
- `iba_extract_sel_links`

4.5.11 Remove All Specified XML Tags - xml_filter

`xml_filter` is the opposite of `xml_extract`. It processes an XML file and removes all the specified tags. The remaining tags are output and indentation can also be reformatted.

4.5.11.1 Usage

```
xml_filter [-t|-k] [-l] [-i indent] [-s element] [-P param_file] [input_file]
```

4.5.11.2 Options

`-t` – Trim leading and trailing whitespace in tag contents.

`-k` – In tags with purely whitespace which contain newlines, keep newlines as is (default is to format as an empty list).

`-l` – Add comments with line numbers after each end tag. This can make comparison of resulting files easier since original line numbers will be available.

`-i indent` – Set indentation to use per level (default 4).

`-s element` – Name of XML element to suppress can be used multiple times, order does not matter.

`-P param_file` – Read command parameters from `param_file`.

`input_file` – XML file to read. Default is stdin.

4.5.12 Re-indenting XML files - xml_indent

(Linux) `xml_indent` takes well-formed XML as input, filters out comments and generates a uniformly indented equivalent XML file. `xml_indent` can be used to reformat files for easier reading and review. It can also be used to reformat a file for easy comparison with diff.



4.5.12.1 Usage

```
xml_indent [-t|-k] [-i indent] [input_file]
```

4.5.12.2 Options

-t - Trim leading and trailing whitespace in tag contents.

-k - In tags with purely while space which contain new lines, keep newlines as is (default is to format as an empty list);

-i *indent* - Set indent to use per level (default is 4);

input_file - Input XML (default is stdin);

4.5.13 Creating iba_report topology_input files - xml_generate

(Linux) `xml_generate` takes delimited (commonly referred to as comma-separated-values (CSV)) data as input and, with user-specified element names, generates sequences of XML containing the element values within start and end tag specifications. The tool is appropriate for creating an XML representation of fabric data from its CSV form.

4.5.13.1 Usage

```
xml_generate [-v][-d delimiter] [-i number] [-g element][-h element] [-e element]
[-X input_file] [-P param_file]
```

4.5.13.2 Options

-g/--generate *element* - Generate value for element using value in next field from the input file; can be used multiple times on the command line

-h/--header *element* - Generate enclosing header start tag for element; can be used multiple times on the command line

-e/--end *element* - Generate enclosing header end tag for element; can be used multiple times on the command line

-d/--delimit *delimiter* - Specifies the delimiter character that separates values in the input file; default is semicolon

-i/--indent *number* - Specifies the number of spaces to indent each level of XML output; default is zero

-X/--infile *input_file* - File to read delimited input data from. One record per line with fields in each record separated by the specified delimiter

-P/--pfile *param_file* - Input command line options (parameters) from *param_file*

-v/--verbose - Verbose output: output progress reports during generation.

4.5.13.3 Details

`xml_generate` takes the CSV data from an input file. It generates fragments of XML, and in combination with a script can be used to generate complete XML sequences. `xml_generate` does not use nor require a connection to an True Scale Fabric.

`xml_generate` reads CSV element values and applies element (tag) names to those values. The element names are supplied as command line options to the tool and constitute a template which is applied to the input.

Element names on the command line are of three (3) types, distinguished by their command line option - "Generate", "Header" and "Header_End". The Header and Header_End types together constitute "enclosing" element types. Enclosing elements do not contain a value, but serve to separate and organize Generate elements.

Generate elements, along with a value from the CSV input file, cause XML in the form of `<element_name>value</element_name>` to be generated. Generate elements are normally the majority of the XML output since they specify elements containing the input values. Header elements cause an XML header start tag of the form:

`<element_name>` to be generated. Header_End elements cause an XML header end tag of the form `</element_name>` to be generated. Output of enclosing elements is controlled entirely by the placement of those element types on the command line. No check for matching start and end tags or proper nesting of tags is performed by `xml_generate`.

Options (parameters) to `xml_generate` can be specified on the command line, with a parameter file, or both. A parameter file is specified with `-P param_file`. When a parameter file specification is encountered on the command line, option processing on the command line is suspended, the parameter file is read and processed entirely, and then command line processing is resumed. Option syntax within a parameter file is the same as on the command line. Multiple parameter file specifications can be made, on the command line or within other parameter files. At each point that a parameter file is specified, current option processing is suspended while the parameter file is processed, then resumed. Options are processed in the order they are encountered on the command line or in parameter files. A parameter file can be up to 8192 bytes in size and may contain up to 512 parameters.

4.5.13.4 Using `xml_generate` to create topology input files

It is anticipated that each user or system integrator will have their own unique format for fabric design and topology information. `xml_generate` is provided as a tool which can be used to create scripts to translate from the user specific format into the `iba_report topology_input` file format. `xml_generate` itself works against a CSV style file with one line per record. Given such a file it can produce hierarchical XML output of arbitrary complexity and depth.

The typical flow for a script which translates from a user specific format into `iba_report topology_input` would be:

- As needed, reorganize the data into link and node data CSV files, in a sequencing similar to that used by `iba_report topology_input`. One link record per line in one temporary file, one node record per line in another temporary file and one SM per line in a third temporary file
- The script must directly output the boilerplate for XML version, etc
- `xml_generate` can be used to output the Link section of the `topology_input`, using the link record temporary file
- `xml_generate` can be used to output the Node sections of the `topology_input` using the node record temporary file. If desired there could be separate node record temporary files for CAs, Switches and Routers.
- `xml_generate` can be used to output the SM section of the `topology_input`, if desired.
- The script must directly output the closing XML tags to complete the `topology_input` file.



4.5.14 Sample Script and Output - iba_gen_topology

A sample script, `/usr/bin/iba_gen_topology`, is available as an example of how to use `xml_generate` and as a prototype for customized needs.

`iba_gen_topology` generates sample topology verification XML. It uses CSV input files `iba_topology_links.txt`, `iba_topology_CAs.txt` and `iba_topology.SWs.txt` to generate LinkSummary, Node CAs, and Node SWs information respectively. These files are samples of what might be produced as part of translating a user custom file format into temporary intermediate CSV files.

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

```
xml_generate -X /opt/iba/samples/iba_topology_1.txt -d \; -h Link -g Rate -g
Rate_Int -g MTU -g LinkDetails -h Cable -g CableLength -g CableLabel -g
CableDetails -e Cable -h Port -g NodeGUID -g PortNum -g NodeDesc -g PortGUID -g
NodeType -g NodeType_Int -g PortDetails -e Port -h Port -g NodeGUID -g PortNum -g
NodeDesc -g PortGUID -g NodeType -g NodeType_Int -g PortDetails -e Port -e Link

xml_generate -X /opt/iba/samples/iba_topology_2.txt -d \; -h Node -g NodeGUID -g
NodeDesc -g NodeDetails -g HostName -g NodeType -g NodeType_Int -g NumPorts -e Node
```

4.5.14.1 iba_topology_links.txt

This file can be found in `/opt/iba/samples/`. For brevity this sample shows only 2 links. The second link is an example of omitting some information, in the second line the MTU, LinkDetails and other fields are not present, this is indicated by an empty value for the field (for example, no information between the semicolon delimiters).

Note: The lines below exceed the available width of the page, consequently a blank line is shown between lines to make it clear where the line ends. In actual use no blank lines should be provided.

```
20g;6;2048;IO Server Link;11m;S4567;gore cable model
456;0x0002c9020020e004;1;bender HCA-1;0x0002c9020020e004;CA;1;Some info about
port;0x00066a0007000df6;7;SilverStorm 9080 GUID=0x00066a00da000159 Leaf 4, Chip
A;;SW;2;

20g;6;;;;;0x0002c9020025a678;1;mindy2
HCA-1;;CA;1;;0x00066a0007000e6d;4;SilverStorm 9080 GUID=0x00066a00da000159 Leaf 5,
Chip A;;SW;2;
```

4.5.14.2 iba_topology_CAs.txt

This file can be found in `/opt/iba/samples/`. For brevity this sample shows only two nodes.

```
0x0002c9020020e004;bender HCA-1;More details about node

0x0002c9020025a678;mindy2 HCA-1;Node details
```

4.5.14.3 iba_topology_SWs.txt

This file can be found in `/opt/iba/samples/`. For brevity this sample shows only two nodes.



```
0x00066a0007000df6;SilverStorm 9080 GUID=0x00066a00da000159 Leaf 4, Chip A;  
0x00066a0007000e6d;SilverStorm 9080 GUID=0x00066a00da000159 Leaf 5, Chip A;
```

4.5.14.4 iba_topology_SMs.txt

This file can be found in /opt/iba/samples/. For brevity this sample shows only one node.

```
0x0002c9020025a678;1;mindy2 HCA-1;0x00066a0007000e6d;CA;  
details about SM
```

4.5.14.5 Sample Output

When run against the supplied topology input files, iba_gen_topology produces:

```
<?xml version="1.0" encoding="utf-8" ?>  
  
<Topology>  
  
<LinkSummary>  
  
<Link>  
  
<Rate>20g</Rate>  
  
<MTU>2048</MTU>  
  
<Internal>0</Internal>  
  
<LinkDetails>IO Server Link</LinkDetails>  
  
<Cable>  
  
<CableLength>11m</CableLength>  
  
<CableLabel>S4567</CableLabel>  
  
<CableDetails>gore cable model 456</CableDetails>  
  
</Cable>  
  
<Port>  
  
<NodeGUID>0x0002c9020020e004</NodeGUID>  
  
<PortNum>1</PortNum>  
  
<NodeDesc>bender HCA-1</NodeDesc>  
  
<PortGUID>0x0002c9020020e004</PortGUID>  
  
<NodeType>CA</NodeType>  
  
<PortDetails>Some info about port</PortDetails>  
  
</Port>  
  
<Port>  
  
<NodeGUID>0x00066a0007000df6</NodeGUID>  
  
<PortNum>7</PortNum>  
  
<NodeDesc>SilverStorm 9080 GUID=0x00066a00da000159 Leaf 4, Chip A</NodeDesc>
```



```

<NodeType>SW</NodeType>

</Port>

</Link>

<Link>

<Rate>20g</Rate>

<Internal>0</Internal>

<Cable>

</Cable>

<Port>

<NodeGUID>0x0002c9020025a678</NodeGUID>

<PortNum>1</PortNum>

<NodeDesc>mindy2 HCA-1</NodeDesc>

<NodeType>CA</NodeType>

</Port>

<Port>

<NodeGUID>0x00066a0007000e6d</NodeGUID>

<PortNum>4</PortNum>

<NodeDesc>SilverStorm 9080 GUID=0x00066a00da000159 Leaf 5, Chip A</NodeDesc>

<NodeType>SW</NodeType>

</Port>

</Link>

</LinkSummary>

<Nodes>

<CAs>

<Node>

<NodeGUID>0x0002c9020020e004</NodeGUID>

<NodeDesc>bender HCA-1</NodeDesc>

<NodeDetails>More details about node</NodeDetails>

</Node>

<Node>

<NodeGUID>0x0002c9020025a678</NodeGUID>

<NodeDesc>mindy2 HCA-1</NodeDesc>

<NodeDetails>Node details</NodeDetails>

</Node>

```



```
</CAs>

<Switches>

<Node>

<NodeGUID>0x00066a0007000df6</NodeGUID>

<NodeDesc>SilverStorm 9080 GUID=0x00066a00da000159 Leaf 4, Chip A</NodeDesc>

</Node>

<Node>

<NodeGUID>0x00066a0007000e6d</NodeGUID>

<NodeDesc>SilverStorm 9080 GUID=0x00066a00da000159 Leaf 5, Chip A</NodeDesc>

</Node>

</Switches>

<SMs>

<SM>

<NodeGUID>0x0002c9020025a678</NodeGUID>

<PortNum>1</PortNum>

<NodeDesc>mindy2 HCA-1</NodeDesc>

<PortGUID>0x00066a0007000e6d</PortGUID>

<NodeType>CA</NodeType>

<SMDetails>details about SM</SMDetails>

</SM>

</SMs>

</Nodes>

</Topology>
```

4.5.15 iba_findgood

The `iba_findgood` command can check for hosts which are pingable, ssh'able and active on the True Scale Fabric and produce a list of good hosts meeting all criteria. The resulting *good* file can then be used in as input to create `mpi_hosts` files for use running `mpi_apps` and the HCA-SW cabletest. Typical usage would be to identify good hosts which will undergo further testing and benchmarking during initial cluster staging and startup. This command assumes the Node description for each host will be based on the `hostname -s` output in conjunction with an optional HCA-# suffix. These names are the default when using OFED.

When using a `/etc/sysconfig/iba/hosts` file which lists the IPoIB hostnames, this assumption may not be correct. The files created (*good*, *alive*, *running*, *active*, *bad*) are in `iba_sorthosts` order with all duplicates removed.



This command automatically generates the file `FF_RESULT_DIR/punchlist.csv`. This file provides a concise summary of the bad hosts found. This can be imported into excel directly as a *.csv file, or be cut/pasted into Excel, and then the "Data/Text to Columns" toolbar can be used to separate the information into multiple columns at the semicolons. Following is a sample of the output that is generated:

```
2012/01/06 11:13:48;trash;Doesn't ping
2012/01/06 11:13:48;mybadhost;Can't ssh
2012/01/06 11:13:48;mindy;No active IB port
```

For a given run a line is generated for each failing host. Hosts are reported exactly once for a given run. Therefore, a host that does not ping will NOT be listed as "can't ssh" nor "No active IB port". It should be noted that there may be cases where IB ports could be active for hosts that do not ping, especially if Ethernet host names are being used for the ping test. However, the lack of ping often implies there are other fundamental issues (e.g., PXE boot, inability to access DNS or DHCP to get proper host name and IP address, etc.), which implies that reporting hosts that do not ping also lack active IB ports will typically be of limited value.

Note that the approach `iba_findgood` uses to determine hosts with active IB ports is to query the SA for NodeDescriptions. As such, ports may be active for hosts that cannot be ssh'ed or pinged.

4.5.15.1 Usage

```
iba_findgood [-RA] [-d dir] [-f hostfile]] [-h 'hosts'] [-t portsfile] [-p ports]
```

or

```
iba_findgood --help
```

4.5.15.2 Options

--help - Produce full help text

-R - Skip the running test (ssh), recommended if password-less ssh not setup.

-A - Skip the active test, recommended if True Scale Fabric software or fabric is not up.

-d *dir* - Directory in which to create alive, active, running, good and bad files default is `/etc/sysconfig/iba`.

-f *hostfile* - File with hosts in cluster, default is `/etc/sysconfig/iba/hosts`.

-h *hosts* - List of hosts to ping

-t *portsfile* - File with list of local HCA ports used to access fabric(s) for analysis. The default is `/etc/sysconfig/iba/ports`.

-p *ports* - List of local HCA ports used to access fabric(s) for analysis. The default is the first active port.

This is specified as `hca:port`

0:0 - First active port in system

0:y - Port y within system

x:0 - First active port on HCA x

x:y - HCA x, port y

The first HCA in the system is 1. The first port on an HCA is 1.



The files `alive`, `running`, `active`, `good`, and `bad` are created in the selected directory listing hosts passing each criteria. The `good` file can be used as input for an `mpi_hosts`. It will list each good host exactly once.

4.5.15.3 Usage Examples

```
iba_findgood  
iba_findgood -f allhosts
```

4.5.16 iba_saquery

(All) `iba_saquery` can perform various queries of the subnet manager/subnet agent and provide detailed fabric information.

Note: In past releases this command was named `saquery`. That name has been deprecated. It will continue to work for near term QuickSilver releases, but may be removed in the future. The `iba_saquery` command name must be used when running Intel® True Scale Fabric OFED+ Host Software.

In many cases `iba_report` and `iba_reports` provide a more powerful tool, however in some cases `iba_saquery` is preferred, especially when dealing with virtual fabrics, service records and multicast.

The command `iba_saquery` is installed on all hosts as part of the True Scale Fabric stack, but it is also included in Intel® FastFabric Toolset. As such it can be a useful tool to run on the Intel® FastFabric Toolset host and is therefore also documented here.

By default `iba_saquery` uses the first active port on the local system. However, if the Fabric Management Node is connected to more than one fabric (for example, a subnet), the HCA and port may be specified to select the fabric whose SA is to be queried.

4.5.16.1 Usage

```
iba_saquery [-v] [-h hca] [-p port] [-o type] [-l lid] [-t type] [-s guid]  
[-n guid] [-g guid] [-k pkey] [-i vfIndex] [-S serviceId] [-L sl] [-u gid] [-m gid]  
[-d name] [-P 'guid guid'] [-G 'gid gid'] [-a 'sguid...;dguid...']  
[-A 'sgid...;dgid...']
```

or

```
iba_saquery --help
```

4.5.16.2 Options

- `--help` - Produce full help text
- `-v/--verbose` - Verbose output
- `-h/--hca hca` - HCA to send by, default is 1st HCA
- `-p/--port port` - Port to send by, default is 1st active port
- `-l/--lid lid` - Query a specific lid
- `-k/--pkey pkey` - Query a specific pkey
- `-i/--vfindex vfIndex` - Query a specific vfindex
- `-S/--serviceId serviceId` - Query a specific service id



-L/--SL *SL* - Query by service level
 -t/--type *type* - Query by node type
 -s/--sysguid *guid* - Query by system image guid
 -n/--nodeguid *guid* - Query by node guid
 -g/--portguid *guid* - Query by port guid
 -u/--portgid *gid* - Query by port gid
 -m/--mcgid *gid* - Query by multicast gid
 -d/--desc *name* - Query by node name/description
 -P/--guidpair *guid guid* - Query by a pair of port Guides
 -G/--gidpair *gid gid* - Query by a pair of Gids
 -a/--guidlist *sguid ...;dguid ...* - Query by a list of port GUIDs
 -A/--gidlist *sgid ...;dgid ...* - Query by a list of Gids
 -o/--output *type* - Output type for query (default is node)

4.5.16.3 Node Types

ca - Channel adapter
 sw - Switch
 rtr - Router

4.5.16.4 GIDs

Specify a 64 bit subnet and 64 bit interface ID as:
subnet:interface.

For example:

0xfe80000000000000:0x00066a00a0000380

4.5.16.5 Output Types

systemguid - List of system image guides
 nodeguid - List of node guides
 portguid - List of port guides
 lid - List of lids
 desc - List of node descriptions/names
 path - List of path records
 node - List of node records
 portinfo - List of port info records
 sminfo - List of SM info records



swinfo – List of switch info records
vswinfo – List of vendor switch info records
link – List of link records
slvl – List of SL to VL mapping table records
vlarb – List of VL arbitration table records
pkey – List of P-Key table records
guids – List of GUID info records
service – List of service records
mcmember – List of multicast member records
inform – List of inform info records
linfdb – List of switch linear FDB records
ranfdb – List of switch random FDB records
mcfdb – List of switch multicast FDB records
trace – List of trace records
vfinfo – List of vFabrics
vfinfocsv – List of vFabrics in CSV format
vfinfocsv2 – List of vFabrics in CSV format with enums

The `vfinfocsv` and `vfinfocsv2` output formats are designed to make it easier to script `vfinfo` queries. One line is output per vFabric of the form:

```
name:index:pkey:sl:mtu:rate
```

The only difference between these two formats is how the `mtu` and `rate` are output. `vfinfocsv` outputs them in human/text format such as 2048 and 40g. `vfinfocsv2` outputs them as the IBTA enumerations defined for the SMA protocol such as 4 and 7. The `iba_getvf` command for a useful tool which is based on this capability of `iba_saquery`.

Table 6 shows the combinations of input (assorted query by options) and output (`-o`) that are permitted.



Table 6. Input Combinations

Input option	-O output permitted	-O output not permitted
None	systemguid, nodeguid, portguid, lid, desc, path, node, portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, service, mcmember, inform, <i>linfdb</i> , <i>ranfdb</i> , <i>mcfdb</i> , vfinfo, vfinfocsv, vfinfocsv2, vswinfo	trace
-t <i>node_type</i>	systemguid, nodeguid, portguid, lid, link, desc, <i>path</i> , node	portinfo, sminfo, swinfo, slvl, vlarb, pkey, guids, service, mcmember, inform, <i>linfdb</i> , <i>ranfdb</i> , <i>mcfdb</i> , trace, vfinfo, vfinfocsv, vfinfocsv2, vswinfo
-l <i>lid</i>	systemguid, nodeguid, portguid, lid, desc, path, node, portinfo, swinfo, slvl, vlarb, pkey, guids, service, mcmember, <i>linfdb</i> , <i>ranfdb</i> , <i>mcfdb</i> , vswinfo	sminfo, link, inform, trace, vfinfo, vfinfocsv, vfinfocsv2
-k <i>pkey</i>	path, vfinfo, vfinfocsv, vfinfocsv2	systemimageguid, nodeguid, portguid, lid, desc, node, portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, service, mcmember, inform, <i>linfdb</i> , <i>ranfdb</i> , <i>mcfdb</i> , vswinfo
-i <i>vindex</i>	vfinfo, vfinfocsv, vfinfocsv2	systemimageguid, nodeguid, portguid, lid, desc, node, portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, service, mcmember, inform, <i>linfdb</i> , <i>ranfdb</i> , <i>mcfdb</i> , vswinfo
-s <i>system_image_guid</i>	systemguid, nodeguid, portguid, lid, desc, <i>path</i> , node	portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, service, mcmember, inform, <i>linfdb</i> , <i>ranfdb</i> , <i>mcfdb</i> , trace, vfinfo, vfinfocsv, vfinfocsv2, vswinfo
-n <i>node_guid</i>	systemguid, nodeguid, portguid, lid, desc, <i>path</i> , node	portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, service, mcmember, inform, <i>linfdb</i> , <i>ranfdb</i> , <i>mcfdb</i> , trace, vfinfo, vfinfocsv, vfinfocsv2, vswinfo
-g <i>port_guid</i>	systemguid, nodeguid, portguid, lid, desc, path, node, service, mcmember, inform, trace	portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, <i>linfdb</i> , <i>ranfdb</i> , <i>mcfdb</i> , vfinfo, vfinfocsv, vfinfocsv2, vswinfo
-u <i>port_gid</i>	path, service, mcmember, inform, trace	systemguid, nodeguid, portguid, lid, desc, node, portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, <i>linfdb</i> , <i>ranfdb</i> , <i>mcfdb</i> , vfinfo, vfinfocsv, vfinfocsv2, vswinfo
-m <i>multicast_gid</i>	mcmember, vfinfo, vfinfocsv, vfinfocsv2	systemguid, nodeguid, portguid, lid, desc, path, node, portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, service, inform, <i>linfdb</i> , <i>ranfdb</i> , <i>mcfdb</i> , trace, vswinfo
-d <i>name</i>	systemguid, nodeguid, portguid, lid, desc, <i>path</i> , node	portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, service, mcmember, inform, <i>linfdb</i> , <i>ranfdb</i> , <i>mcfdb</i> , trace, vfinfo, vfinfocsv, vfinfocsv2, vswinfo

Table 6. Input Combinations (Continued)

Input option	-O output permitted	-O output not permitted
<i>-P port_guid_pair</i>	path, trace	systemguid, nodeguid, portguid, lid, desc, node, portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, service, mcmember, inform, linfdb, ranfdb, mcfdb, vswinfo
<i>-S serviceId</i>	path, vfinfo, vfinfocsv, vfinfocsv2	systemimageguid, nodeguid, portguid, lid, desc, node, portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, service, mcmember, inform, linfdb, ranfdb, mcfdb, vswinfo
<i>-L SL</i>	path, vfinfo, vfinfocsv, vfinfocsv2	systemimageguid, nodeguid, portguid, lid, desc, node, portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, service, mcmember, inform, linfdb, ranfdb, mcfdb, vswinfo
<i>-G gid_pair</i>	path, trace	systemguid, nodeguid, portguid, lid, desc, node, portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, service, mcmember, inform, linfdb, ranfdb, mcfdb, vswinfo
<i>-a port_guid_list</i>	<i>path</i>	systemguid, nodeguid, portguid, lid, desc, node, portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, service, mcmember, inform, linfdb, ranfdb, mcfdb, trace, vswinfo
<i>-A gid_list</i>	<i>path</i>	systemguid, nodeguid, portguid, lid, desc, node, portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, service, mcmember, inform, linfdb, ranfdb, mcfdb, trace, vswinfo

Note: In the table above, when using Intel® True Scale Fabric OFED+ Host Software, the combinations displayed in italics are currently not available. This includes the path-given input of node description, node type, system image guid, node guid, and port guid list.

4.5.17 iba_getvf

This command is designed to help when scripting application use of vFabrics, such as for mpirun parameters. It can fetch the Virtual Fabric info in a delimited format. It returns exactly 1 matching VF. When multiple VFs match the query, it prefers non-Default VFs which the calling server is a full member in. If multiple choices remain, it returns the one with the lowest VF Index (for example, Index typically represents order in config file). This algorithm is the same as that used by the Distributed SA.

The tool is intended to be part of additional scripts to help set PKey, SL, MTU and Rate when running MPI jobs.

4.5.17.1 Usage

```
iba_getvf [-h hca] [-p port] [-e] [-d vfname|-S serviceId|-m mcgid|-i
vfIndex|-k pkey|-L SL]
```

or

```
iba_getvf --help
```



4.5.17.2 Options

--help - Produce full help text

-h *hca* - HCA to send by, default is 1st HCA

-p *port* - Port to send by, default is 1st active port

-e - Output mtu and rate as enum values, 0=unspecified

-d *vfname* - Query by VirtualFabric Name

-S *serviceId* - Query by Application ServiceId

-m *gid* - Query by Application Multicast GID

-i *vfindex* - Query by VirtualFabric Index

-k *pkey* - Query by VirtualFabric PKey

-L *SL* - Query by VirtualFabric SL

4.5.17.3 Usage Examples

```
iba_getvf -d 'Compute'
iba_getvf -h 2 -p 2 -d 'Compute'
```

The output is of the form:

```
name:index:pkey:sl:mtu:rate
```

4.5.17.4 Sample Outputs

```
iba_getvf -d Default
Default:0:0xffff:0:unlimited:unlimited
```

Options allow for query by VF Name, VF Index, Service ID, MGID, PKey or SL.

Internally this is based on the `iba_saquery -o vfinfocsv` command

4.5.18 iba_getvf_env

This is a script designed to be included in bash scripts. It provides the `iba_getvf_func` and `iba_getvf2_func` shell functions which can be invoked to query a vFabric's parameters and export the values in the specified shell variables to indicate the PKEY, SL, MTU and RATE associated with the vFabric. An example of its use is provided in `/opt/iba/src/mpi_apps/ofed.openmpi.params`

4.5.19 iba_gen_ibnodes

This tool analyzes the present fabric and produces a list of Intel® Externally-Managed switches in the format required for use in the `/etc/sysconfig/iba/ibnodes` file.

4.5.19.1 Usage

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

or



```
iba_gen_ibnodes --help
```

4.5.19.2 Options

--help - Produce full help text

-t *portsfile* - File with list of local HCA ports used to access fabric(s) for analysis, default is /etc/sysconfig/iba/ports

-p *ports* - List of local HCA ports used to access fabric(s) for analysis. Default is 1st active port. This is specified as hca:port

0:0 = 1st active port in system

0:y = Port y within system

x:0 = 1st active port on HCA x

x:y = HCA x, port y

The first HCA in the system is 1. The first port on an HCA is 1.

-R - Do not attempt to get routes for computation of distance

-s - Update/resolve ibnodes switch names using topology XML data

-L *ibnodes_file* - Use *ibnodes_file* as ibnodes input (do not generate ibnodes data; must also use -s)

-o *output_file* - Write ibnodes data to *output_file* (default is stdout)

-T *topology_file* - Use *topology_file* as topology XML to update ibnodes NodeDesc values (may contain '%P'; must also use -s)

-X *snapshot_file* - Use *snapshot_file* XML for fabric link information (may contain '%P'; must also use -s)

-v *level* - Verbose level (0-8, default 0)

0 - No output

1 - Progress output

2 - Reserved

4 - Time stamps

8 - Reserved

-K - Do not clean temporary files

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

-X specifies a snapshot file to be used in conjunction with a topology file as previously described.

-L specifies the name of a pre-existing ibnodes file to be used in conjunction with a topology file. When specified, the ibnodes file will be used instead of ibnodes data obtained from the actual fabric. The updated ibnodes data is output to stdout (common to all `iba_gen_ibnodes` operations).



-v specifies an output verbosity level. Progress reports (including time stamps, if desired) are output.

-K is used to prevent temporary files from being removed. Temporary (CSV) files contain lists of links used during script operation. The files are not normally needed after execution, but they can be retained for subsequent inspection or processing.

4.5.19.3 Environment

ports – List of ports, used in absence of **-t** and **-p**

portsfile – File containing list of ports, used in absence of **-t** and **-p**

FF_TOPOLOGY_FILE – File containing topology XML data, used in absence of **-T**

4.5.19.4 Usage Examples

```
iba_gen_ibnodes
```

```
iba_gen_ibnodes -p '1:1 1:2 2:1 2:2'
```

```
iba_gen_ibnodes -o ibnodes
```

```
iba_gen_ibnodes -s -o ibnodes
```

```
iba_gen_ibnodes -L ibnodes -s -o ibnodes
```

```
iba_gen_ibnodes -s -T topology.%P.xml
```

```
iba_gen_ibnodes -L ibnodes -s -T topology.%P.xml -X snapshot.%P.xml
```

4.5.20 iba_gen_chassis

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

4.5.20.1 Usage

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

OR

```
iba_gen_chassis --help
```

4.5.20.2 Options

--help – Produce full help text

-t *portsfile* – File with list of local HCA ports used to access fabric(s) for analysis, default is `/etc/sysconfig/iba/ports`

-p *ports* – List of local HCA ports used to access fabric(s) for analysis. Default is 1st active port. This is specified as `hca:port`

`0:0` – First active port in system

`0:y` – Port *y* within system

`x:0` – First active port on HCA *x*

`x:y` – HCA *x*, port *y*

The first HCA in the system is 1. The first port on an HCA is 1.



4.5.20.3 Environment

ports – List of ports, used in absence of *-t* and *-p*

portsfile – File containing list of ports, used in absence of *-t* and *-p*

4.5.20.4 Usage Examples

```
iba_gen_chassis
```

```
iba_gen_chassis -p '1:1 1:2 2:1 2:2'
```

```
iba_gen_chassis >> /etc/sysconfig/iba/chassis
```

or while editing the file use a vi command to include its output such as:

```
:r! iba_gen_chassis
```

4.5.21 iba_gen_esm_chassis

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

4.5.21.1 Usage

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

OR

```
iba_gen_esm_chassis --help
```

4.5.21.2 Options

--help - Produce full help text

-u user – User to perform command as for chassis default is admin

-S – Securely prompt for password for user on chassis

-t portsfile – File with a list of local HCA ports used to access fabric(s) for analysis, default is */etc/sysconfig/iba/ports*

-p ports – List of local HCA ports used to access fabric(s) for analysis. Default is 1st active port. This is specified as *hca:port*

0:0 – 1st active port in system

0:y – Port y within system

x:0 – 1st active port on HCA x

x:y – HCA x, port y

The first HCA in the system is 1. The first port on an HCA is 1.

4.5.21.3 Environment

FF_CHASSIS_ADMIN_PASSWORD – Password for chassis, used in absence of *-S*

ports – List of ports, used in absence of *-t* and *-p*

portsfile – File containing list of ports, used in absence of *-t* and *-p*



4.5.21.4 Usage Examples

```
iba_gen_esm_chassis
iba_gen_esm_chassis -S -p '1:1 1:2 2:1 2:2'
iba_gen_esm_chassis >> /etc/sysconfig/iba/esm_chassis
```

or while editing the file use a vi command to include its output such as:

```
:r! iba_gen_esm_chassis
```

4.5.22 iba_fequery

(All): Is helpful when testing or debugging PA operations to the Fabric Executive (FE). This tool performs the custom PA client/server queries. The output formats and arguments are very similar to `iba_saquery`.

4.5.22.1 Usage:

```
iba_fequery [-v] [-a ipAdr | -h hostName] -o type [-g groupName] [-l nodeLid]
[-P portNumber] [-d delta] [-s select] [-f focus] [-S start] [-r range] [-n imgNum]
[-O imgOff] [-m moveImgNum] [-M moveImgOff]
```

4.5.22.2 Options:

- v - Verbose output
- a *ipAdr* - IP address of node running the FE
- h *hostName* - Host name of node running the FE
- o *type* - Output type
- g *groupName* - Group name for groupInfo query
- l *nodeLid* - Lid of node for portCounters query
- P *portNumber* - Port number for portCounters query
- d *delta* - Delta flag for portCounters query - 0 or 1
- s *select* - 16-bit select flag for clearing port counters
Select bits (0 is the least significant)
 - 0 - SymbolErrorCounter
 - 1 - LinkErrorRecoveryCounter
 - 2 - LinkDownedCounter
 - 3 - PortRcvErrors
 - 4 - PortRcvRemotePhysicalErrors
 - 5 - PortRcvSwitchRelayErrors
 - 6 - PortXmitDiscards
 - 7 - PortXmitConstraintErrors
 - 8 - PortRcvConstraintErrors
 - 9 - LocalLinkIntegrityErrors
 - 10 - ExcessiveBufferOverrunErrors
 - 11 - VL15Dropped
 - 12 - PortXmitData
 - 13 - PortRcvData



- 14 – PortXmitPackets
- 15 – PortRcvPackets
- f *focus* – Focus select value for getting focus ports
Focus select values:
 - 0x00020001 – Sorted by utilization - highest first
 - 0x00020081 – Sorted by packet rate - highest first
 - 0x00020101 – Sorted by utilization - lowest first
 - 0x00030001 – Sorted by integrity errors - highest first
 - 0x00030002 – Sorted by sma congestion errors - highest first
 - 0x00030003 – Sorted by congestion errors - highest first
 - 0x00030004 – Sorted by security errors - highest first
 - 0x00030005 – Sorted by routing errors - highest first
 - 0x00030006 – Sorted by adaptive routing - highest first
- S *start* – Start of window for focus ports – Should always be 0 for now
- r *range* – Size of window for focus ports list
- n *imgNum* – 64-bit image number – may be used with groupInfo, groupConfig, portCounters (delta)
- O *imgOff* – Image offset – May be used with groupInfo, groupConfig, portCounters (delta)
- m *moveImgNum* – 64-bit image number; used with moveFreeze to move a freeze image
- M *moveImgOff* – Image offset – May be used with moveFreeze to move a freeze image

4.5.22.3 Output Types:

- classPortInfo – Class port info
- groupList – List of PA groups
- groupInfo – Summary statistics of a PA group - requires -g option for groupName
- groupConfig – Configuration of a PA group - requires -g option for groupName
- portCounters – Port counters of fabric port; requires -l (lid) and -P (port) options, -d (delta) is optional
- pmConfig – Retrieve PM configuration information
- freezeImage – Create freeze frame for image ID; requires -n (imgNum)
- releaseImage – Release freeze frame for image ID; requires -n (imgNum)
- renewImage – Renew lease for freeze frame for image ID; requires -n (imgNum)
- moveFreeze – Move freeze frame from image ID to new image ID; requires -n (imgNum) and -m (moveImgNum)
- focusPorts – Get sorted list of ports using utilization or error values (from group buckets)



`imageInfo` – Get information about a PA image (timestamps, etc.); requires `-n (imgNum)`

4.5.22.4 Usage Examples:

```
iba_fequery -o classPortInfo

iba_fequery -h stewart -o classPortInfo

iba_fequery -a 172.21.2.155 -o classPortInfo

iba_fequery -o groupList

iba_fequery -o groupInfo -g All

iba_fequery -o groupConfig -g All

iba_fequery -h stewart -o groupInfo -g All

iba_fequery -a 172.21.2.155 -o groupInfo -g All

iba_fequery -o portCounters -l 1 -P 1 -d 1

iba_fequery -o portCounters -l 1 -P 1 -d 1 -n 0x20000000d02 -O 1

iba_fequery -o pmConfig

iba_fequery -o freezeImage -n 0x20000000d02

iba_fequery -o releaseImage -n 0xd01

iba_fequery -o renewImage -n 0xd01

iba_fequery -o moveFreeze -n 0xd01 -m 0x20000000d02 -M -2

iba_fequery -o focusPorts -g All -f 0x00030001 -S 0 -r 20

iba_fequery -o imageInfo -n 0x20000000d02
```

4.5.23 iba_smaquery

(All) This tool can perform the majority of IBTA defined SMA queries and display the resulting response. It should be noted that each query is issued directly to the SMA and does not involve SM interaction.

4.5.23.1 Usage

```
iba_smaquery [-v] [-d] [-o otype] [-l lid]
[-m dest_port|inport,output] [-h hca] [-p port] [-K mkey]
[-f flid] [-b block] [hop hop ...]
```

4.5.23.2 Options

`-v` – Verbose output

`-d` – Turn on debug

`-o otype` – Output type. Valid *otypes* are: `nodeDesc` (or `desc`), `nodeInfo` (or `node`), `portInfo`, `smInfo`, `swInfo`, `slvl`, `vlarb`, `pkey`, `guids`, `linfdb`, `ranfdb`, `mcfdb`, `vswInfo`, `portgroup`, `lidmask`.

`-l lid` – Destination lid, default is local port



`-m dest_port` – Port in destination device to query

`inport, outport` – SLVL's input/output port – Switch only. Default is to show all port combinations

`-h hca` – HCA to send by, default is first HCA

`-p port` – Port to send by, default is port 1

`-K mkey` – SM management key to access remote ports

`-f flid` – LID to lookup in forwarding table to select which LFT or MFT block to display. Default is to show entire table

`-b block` – Block number of either guides, pkey, or ranfdb. Default is to show entire table

4.5.23.3 Usage Examples

```
iba_smaquery -o nodedesc -l 6           # get nodedesc via lid routed
iba_smaquery -o nodedesc 1 3           # get nodedesc via directed route
                                         # (2 dr hops)
iba_smaquery -o nodeinfo -l 2 3       # get nodeinfo via a combination of
                                         # lid routed and directed route
                                         # (1 dr hop)
iba_smaquery -o portinfo              # get local port info
iba_smaquery -o portinfo -l 6 -m 1    # get port info of port 1 of lid 6
iba_smaquery -o slvl -l 6             # get slvl of CA at lid 6
iba_smaquery -o slvl -l 2 -m 2,3      # get slvl of Switch at lid 2
                                         # with input port =2,output port=3
iba_smaquery -o mcfdb -l 2 -f 0xc004  # get a block of entries that
                                         # includes mc lid 0xc004 from
                                         # the MFT of the switch with lid 2
iba_smaquery -o mcfdb -l 2 -f 0xc004 -m 17 # same as above with position bit
iba_smaquery -o linfdb -l 2 -m 1 -f 1   # get a block of entries of port 1 that
                                         # includes lid 1 entry from LFT
iba_smaquery -o ranfdb -l 2 -b 5       # get a fixed 10 blocks starting
                                         # from block 5 from RFT
iba_smaquery -o guides -l 6 -b 0      # get block 0 of GUIDs
iba_smaquery -o vlarb -l 6            # get vlarb table entries from lid 6
iba_smaquery -o pkey -l 2 3           # get pkey table entries starting
```



```

# (lid routed to lid 2,
# then 1 dr hop to port 3)

iba_smaquery -o swinfo -l 2      # get switch info
iba_smaquery -o sminfo -l 1      # get SM info
iba_smaquery -o vswinfo -l 2     # get vendor switch info
iba_smaquery -o portgroup -l 2   # get port group info
iba_smaquery -o lidmask -l 2     # get lidmask info

```

4.5.24 iba_paquery

(All) `iba_paquery` can perform various queries of the performance management/performance administration agent and provide details about fabric performance. Refer to the *Intel® True Scale Fabric Suite Fabric Manager User Guide* for a description of the operation and client services of the PM/PA.

`iba_paquery` is included in the Intel® FastFabric Toolset. Its operation is dependent on a Intel® True Scale Fabric Suite Fabric Manager version 6.0 or greater running as master SM/PM in the fabric.

By default, `iba_paquery` uses the first active port on the local system. However, if the Fabric Management Node is connected to more than one fabric (for example, a subnet), the HCA and port may be specified to select the fabric whose PA is to be queried.

4.5.24.1 Usage

```

iba_paquery [-v] [-h hca] [-p port] -o type [-g groupName] [-l nodeLid] [-P
portNumber] [-d delta] [-s select] [-f focus] [-S start] [-r range] [-n imgNum]
[-O imgOff] [-m moveImgNum] [-M moveImgOff]

```

4.5.24.2 Options

```

-v/--verbose - Verbose output

-h/--hca hca - HCA to send by, default is 1st HCA

-p/--port port - Port to send by, default is 1st active port

-o/--output type - Output type

-g/--groupName groupName - Group name for groupInfo query

-l/--lid nodeLid - LID of node for portCounters query

-P/--portNumber portNumber - Port number for portCounters query

-d/--delta delta - Delta flag for portCounters query - 0 or 1

-s/--select select - 16-bit select flag for clearing port counters
    Select bits (0 is least significant (right-most))
    0 - SymbolErrorCounter
    1 - LinkErrorRecoveryCounter
    2 - LinkDownedCounter
    3 - PortRcvErrors
    4 - PortRcvRemotePhysicalErrors

```



5 – PortRcvSwitchRelayErrors
6 – PortXmitDiscards
7 – PortXmitConstraintErrors
8 – PortRcvConstraintErrors
9 – LocalLinkIntegrityErrors
10 – ExcessiveBufferOverrunErrors
11 – VL15Dropped
12 – PortXmitData
13 – PortRcvData
14 – PortXmitPackets
15 – PortRcvPackets

-f/--focus *focus* – Focus select value for getting focus ports focus select values:
0x00020001 – sorted by utilization - highest first
0x00020081 – sorted by packet rate - highest first
0x00020101 – sorted by utilization - lowest first
0x00030001 – sorted by integrity errors - highest first
0x00030002 – sorted by sma congestion errors - highest first
0x00030003 – sorted by congestion errors - highest first
0x00030004 – sorted by security errors - highest first
0x00030005 – sorted by routing errors - highest first
0x00030006 – sorted by adaptive routing - highest first

-S/--start *start* – Start of window for focus ports - Should always be 0 for now

-r/--range *range* – Size of window for focus ports list

-n/--imgNum *imgNum* – 64-bit image number - May be used with *groupInfo*, *groupConfig*, *portCounters* (delta)

-O/--imgOff *imgOff* – Image offset - May be used with *groupInfo*, *groupConfig*, *portCounters* (delta)

-m/--moveImgNum *moveImgNum* – 64-bit image number - Used with *moveFreeze* to move a freeze image

-M/--moveImgOff *moveImgOff* – Image offset - May be used with *moveFreeze* to move a freeze image

4.5.24.3 Output Types

classPortInfo – Class port info

groupList – List of PA groups

groupInfo – Summary statistics of a PA group – Requires -g option for *groupName*

groupConfig – Configuration of a PA group – Requires -g option for *groupName*

portCounters – Port counters of fabric port – Requires -l *lid* and -P *port* options, -d *delta* is optional

clrPortCounters – Clear port counters of fabric port – Requires -l *lid* and -P *port*, and -s *select* options

clrAllPortCounters – Clear all port counters in fabric



pmConfig – Retrieve PM configuration information

freezeImage – Create freeze frame for image ID – Requires `-n imgNum`

releaseImage – Release freeze frame for image ID – Requires `-n imgNum`

renewImage – Renew lease for freeze frame for image ID – Requires `-n imgNum`

moveFreeze – Move freeze frame from image ID to new image ID – Requires `-n imgNum` and `-m moveImgNum`

focusPorts – Get sorted list of ports using utilization or error values (from group buckets)

imageinfo – Get configuration of a PA image (timestamps, etc.) – Requires `-n imgNum`

4.5.24.4 Usage Examples

```
iba_paquery -o classPortInfo
iba_paquery -o groupList
iba_paquery -o groupInfo -g All
iba_paquery -o groupConfig -g All
iba_paquery -o portCounters -l 1 -P 1 -d 1
iba_paquery -o portCounters -l 1 -P 1 -d 1 -n 0x20000000d02 -O 1
iba_paquery -o clrPortCounters -l 1 -P 1 -s 0x0048 (clears PortXmitDiscards &
PortRcvErrors)
iba_paquery -o clrAllPortCounters -s 0x0048 (clears PortXmitDiscards &
PortRcvErrors)
iba_paquery -o getPMConfig
iba_paquery -o freezeImage -n 0x20000000d02
iba_paquery -o releaseImage -n 0xd01
iba_paquery -o renewImage -n 0xd01
iba_paquery -o moveFreeze -n 0xd01 -m 0x20000000d02 -M -2
iba_paquery -o focusPorts -g All -f 0x00030001 -S 0 -r 20
iba_paquery -o imageConfig -n 0x20000000d02
```

4.5.25 iba_pmaquery

(All) This is a low level tool which can perform individual PMA queries against a specific LID. It is very useful in displaying port runtime information.

4.5.25.1 Usage

```
iba_pmaquery [-v] [-d] [-o otype] [-l lid] [-m dest_port] [-b select] [-h hca] [-p
port]
```



4.5.25.2 Options

- v - Verbose output
- d - Turn on debug
- o *otype* - Output type. Valid *otypes* are: classportinfo, stats, extstats, clearstats, clearextstats, vendstats, clearvendstats
- l *lid* - Destination lid, default is local port
- m *dest_port* - Port in destination device to query/clear. Required when using -l option for all but option -o *classportinfo*
- b *select* - Counter select for clearstats, clearextstats, and clearvendstats. Default is to clear all.
- h *hca* - HCA to send by, default is 1st HCA
- p *port* - Port to send by, default is port 1

4.5.25.3 Usage Examples

```
iba_pmaquery -o classportinfo -l 6           # get PMA classportinfo
iba_pmaquery -o stats -l 6 -m 1              # get PMA PortCounters
iba_pmaquery -o clearstats -l 6 -m 1         # clear PMA PortCounters
iba_pmaquery -o clearstats -l 6 -m 1 -b 0xfff # clear PMA error PortCounters
iba_pmaquery -o extstats -l 6 -m 1           # get PMA PortCountersExtended
iba_pmaquery -o clearextstats -l 6 -m 1      # clear PMA PortCountersExtended
iba_pmaquery -o vendstats -l 6 -m 1          # get PMA Vendor PortCounters
iba_pmaquery -o clearvendstats -l 6 -m 1     # clear PMA PortCounters
```

4.5.25.4 Sample Outputs

```
[root@luanne ~]# iba_pmaquery
Performance: Transmit

    Xmit Data                0 MiB (72 Quads)
    Xmit Pkts                 1

Performance: Receive

    Rcv Data                 0 MiB (0 Quads)
    Rcv Pkts                 0

Errors:

    Symbol Errors            0
    Link Error Recovery      0
    Link Downed              0
    Port Rcv Errors          0
```



Port Rcv Rmt Phys Err	0
Port Rcv Sw Relay Err	0
Port Xmit Discards	2
Port Xmit Constraint	0
Port Rcv Constraint	0
Local Link Integrity	0
Exc. Buffer Overrun	0
VL15 Dropped	2

4.5.26 iba_ccaquery

The `iba_ccaquery` queries CCA on S20, Intel® True Scale HCAs, and Mellanox Devices. It will decide if the vendor specific CCA packets should be used or standard packets.

4.5.26.1 Usage

```
iba_ccaquery [-v] [-d] [-o otype] [-l lid] [-h hca] [-p port] [-K ckey] [-b block]
```

4.5.26.2 Options

- v – Verbose output
- d – Turn on debug
- o *otype* – Output type. Valid *otypes* are `classportinfo`, `key`, `info`, `log`, `swsetting`, `swportsetting`, `casetting`, `ctltable`, `timestamp`
- l *lid* – Destination lid, default is local port
- h *hca* – HCA to send via, default is 1st HCA
- p *port* – Port to send via, default is port 1
- K *ckey* – CC management key to access remote ports
- b *block* – Block number of `swportsetting` or `ctltable`. Default is to show entire table

4.5.26.3 Examples

```
iba_ccaquery -o classportinfo -l 6      # get CCA classportinfo
iba_ccaquery -o info -l 6               # get CCA Info
iba_ccaquery -o key -l 6                # get CCA KeyInfo
iba_ccaquery -o log -l 6                # get event log
iba_ccaquery -o swsetting -l 6          # get CCA Switch Setting
iba_ccaquery -o swportsetting -l 6 -b 0 # get CCA Switch Port Settings block 0
iba_ccaquery -o casetting -l 2          # get CCA CA Setting
iba_ccaquery -o ctltable -l 2           # get CCA CA Control Table
iba_ccaquery -o timestamp -l 2          # get CCA running timestamp
```

4.5.27 iba_smjobmgmt

`iba_smjobmgmt` is a command line tool which provides a means to query the SM for information about active MPI jobs as well as terminate MPI jobs if necessary.

`iba_smjobmgmt` can perform queries on multiple fabrics and allows multiple ways to specify the fabrics as well as the jobs.

An MPI job is initiated with a list of HCA port GUIDs for the hosts upon which the job will execute (port GUID vector). Associated with the port GUID vector, the SM provides a list of switches to which the HCA ports connect (switch map) and a table of 'costs' ('cost matrix') encoding the number of hops and data bandwidth between each pair of switches in the switch map.

The switch map is a list of integers (SM-generated, starting from 1) with each switch map entry[n] giving the switch number to which the HCA port GUID vector[n] connects. As multiple HCAs connect to the same switch, they will have the same switch number in their switch map entry.

The cost matrix is a table of size NxN where N is the number of switches in the switch map. Entry (a,b) in the table represents the cost for data traveling from switch a to switch b. Entry (a,b) has the same cost as (b,a), and the cost for every entry (a,a) is 0.

`iba_smjobmgmt` specifies the fabrics on which to operate by the ports on the host running `iba_smjobmgmt`. The ports can be specified in three (3) mutually exclusive ways:

- As a single `hca_number/port_number` pair (both need not be specified)
- As one or more portguid values
- All (all ports on all HCAs of the host)

For MPI job queries, MPI jobs (previously created) can be specified by seven (7) methods and work in combination:

- As one or more `job_id` values
- As one or more `job_name` values
- As one or more `application_name` values
- As one or more PID values
- As one or more UID values
- As one or more `time_stamp` values
- All (all jobs on the specified port(s)); cannot be used with other job specification methods

If more than one method is used to specify jobs (for example, `job_id` and `job_name`), a job must be matched by all specified methods to be considered matched ('AND' operation). Within a method if more than one specification is made (for example, multiple `job_id` values) a job must match at least one specification to be considered matched ('OR' operation).

4.5.27.1 Usage

```
show | clear [-g portguid] [-h hca] [-p port] [-j job_id] [-J job_name]
[-N app_name] [-i pid] [-I uid] [-t timestamp] [-T timestamp]
[-a] [-f] [-c] [-s] [-u] [-S index] [-v]
```

4.5.27.2 Options

`-v/--verbose` - Verbose output



`-h/--hca hca` - A desired HCA to bind to. The default is the first HCA. Can be specified at most 1 time on the command line. Not valid with `-g` or `-f`.

`-p/--port port` - A desired HCA port to bind to. The default is the first port. Can be specified at most 1 time on the command line. Not valid with `-g` or `-f`.

`-g/--portguid port_guid` - The PortGuid of a port to bind to. Can be specified multiple times on the command line for multiple ports (fabrics). No default value. Not valid with `-h`, `-p` or `-f`.

`-j/--jobid job_id` - A job id to query for. Can be specified multiple times on the command line for multiple jobs. `job_id` is specified as a hexadecimal number (with or without a leading '0x'). If the number of digits `n` specified is 8 or less, then job id is taken as a partial value and is compared against the most significant `n` digits of the `job_ids` in the active list. The remaining (least significant) digits are not compared. Not valid with `-a`.

`-J/--jobname job_name` - A job name to query for. Can be specified multiple times on the command line. `job_name` can match any portion of a job name string. Not valid with `-a`.

`-N/--appname app_name` - An application name to query for. Can be specified multiple times on the command line. `app_name` can match any portion of an application name string. Not valid with `-a`.

`-i/--pid pid` - A job PID to query for. Can be specified multiple times on the command line. PID can be specified in any desired radix, but must match the exact PID of a job. Not valid with `-a`.

`-I/--uid uid` - A job UID to query for. Can be specified multiple times on the command line. UID can be specified in any radix, but must match the exact UID of a job. Not valid with `-a`.

`-t/--timeless timestamp` - A time stamp to query for. Can be specified multiple times on the command line. `timestamp` must be of the form `mm/dd/yyyy hh:mm:ss`. The time stamp of a job must be less than the specified timestamp. If `-t` and `-T` are used multiple times or together, the timestamp of a job is required to match only 1 specification. Not valid with `-a`.

`-T/--timemore timestamp` - A time stamp to query for. Same as `-t` except that the time stamp of a job must be greater than the specified timestamp. Not valid with `-a`.

`-a/--alljobs` - All jobs. Not valid with `-j`, `-J`, `-N`, `-i`, `-I`, `-t` or `-T`.

`-f/--allfabrics` - All fabrics (ports). Not valid with `-h`, `-p` or `-g`.

`show` - displays information about the matched jobs on the specified ports. If no port specification is made, the default is all ports on all HCAs. If no job specification is made, the default is all jobs.

By default, `show` displays a basic list of job information for each job. The basic information is one line and includes the following: `job_id`, `routed`, `has_use` and job name. If `-v` is specified, `timestamp`, `app_name`, `pid` and `uid` are also included (on multiple lines). If `-s`, `-c` or `-u` are also specified, the corresponding port GUID vector, switch map, cost matrix or use matrix for the job is displayed.

4.5.27.3 Show Options

`-s/--switch` - Display the port GUID vector and the switch map for the queried job(s).



`-c/--cost` – Display the cost matrix for the queried job(s) starting at the switch index value specified by `-S` (default 0). Up to 14 cost matrix values will be displayed per output line.

`-S/--index index` – Starting switch index value for cost matrix display. Default value is 0.

`-u/--use` – Display the use matrix for the queried job(s).

`-v/--verbose` – Include *timestamp*, *app_name*, *pid* and *uid* information with *job_info* information.

`clear` – clears (completes) the specified jobs on the specified fabrics. If no port specification is made, the default is the first port on the first HCA. If no job specification is made, the default is no jobs.

By default (with no port or job specifications), `clear` displays only the number of active jobs on the default port, but does not clear any jobs. Ports and jobs can be specified in the same ways as `show` and the specified jobs will be completed using. As each complete command is issued, the *job_id* of the job will be displayed.

4.5.27.4 Examples

Show a job summary for the fabrics on the indicated 2 ports:

```
iba_smjobmgmt show -g 0x0011750000FF8F4C -g 0x00066A01A0006F74
```

Show job information (verbose), including the cost matrix starting at switch index 6, for each job on the fabric on port 2 of the first HCA:

```
iba_smjobmgmt show -v -c -S 6 -p 2
```

Show a job summary, port GUID vector, and switch map for all jobs on the indicated port fabric whose time stamp is less than '06/24/2009 14:07:30':

```
iba_smjobmgmt show -g 0x00066A01A0006F74 -s -t "06/24/2009 14:07:30"
```

Clear jobs on the first port of the first HCA whose *job_id* begins with 0x4D OR 0x7A, AND whose application name contains the string "APL23":

```
iba_smjobmgmt clear -j 4D -j 7A -N APL23
```

4.5.28 iba_smjobgen

`iba_smjobgen` is a command line tool which provides a means to list ports available from which to create MPI jobs as well as create MPI jobs and display job information (port GUID vector, switch map, cost matrix). See "[iba_smjobmgmt](#)" on page 244 for a description of MPI jobs and job information. The actual creation of a job by `iba_smjobgen` can be suppressed, while still allowing a query of the job information. `iba_smjobgen` works on a single fabric at a time, specified as an *hca_number/port_number* pair (both need not be specified) or as a portguid value.

The port GUID vector necessary to create an MPI job can be specified as a sequence of port GUIDs on the command line or as 'All', which will generate a list of all CA port GUIDs available on the specified fabric.

4.5.28.1 Usage

```
iba_smjobgen show | create [-g portguid] [-h hca] [-p port] [-J job_name] [-N app_name] [-i pid] [-I uid] [-a] [-G portguid] [-n] [-c] [-s] [-u] [-S index] [-v]
```



4.5.28.2 Options

`-h/--hca hca` - A desired HCA to bind to. The default is the first HCA. Not valid with `-g`.

`-p/--port port` - A desired HCA port to bind to. The default is the first port. Not valid with `-g`.

`-g/--portguid port_guid` - The PortGuid of a port to bind to. No default value. Not valid with `-h` or `-p`.

`-a/--allports` - Show/create all CA ports on a fabric. Not valid with `-G`.

`-G/--guid portguid` - Show/create with the CA PortGuid on a fabric. Can be specified multiple times on the command line. Not valid with `-a`.

`show` - Displays a list of CA port GUIDs for a job (either explicitly specified with `-G` or generated with `-a`). The default is all CA ports on a fabric (`-a`).

`create` - Creates an MPI job using a port GUID vector specified with `-a` or `-G` (no default). By default, `create` displays the job ID of the created job. Optionally, port GUID vector, switch map, cost matrix and job parameters can also be displayed. The actual creation of a job can be suppressed with `-n`; optional parameters can still be displayed.

4.5.28.3 Create Options

`-J/--jobname job_name` - The job name for a created job.

`-N/--appname app_name` - The application name for a created job.

`-i/--pid pid` - The job PID for a created job.

`-I/--uid uid` - The job UID for a created job.

`-n/--nocreate` - Don't create job.

`-s/--switch` - Display the port GUID vector and the switch map for a job.

`-c/--cost` - Display the cost matrix for a job starting at the switch index value specified by `-S` (default 0). Up to 14 cost matrix values will be displayed per output line.

`-S/--index index` - Starting switch index value for cost matrix display. Default value is 0.

`-v/--verbose` - Display the job name, application name, PID and UID information for a job.

4.5.28.4 Examples

Show a list of all CA port GUIDs on the first port of the first HCA:

```
iba_smjobgen show
```

Show the cost matrix for a (non-created) job using all available CA port GUIDs on a fabric:

```
iba_smjobgen create -n -a -c
```

Show job information for a (non-created) job using all available CA port GUIDs on the fabric at the specified portguid:



```
iba_smjobgen create -n -g 0x00117500005A6E8A -a -s -c
```

Create a default job using all available CA port GUIDs and show cost matrix information for the job:

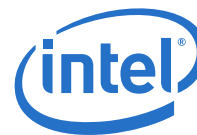
```
iba_smjobgen create -a -c
```

Output

```
iba_smjobgen: Create Job: ID:0x553A81674E84734F
```

```
Cost Matrix (hex): Switches:24
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	
0:	0000	0006	0006	0006	0006	000C	000C	000C	000C	000C	000C	000C	000C	0012	0012
1:	0006	0000	000C	000C	000C	0006	0006	0006	0012	0012	0012	0012	000C	000C	
2:	0006	000C	0000	000C	000C	000C	0012	0012	0006	0006	0012	0012	0012	0012	0018
3:	0006	000C	000C	0000	000C	0012	0006	0012	0006	0012	0006	000C	0018	000C	
4:	0006	000C	000C	000C	0000	0012	0012	0006	0012	0006	000C	0006	000C	0018	
5:	000C	0006	000C	0012	0012	0000	000C	000C	0018	0012	0018	0018	0006	0012	
6:	000C	0006	0012	0006	0012	000C	0000	000C	000C	0018	000C	0012	0012	0006	
7:	000C	0006	0012	0012	0006	000C	000C	0000	0018	000C	0012	000C	0006	0012	
8:	000C	0012	0006	0006	0012	0018	000C	0018	0000	000C	000C	0012	001E	000C	
9:	000C	0012	0006	0012	0006	0012	0018	000C	000C	0000	0012	000C	000C	001E	
10:	000C	0012	0012	0006	000C	0018	000C	0012	000C	0012	0000	0006	0018	0012	
11:	000C	0012	0012	000C	0006	0018	0012	000C	0012	000C	0006	0000	0012	0018	
12:	0012	000C	0012	0018	000C	0006	0012	0006	001E	000C	0018	0012	0000	0018	
13:	0012	000C	0018	000C	0018	0012	0006	0012	000C	001E	0012	0018	0018	0000	
14:	0012	000C	0018	000C	0012	0012	0006	000C	0012	0018	0006	000C	0012	000C	
15:	0012	000C	0018	0012	000C	0012	000C	0006	0018	0012	000C	0006	000C	0012	
16:	0012	0012	000C	000C	0018	001E	000C	0018	0006	0012	0012	0018	0024	0006	
17:	0012	0012	000C	0018	000C	000C	0018	000C	0012	0006	0018	0012	0006	0024	
18:	0012	0018	000C	000C	0012	001E	0012	0018	0006	000C	0006	000C	0018	0012	
19:	0012	0018	000C	0012	000C	0018	0018	0012	000C	0006	000C	0006	0012	0018	
20:	0018	0012	0018	0018	0012	000C	0012	000C	0018	0012	0012	000C	0006	000C	
21:	0018	0012	001E	0012	0018	0018	000C	0012	0012	0018	000C	0012	000C	0006	
22:	0018	0018	0012	0012	0018	0024	0012	0018	000C	0012	000C	0012	0012	000C	
23:	0018	0018	0012	0018	0012	0012	0018	0012	0012	000C	0012	000C	000C	0012	



Create the named job using the specified list of CA port GUIDs and show all job information:

```
iba_smjobgen create -J Job123 -N Appl456 -i 0x0770123 -I 0xAB894321 -G
0x00117500005A6E8A -G 0x00117500005A6E84 -G 0x001175000079E4BC
-G 0x001175000079E39E -G 0x001175000079E45A -G 0x001175000079E360
-s -c -v
```

Output

```
iba_smjobgen: Create Job: ID:0xBF0ED6BD9304E30C
```

```
Job Parameters: Name:Job123 AppName:Appl456
```

```
PID:0x770123 UID:0xAB894321
```

```
PortGUID Vector: GUIDs:6      Switch Map: Switches:6

0: 0x00117500005A6E8A      0
1: 0x00117500005A6E84      1
2: 0x001175000079E4BC      2
3: 0x001175000079E39E      3
4: 0x001175000079E45A      4
5: 0x001175000079E360      5
```

```
Cost Matrix (hex): Switches:6

    0    1    2    3    4    5
0: 0000 0006 000C 0012 0012 0018
1: 0006 0000 0006 000C 0018 0018
2: 000C 0006 0000 0012 0018 0012
3: 0012 000C 0012 0000 000C 000C
4: 0012 0018 0018 000C 0000 0006
5: 0018 0018 0012 000C 0006 0000
```

4.5.29 iba_extract_bad_links

(Linux) Produces a csv file listing all the links that exceed the present or specified iba_report -o errors thresholds. The output from this tool can be reviewed and supplied as input to iba_disable_ports.



4.5.29.1 Usage

```
iba_extract_bad_links [iba_report options]
```

OR

```
iba_extract_bad_links --help
```

4.5.29.2 Options

iba_report options – Options will be passed to iba_report.

4.5.29.3 Examples

```
iba_extract_bad_links
```

```
iba_extract_bad_links -h 1 -p 2
```

4.5.30 iba_disable_ports

(Linux) Accepts a csv file listing links to disable. For each HCA-SW link, the switch side of the link is disabled. For each SW-SW link, the side of the link with the lower LID (that is typically the side closest to the SM) is disabled. This approach generally permits a future `iba_enable_ports` operation to re-enable the port once the issue is corrected or ready to be retested. When using the `-R` option this tool does not look at the routes, it disables the switch ports with the lower value LID. The list of disabled ports is tracked in `/etc/sysconfig/iba/disabled*.csv`.

4.5.30.1 Usage

```
iba_disable_ports [-R] [-t portsfile] [-p ports] [reason] < disable.csv
```

OR

```
iba_disable_ports --help
```

4.5.30.2 Options

--help – Produce full help text

-R – Do not attempt to get routes for computation of distance instead just disable switch port with lower LID assuming that will be closer to this node

-t portsfile – File with list of local HCA ports used to access fabric(s) for operation, default is `/etc/sysconfig/iba/ports`

-p ports – List of local HCA ports used to access fabric(s) for analysis default is 1st active port.

This is specified as `hca:port`

`0:0` – First active port in system

`0:y` – Port y within system

`x:0` – First active port on HCA x

`x:y` – HCA x, port y

The first HCA in the system is one. The first port on an HCA is one.

reason – Optional text description of reason ports are being disabled, will be saved at the end of any new lines in the disabled file. For ports already in the disabled file, this is ignored.



`disable.csv` – File listing the links to disable. It is of the following form:

```
NodeGUID;PortNum;NodeType;NodeDesc;NodeGUID;PortNum;NodeType;NodeDesc;Reason
```

For each listed link, the switch port with the lower LID (closer to the SM) will be disabled. The `Reason` field is optional. The `Reason` field and any additional fields provided will be saved in the disabled file. An input file such as this can be generated by `iba_extract_bad_links` or `iba_extract_sel_links`.

4.5.30.3 Environment

The following environment variables are also used by this command:

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

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

4.5.30.4 Examples

```
iba_disable_ports 'bad cable' < disable.csv
```

```
iba_disable_ports -p '1:1 1:2 2:1 2:2' 'dead servers' < disable.csv
```

4.5.31 iba_enable_ports

(Linux) Accepts a disabled ports input file and re-enables the specified ports. The input file can be `/etc/sysconfig/iba/disabled*.csv` or a user-created subset of such a file. After enabling the port, it is removed from `/etc/sysconfig/iba/disabled*.csv`.

4.5.31.1 Usage

```
iba_enable_ports [-t portsfile] [-p ports] < disabled.csv
```

OR

```
iba_enable_ports --help
```

4.5.31.2 Options

`--help` – Produce full help text

`-t portsfile` – File with list of local HCA ports used to access fabric(s) for operation, default is `/etc/sysconfig/iba/ports`

`-p ports` – List of local HCA ports used to access fabric(s) for analysis default is first active port.

This is specified as `hca:port`

`0:0` – First active port in system

`0:y` – Port `y` within system

`x:0` – First active port on HCA `x`

`x:y` – HCA `x`, port `y`

The first HCA in the system is one. The first port on an HCA is one.

`disable.csv` – File listing the ports to enable. It is of the following form:

```
NodeGUID;PortNum;NodeDesc
```

A input file such as this is generated in `/etc/sysconfig/iba/disabled*`



4.5.31.3 Examples

```
iba_enable_ports < disabled.csv  
iba_enable_ports -p '1:1 1:2 2:1 2:2' < disabled.csv
```

4.5.31.4 Environment

The following environment variables are also used by this command:

PORTS – List of ports, used in absence of *-t* and *-p*

PORTS_FILE – File containing list of ports, used in absence of *-t* and *-p*

4.5.32 iba_disable_hosts

(Linux) Searches for a set of hosts in the fabric and disables their corresponding switch port.

4.5.32.1 Usage

```
iba_disable_hosts [-h hca] [-p port] reason host ...
```

OR

```
iba_disable_hosts --help
```

4.5.32.2 Options

--help – Produce full help text

-h hca – HCA to send through. The default is the first HCA

-p ports – Port to send through. The default is the first active port.

reason – Text description of reason hosts are being disabled, will be saved at end of any new lines in disabled file. For ports already in disabled file, this is ignored.

4.5.32.3 Examples

```
iba_disable_hosts 'bad DRAM' compute001 compute045  
iba_disable_hosts -h 1 -p 2 'crashed' compute001 compute045
```

4.5.33 iba_extract_lids

(Linux) Supporting tool that generates a csv file listing the map of LIDs that are currently present in the fabric.

4.5.33.1 Usage

```
iba_extract_lids [-h hca] [-p port]
```

OR

```
iba_extract_lids --help
```

4.5.33.2 Options

--help – Produce full help text



-h *hca* – HCA to send through. The default is the first HCA

-p *ports* – Port to send through. The default is the first active port.

4.5.33.3 Examples

```
iba_extract_lids > lids.csv
iba_extract_lids -h 2 -p 1 > lids.csv
```

4.6 Advanced Chassis Initialization and Verification

4.6.1 iba_chassis_admin

(Switch) `iba_chassis_admin` performs a number of multi-step operations. In general operations performed by `iba_chassis_admin` involve a login to one or more Intel® Chassis. `iba_chassis_admin` can perform initial chassis setup, firmware upgrades, reboot chassis and other operations.

4.6.1.1 Usage

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

or

```
iba_chassis_admin --help
```

4.6.1.2 Options

--help – Produce full help text

-c – Clobber result files from any previous run before starting this run

-F *chassisfile* – File with chassis in cluster. The default is /etc/sysconfig/iba/chassis

-H *chassis* – List of chassis to execute the operation against

-P *packages* – Filenames/directories of firmware images to install. For directories specified, all .pkg files in directory tree will be used. shell wild cards may also be used within quotes, or for fmconfig, filename of FM config file to use, or for fmgetconfig, filename to upload to (default ifs_fm.xml)

-a *action* – Action for supplied file

For chassis upgrade:

- push – Ensure firmware is in primary or alternate
- select – Ensure firmware is in primary
- run – Ensure firmware is in primary and running

The default is push.

For chassis fmconfig:

- push – Ensure config file is in chassis
- run – After push restart FM on master, stop on slave
- runall – After push restart FM on all MM

For chassis fmcontrol:

- stop – Stop FM on all MM
- run – Make sure FM running on master, stopped on slave



runall – Make sure FM running on all MM
restart – Restart FM on master, stop on slave
restartall – Restart FM on all MM

-I *fm_bootstate* – fmconfig and fmcontrol install options
 disable – Disable FM start at chassis boot
 enable – Enable FM start on master at chassis boot
 enableall – Enable FM start on all MM at chassis boot

-d *upload_dir* – Directory to upload FM config files to, default is uploads

-S – Securely prompt for password for user on chassis

operation – Operation to perform.
 Can be one or more of:
 reboot – Reboot chassis, ensure they go down and come back
 configure – Run wizard to perform chassis configuration
 upgrade – Upgrade install of all chassis
 getconfig – Get basic configuration of chassis
 fmconfig – FM config operation on all chassis
 fmgetconfig – Fetch FM config from all chassis
 fmcontrol – Control FM on all chassis

4.6.1.3 Example

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

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

- test.res – Appended with summary results of run
- test.log – Appended with detailed results of run
- save_tmp/ – Contains a directory per failed test with detailed logs
- test_tmp*/ – Intermediate result files while test is running

The -c option will remove all of the above.

When performing operations against chassis, set up of ssh keys is recommended (see [“setup_ssh” on page 159](#)). If ssh keys are not setup, all chassis must be configured with the same admin password and use of the -S option is recommended. The -S option avoids the need to keep the password in configuration files.

Results from *iba_chassis_admin* are grouped into Test Suites, Test Cases and Test Items. A given run of *iba_chassis_admin* represents a single Test Suite. Within a Test Suite multiple Test Cases will occur, typically one Test Case per chassis being operated on. Some of the more complex operations may have multiple Test Items per Test Case. Each such item represents a major step in the overall Test Case.

Each *iba_chassis_admin* run appends to *test.res* and *test.log*, and creates temporary files in *test_tmp\$PID* in the current directory. *test.res* will provide an overall summary of operations performed and their results. The same information will also be displayed while *iba_chassis_admin* is executing. *test.log* will contain



detailed information about what was performed. This will include the specific commands executed and the resulting output. The `test_tmp` directories will contain temporary files which reflect tests in progress (or killed). The logs for any failures will be logged in the `save_temp` directory with a directory per failed test case. If the same test case fails more than once, `save_temp` will retain the information from the first failure and subsequent runs of `iba_chassis_admin` will only append to `test.log`. It is recommended to review failures and use the `-c` option to remove old logs before subsequent runs of `iba_chassis_admin`.

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

4.6.1.4 Environment Variables

The following environment variables are also used by this command:

`CHASSIS` – List of chassis, used if `-H` and `-F` option not supplied. Refer to [“Selection of Chassis” on page 24](#) for more information.

`CHASSIS_FILE` – File containing list of chassis, used in absence of `-F` and `-H`. Refer to [“Selection of Chassis” on page 24](#) for more information.

`FF_MAX_PARALLEL` – Maximum concurrent operations.

`FF_TIMEOUT_MULT` – Multiplier for response time-outs. Default is 2. This typically does not need to be set, but in the event of unexpected time-outs or extremely slow hosts or chassis or management network, a larger value can be used.

`FF_CHASSIS_LOGIN_METHOD` – How to login to chassis. Can be SSH or Telnet

`FF_CHASSIS_ADMIN_PASSWORD` – Password for admin on all chassis. Used in absence of `-S` option.

`FF_PACKAGES` – Directories or `.pkg` files to load during "upgrade". Used In absence of `-P` option.

`UPLOADS_DIR` – Directory to upload to, used in absence of `-d`.

4.6.2 iba_chassis_admin Chassis Operations

(Switch) All chassis operations will login to the chassis as chassis user admin. It is recommended to use the `-s` option to securely prompt for a password, in which case the same password is used for all chassis. Alternately, the password may be put in the environment or the `fastfabric.conf` file using `FF_CHASSIS_ADMIN_PASSWORD`.

Note: All versions of Intel® 12000 Chassis firmware permit SSH keys to be configured within the chassis for secure password-less login. In this case there is no need to configure a `FF_CHASSIS_ADMIN_PASSWORD` and `FF_CHASSIS_LOGIN_METHOD` can be SSH. Refer to [“setup_ssh” on page 159](#), or the *Intel® True Scale Fabric Switches 12000 Series CLI Reference Guide* for more information.

4.6.2.1 upgrade

This upgrades the firmware on each chassis or slot specified. The `-P` option selects a directory containing `.pkg` files or provides an explicit list of `.pkg` files for the chassis and/or slots. The `-a` option selects the desired minimal state for the new firmware. For each chassis and/or slot selected for upgrade, the `.pkg` file applicable to that slot will be selected and used. If more than one `.pkg` file is specified of a given card type, the operation is undefined.



The upgrade is intelligent and does not upgrade chassis that already have the desired firmware in the desired state (as specified by `-a`).

When the `-a` option specifies `run`, chassis that are not already running the desired firmware will be rebooted. By selecting the proper `FF_MAX_PARALLEL` value, a rolling upgrade or a parallel upgrade may be accomplished. In most cases a parallel upgrade is recommended for expediency.

For more information about chassis firmware refer to the *Intel® True Scale Fabric Switches 12000 Series Users Guide*, *Intel® True Scale Fabric Switches 12000 Series Release Notes*, *Intel® 9000 Users Guide* and *Intel® 9000 Release Notes*.

4.6.2.2 **configure**

This runs the chassis setup wizard, which asks the user a series of questions. Once the wizard has finished prompting for configuration information, all the selected chassis are configured through the CLI interface according to the responses. The following may be configured for all chassis:

- syslog server IP address, TCP/UDP port number, syslog facility code and the chassis LogMode
- NTP server
- local timezone
- maximum packet MTU
- VL Capability
- VL credit distribution
- link width supported
- IB Node Description
- IB Node Description format
- disable chassis auto clear of port counters

Note: In a fabric where FastFabric tools such as `iba_rfm` and `iba_top` that work in conjunction with the PM/PA to monitor port counters, it is required to disable the chassis port counter auto-clear feature.

4.6.2.3 **reboot**

This reboots the given chassis and ensures they go down and come back up by pinging them during the reboot process.

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

4.6.2.4 **getconfig**

This retrieves basic information from a chassis such as syslog, NTP configuration, timezone info, MTU Capability, VL Capability, VL Credit Distribution, Link Width and node description.

This operation also outputs a summary of various configuration settings for each switch within a fabric. For example, in a fabric with seven switches, a report similar to the following is displayed.

Summary:



```

count - configuration

1 - Auto clear status: Auto clear is enabled

1 - Firmware Active: 7.0.1.0.43

1 - Firmware Primary: 7.0.1.0.43

1 - LinkWidth Support: 4X

1 - MTU Capability: 2048 Bytes

1 - NTP: Configured to use NTP server: 10.32.2.3

1 - Product Family: 12000

1 - Syslog Configuration: Syslog host set to: 0.0.0.0 port 514 facility 22

1 - time zone: Current time zone offset is: -5

1 - VL Capability: 4 VLs

1 - VL Credit Distribution: 4

```

4.6.2.5 fmconfig

This updates the FM config file on each chassis specified. The `-P` option selects a file to transfer to the chassis. The `-a` option selects the desired minimal state for the new configuration and will control if the FM is started/restarted after the file is updated.

The `-I` option can be used to configure the FM start at boot for the selected chassis.

4.6.2.6 fmgetconfig

This uploads the FM config file from all selected chassis. The file is uploaded to the selected uploads directory. The `-P` option can specify the desired destination filename within the uploads directory.

4.6.2.7 fmcontrol

This allows the FM to be controlled on each chassis specified. The `-a` option selects the desired state for the FM.

The `-I` option can be used to configure the FM start at boot for the selected chassis.

4.7 Externally Managed Switch Initialization and Verification

4.7.1 iba_switch_admin

(Switch) `iba_switch_admin` performs a number of multi-step operations, against one or more externally managed Intel® Switches. `iba_switch_admin` can perform firmware upgrades, reboot switches as well as perform a variety of other operations.

4.7.1.1 Usage

```

iba_switch_admin [-c] [-N 'nodes'] [-L 'nodeFile'] [-d upload_dir] [-S] [-s] [-P
packages] [-a action] [-O override] [-t portsfile] [-p ports] operation ...

```

or

```

iba_switch_admin --help

```



4.7.1.2 Options

- help – Produce full help text
- c – Clobber result files from any previous run before starting this run
- N *nodes* – List of nodes to execute the command
- L *nodefile* – File with nodes in cluster. The default is `/etc/sysconfig/iba/ibnodes`
- d *upload_dir* – Directory to upload capture files to (default is uploads)
- S – Securely prompt for password for user on remote system/chassis test – Test to run.
- s – Securely prompt for new password for switch – Valid only with upgrade operation.
- P *packages* – Filename/directory of firmware image to install. For the directory specified, `.emfw` files in the directory tree will be used. `shell` wild cards may also be used within quotes.
- a *action* – For firmware file for chassis upgrade. The *action* argument can be one or more of the following:
 - select – Ensure firmware is in primary
 - run – Ensure firmware is in primary and runningThe default is `select`.
- O *override* – For firmware upgrades, bypass the previous firmware version checks, and force the update. The *operation* argument can be one or more of the following:
 - reboot – Reboot switches, ensure they go down and come back
 - configure – Run wizard to setup switch node configuration
 - upgrade – Upgrade install of all switches.
 - info – Report firmware and hardware version, part number and capabilities of all switches.
 - hwvpd – Complete Vital Product Data (VPD) report of all switches.
 - ping – Ping all switches – Test for presence
 - fwverify – Report integrity of firmware of all nodes.
 - capture – Captures switch hardware and firmware state(s) of all nodes.
 - getconfig – Get port configurations of a externally managed switch

4.7.1.3 Example

```
iba_switch_admin -c reboot
```

```
iba_switch_admin -a run -P '*.emfw' upgrade
```

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

- `test.res` – Appended with summary results of run
 - `test.log` – Appended with detailed results of run
 - `save_tmp/` – Contains a directory per failed test with detailed logs
 - `test_tmp*/` – Intermediate result files while test is running
- The `-c` option will remove all of the above.



Results from `iba_switch_admin` are grouped into Test Suites, Test Cases and Test Items. A given run of `iba_switch_admin` represents a single Test Suite. Within a Test Suite multiple Test Cases will occur, typically one Test Case per being operated on. Some of the more complex operations may have multiple Test Items per Test Case. Each such item represents a major step in the overall Test Case.

Each `iba_switch_admin` run appends to `test.res` and `test.log` and creates temporary files in `test_tmp$PID` in the current directory. `test.res` will provide an overall summary of operations performed and their results. The same information will also be displayed while `iba_switch_admin` is executing. `test.log` will contain detailed information about what was performed. This will include the specific commands executed and the resulting output. The `test_tmp` directories will contain temporary files which reflect tests in progress (or killed). The logs for any failures will be logged in the `save_temp` directory with a directory per failed test case. If the same test case fails more than once, `save_temp` will retain the information from the first failure and subsequent runs of `iba_switch_admin` will only append to `test.log`. It is recommended to review failures and use the `-c` option to remove old logs before subsequent runs of `iba_switch_admin`. `iba_switch_admin` also appends to `punchlist.csv` for failing switches.

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

4.7.1.4 Environment Variables

The following environment variables are also used by this command:

`IBNODES`, `IBNODES_FILE` – See discussion on “Selection of Switches” on page 26.

`FF_MAX_PARALLEL` – Maximum concurrent operations.

`FF_TIMEOUT_MULT` – Multiplier for response time-outs. Default is 2. This typically does not need to be set, but in the event of unexpected time-outs or extremely slow hosts or chassis or management network, a larger value can be used.

`FF_PACKAGES` – Directories or `.emfw` files to load during switch “upgrade”. Used in absence of `-P` option.

4.7.2 iba_switch_admin Operations

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

4.7.2.1 reboot

Reboots the given switches.

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



4.7.2.2 upgrade

Upgrades the firmware on each specified switch. The `-P` option selects a directory containing a `.emfw` file or provides an explicit `.emfw` file for the switches. If more than one `.emfw` file is specified, the operation is undefined. The `-a` option selects the desired minimal state for the new firmware. Only the `select` and `run` options are valid for this operation.

When the `-a` option specifies `run`, switches will be rebooted. By selecting the proper `FF_MAX_PARALLEL` value a rolling upgrade or a parallel upgrade may be accomplished. In most cases a parallel upgrade is recommended for expediency.

The upgrade process will also set the switch name. See discussion on “[Selection of Switches](#)” on [page 26](#) above.

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

For more information about switch firmware refer to the *Intel® True Scale Fabric Switches 12000 Series Users Guide*, *Intel® True Scale Fabric Switches 12000 Series Release Notes*, *Intel® 9000 Users Guide* and *Intel® 9000 Release Notes*.

4.7.2.3 configure

This runs the switch setup wizard, which asks the user a series of questions. Once the wizard has finished prompting for configuration information, all the selected switches are configured according to the responses to the questions. The following items are configurable for all Intel® 12000 Switches:

- MTU
- VL Capability
- VL credit distribution
- Link Width Supported
- IB Node Description

Note: If 4X capability is not enabled in the user selection (for example, selecting 8X only, or selecting 1X/8X), 4X capability is added to port 1 for each switch being configured. This is so that it is always possible to “rescue” the switch with FastFabric (by connecting to it with 4X) should the link be unable to connect to with a link width other than 4X.

Note: Normally, the IB Node Description is updated automatically as part of a firmware upgrade, if it is configured properly in the `ibnodes` file. Update of the node description is also available with the `configure` option without the need for a firmware upgrade.

4.7.2.4 info

Queries the switches and displays the following information:

- Firmware version
- Hardware version
- Hardware part number, including revision information
- Speed capability (SDR, DDR, QDR)
- Fan Status



- Power Supply Status

This operation also outputs a summary of various configuration settings for each switch within a fabric. For example, in a fabric with seven switches, a report similar to the following is displayed.

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

4.7.2.5 hwwpd

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

- Serial Number
- Part Number
- Model Name
- Hardware Version
- Manufacturer
- Product description
- Manufacturer ID
- Manufacture date
- Manufacture time

4.7.2.6 ping

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

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

4.7.2.7 fwverify

Verifies the integrity of the firmware images in the eeproms of the selected switches.

4.7.2.8 capture

Get switch hardware and firmware state capture of all nodes.



4.7.2.9 getconfig

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

Summary:

```
count - configuration
    7 - Link Speed : 2.5-10Gb
    1 - Link Width : 1-8x
    6 - Link Width : 4x
    7 - MTU : 2048
    7 - VL Capability : 4+1
    1 - VL Credit Distribution Method : 0
    6 - VL Credit Distribution Method : 4
```

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

4.8 Advanced Host Initialization and Verification

4.8.1 iba_host_admin

(Host) `iba_host_admin` performs a number of multi-step operations. In general operations performed by `iba_host_admin` involve a login to one or more host systems. `iba_host_admin` can perform software or firmware upgrades, reboot hosts, as well as perform a variety of host and fabric verification operations.

4.8.1.1 Usage

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

or

```
iba_host_admin --help
```

4.8.1.2 Options

`--help` – Produce full help text

`-c` – Clobber result files from any previous run before starting this run

`-i ipoib_suffix` – Suffix to apply to host names to create ipoib host names. The default is `-ib`.

`-f hostfile` – File with hosts in cluster, default is `/etc/sysconfig/iba/hosts`

`-h HOSTS` – List of hosts to execute the command



-r *release* – IntelIB or InfiniServ release to load/upgrade to, default is version of Intel® True Scale Fabric Suite presently being run on the server running this command.

-d *dir* – Directory to get product release.tgz from for load/upgrade

-I *install_options* – InfiniServ install options

-U *upgrade_options* – InfiniServ upgrade options

-T *product* – InfiniServ product type to install, default is IntelIB-Basic. Other options include: InfiniServBasic, InfiniServPerf, InfiniServMgmt, InfiniServTools.

-P *packages* – InfiniServ packages to install; default is *iba*, *ipoib*, and *mpi*. The host allows: *ib_stack*, *oftools*, *ib_stack_dev*, *fastfabric*, *ofed_ipoib*, *ofed_srp*, *ofed_srpt*, *ofed_iser*, *ofed_iwarp*, *opensm*, *ofed_debug*, *iba_ibdev*, *ibboot*, *fastfabric*, *ifibre*, *inic*, *ipoib*, *mpi*, *mpidev*, *mpisrc*, *udapl*, *sdg*, and *rds*.

-m *netmask* – IPoIB netmask to use for configipoib

-S – Securely prompt for password for user on remote system

operation – Operation to perform. The *operation* argument can be one or more of the following:

load – Initial install of all hosts

upgrade – Upgrade install of all hosts

configipoib – Create ifcfg-ib1 using host IP address from /etc/hosts

reboot – Reboot hosts, ensure they go down and come back

sacache – Confirm sacache has all hosts in it

ipoibping – Verify this host can ping each host through IPoIB

mpiperf – Verify latency and bandwidth for each host

mpiperfdeviation – Verify latency and bandwidth for each host against a defined threshold (or relative to average host performance)

4.8.1.3 Example

```
iba_host_admin -c reboot
```

```
iba_host_admin -h 'elrond arwen' reboot
```

```
HOSTS='elrond arwen' iba_host_admin reboot
```

4.8.1.4 Details

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

test.res – Appended with summary results of run

test.log – Appended with detailed results of run

save_tmp/ – Contains a directory per failed test with detailed logs

test_tmp*/ – Intermediate result files while test is running

The -c option will remove all of the above.

Results from *iba_host_admin* are grouped into Test Suites, Test Cases and Test Items. A given run of *iba_host_admin* represents a single Test Suite. Within a Test Suite multiple Test Cases will occur, typically one Test Case per host being operated on. Some of the more complex operations (such as *ipoibping*) may have multiple Test Items per Test Case. Each such item represents a major step in the overall Test Case.



Each `iba_host_admin` run appends to `test.res` and `test.log` and creates temporary files in `test_tmp$PID` in the current directory. `test.res` will provide an overall summary of operations performed and their results. The same information will also be displayed while `iba_host_admin` is executing. `test.log` will contain detailed information about what was performed. This will include the specific commands executed and the resulting output. The `test_tmp` directories will contain temporary files which reflect tests in progress (or killed). The logs for any failures will be logged in the `save_temp` directory with a directory per failed test case. If the same test case fails more than once, `save_temp` will retain the information from the first failure and subsequent runs of `iba_host_admin` will only append to `test.log`. It is recommended to review failures and use the `-c` option to remove old logs before subsequent runs of `iba_host_admin`.

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

4.8.1.5 Environment Variables

The following environment variables are also used by this command:

`HOSTS`, `HOSTS_FILE` – See discussion on “[Selection of Hosts](#)” on page 23

`FF_IPOIB_SUFFIX` – Suffix to append to hostname to create IPoIB hostname. Used in absence of `-i`

`FF_MAX_PARALLEL` – Maximum concurrent operations will be performed.

`FF_USERNAME` – User name to login to hosts as, default is `root`

`FF_PASSWORD` – Password to use to login as `FF_USERNAME`. Used in absence of `-S` option.

`FF_ROOTPASS` – Password to use when `su` to `root` (if `FF_USERNAME` is not `root`). Used in absence of `-S` option.

`FF_LOGIN_METHOD` – How to login to hosts (Telnet, RSH or SSH), default is SSH

`FF_TIMEOUT_MULT` – Multiplier for response time-outs. Default is 2. This typically does not need to be set, but in the event of unexpected time-outs or extremely slow hosts or chassis or management network, a larger value can be used.

`FF_PRODUCT` – During host install and upgrade, what product should be used for installation (IntelIB-Basic, InfiniServPerf, InfiniServBasic, etc)

`FF_INSTALL_OPTIONS` – Installation options for host software `INSTALL` during host “load”. Used in absence of `-I` option.

`FF_UPGRADE_OPTIONS` – Upgrade options for host software `INSTALL` during host “upgrade”. Used in absence of `-U` option.

`FF_PACKAGES` – Host packages to load during host “load”. Used in absence of `-P` option.

`FF_IPOIB_NETMASK` – Netmask to use for IPoIB IP address during `configipoib`

`FF_IPOIB_CONFIG` – Assists in the configuration of the IPoIB interface

`FF_DEVIATION_ARGS` – Arguments to `/opt/iba/src/mpi_apps/deviation/deviation` application to use during `mpiperfdeviation`



4.8.2 iba_host_admin Host Operations

(Host) It is recommended to set up password SSH or SCP for use during this operation. Alternatively, the `-s` option can be used to securely prompt for a password, in which case the same password is used for all hosts. Alternately, the password may be put in the environment or the `fastfabric.conf` file using `FF_PASSWORD` and `FF_ROOTPASS`.

4.8.2.1 load

This performs an initial installation of Fabric Access software on a group of hosts. Any existing Fabric Access installation will first be uninstalled and any Fabric Access configuration files will be removed. Therefore, the hosts will end up installed with a default Fabric Access configuration. The `-I` option can be used to select different install packages, the defaults are `iba`, `ipoib`, and `mpi` (for example, True Scale Fabric Stack, IPoIB and MPI). The default is the typical configuration for an MPI cluster compute node. The `-r` option can be used to specify a release to install other than the one that this host is presently running. The `FF_PRODUCT.FF_PRODUCT_VERSION.tgz` file (for example, `IntelIB-Basic.version.tgz`) is expected to exist in the directory specified by `-d` (the default is the current working directory) and will be copied to all the selected hosts and installed.

Note: When using the present version of Intel® FastFabric Toolset for OFED+, Intel® FastFabric Toolset may be used to install Intel® True Scale Fabric OFED+ Host Software (`IntelIB-Basic.version.tgz`) or the True Scale Fabric Stack Tools (`InfiniServTools.version.tgz`) on the remaining hosts. When using Intel® FastFabric Toolset only to install InfiniServ Tools, OFED must be installed on each host manually.

4.8.2.2 upgrade

This is very similar to the `load` option, however all the selected hosts are upgraded without affecting existing Fabric Access configuration. This is comparable to the `-U` option when running `INSTALL` manually. The `-r` option can be used to upgrade to a release different from this host, the default will be to upgrade to the same release as the this host. The `FF_PRODUCT.FF_PRODUCT_VERSION.tgz` file (for example, `IntelIB-Basic.version.tgz`) is expected to exist in the directory specified by `-d` (the default is the current working directory) and will be copied to all the end nodes and installed.

Note: Only those Fabric Access components that are currently installed will be upgraded. This operation will fail for hosts that do not have Fabric Access software installed.

Note: When using the present version of Intel® FastFabric Toolset for OFED+, Intel® FastFabric Toolset may be used to install Intel® True Scale Fabric OFED+ Host Software (`IntelIB-Basic.version.tgz`) or the True Scale Fabric Stack Tools (`InfiniServTools.version.tgz`) on the remaining hosts. When using Intel® FastFabric Toolset only to install InfiniServ Tools, OFED must be installed on each host manually.

4.8.2.3 configipoib

Creates a `ifcfg-ib1` configuration file (when running OFED+, this configures `ifcfg-ib0`) for each node using the IP address found using the resolver on the node (the standard Linux resolver is used through the `host` command). If the host is not found, `/etc/hosts` on the node is checked. The `-i` option can specify an IPoIB suffix to apply to the host name to create the IPoIB host name for the node (that will be looked up in `/etc/hosts`). The default suffix is `-ib`. The `-m` option can be used to specify a netmask other than the default for the given class of IP address (such as



when dividing a class A or B address into smaller IP subnets). IPoIB will be configured for a static IP address and will be autostarted at boot. For the Intel® True Scale Fabric Stack, the default `/etc/sysconfig/ipoib.cfg` file will be used, which provides a redundant IPoIB configuration using both ports of the first HCA in the system.

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

4.8.2.4 reboot

This reboots the given hosts and ensures they go down and come back up by pinging them during the reboot process. The ping rate is slow (5 seconds), so if the servers boot faster than this, false failures may be seen.

4.8.2.5 sacache

This verifies the given hosts can properly communicate with the SA and any cached SA data that is up to date. To run this command, True Scale Fabric software must be installed and running on the given hosts. The subnet manager and switches must be up. If this test fails, for QuickSilver hosts: `cmdall 'cat /proc/driver/ics_dsc/gids'` can be run against any problem hosts to see what they have cached. If this test fails, for OFED hosts: `cmdall 'iba_saquery -o desc'` can be run against any problem hosts to see what they see.

Note: This operation requires that the hosts being queried be specified by a resolvable TCP/IP host name. This operation will FAIL if the selected hosts are specified by IP address. See [“Selection of Hosts” on page 23](#) for more information.

4.8.2.6 ipoibping

This verifies IPoIB basic operation by ensuring that the host can ping all other nodes through IPoIB. To run this command True Scale Fabric software must be installed, IPoIB must be configured and running on the host and the given hosts, the SM and switches must be up. The `-i` option can specify an alternate IPoIB hostname suffix.

4.8.2.7 mpipeperf

Verifies that MPI is operational and checks MPI end-to-end latency and bandwidth between pairs of nodes (for example, 1-2, 3-4, 5-6). This can be used to verify switch latency/hops, PCI bandwidth and overall MPI performance. The `test.res` file will have the results of each pair of nodes tested.

Note: This option is available for the IntelIB packaging of OFED, but is not presently available for other packagings of OFED.

To obtain accurate results, this test should be run at a time when no other stressful applications (for example, MPI jobs, high stress file system operations) are running on the given hosts.

The following is a sample of expected MPI bandwidths for various server slot speeds:

- PCI-X 66 MHz (32 bit) - 250 MB/s or less
- PCI-X 66MHz - 400-450 MB/s or less
- PCI-X 100 MHz - 600-700 MB/s



- PCI-X 133 MHz - 800-900 MB/s
- PCIe x8 SDR HCA - 900+ MB/s
- PCIe x8 DDR HCA - 1300+ MB/s
- PCIe Gen 2 x8 QDR HCA - 2400+ MB/s

Bandwidth issues typically indicate server configuration issues (for example, incorrect slot used, incorrect BIOS settings, incorrect HCA model), or fabric issues (for example, symbol errors, incorrect link width or speed). Assuming `iba_report` has previously been used to check for link errors and link speed issues, the server configuration should be verified.

Note that BIOS settings and differences between server models can account for 10-20% differences in bandwidth. The numbers above are conservative numbers representative of what most servers can achieve. Some server models may have 10-20% higher results. A result 5-10% below the above numbers is typically not cause for serious alarm, but may reflect limitations in the server design or the chosen BIOS settings.

For more details about BIOS settings, consult the documentation from the server supplier and/or the server PCI chipset manufacturer.

4.8.2.8 **mpiperfdeviation**

Is an upgraded version of `mpiperf`. It verifies that MPI is operational and checks MPI end-to-end latency and bandwidth between pairs of nodes. This can be used to verify switch latency/hops, PCI bandwidth and overall MPI performance. It performs assorted pairwise bandwidth and latency tests and report pairs outside an acceptable tolerance range. The tool will identify specific nodes which have problems and provide a concise summary of results. The `test.res` file will have the results of each pair of nodes tested.

By default concurrent mode is used to quickly analyze the fabric and host performance. Pairs which have 20% less bandwidth or 50% more latency than the average pair will be reported as failures.

The tool can be run in a sequential or a concurrent mode. Sequential mode will run each host against a reference host. By default the reference host is selected based on the best performance from a quick test of the first 40 hosts.

In concurrent mode, hosts are paired up and all pairs are run concurrently. Since there may be fabric contention during such as run, any poor performing pairs are then rerun sequentially against the reference host.

Concurrent mode will run the tests in the shortest amount of time, however the results could be slightly less accurate due to switch contention. In heavily oversubscribed fabric designs, if concurrent mode is producing unexpectedly low performance, try sequential mode.

Note: This option is available for the IntelIB packaging of OFED, but is not presently available for other packagings of OFED.

To obtain accurate results, this test should be run at a time when no other stressful applications (for example, MPI jobs, high stress file system operations) are running on the given hosts.



Bandwidth issues typically indicate server configuration issues (for example, incorrect slot used, incorrect BIOS settings, incorrect HCA model), or fabric issues (for example, symbol errors, incorrect link width or speed). Assuming `iba_report` has previously been used to check for link errors and link speed issues, the server configuration should be verified.

Note that BIOS settings and differences between server models can account for 10-20 percent differences in bandwidth. A result 5-10 percent below the average is typically not cause for serious alarm, but may reflect limitations in the server design or the chosen BIOS settings.

For more details about BIOS settings, consult the documentation from the server supplier and/or the server PCI chipset manufacturer.

The deviation application supports a number of parameters which allow for more precise control over the mode, benchmark and pass/fail criteria. The parameters to use can be selected using the `FF_DEVIATION_ARGS` configuration parameter in `fastfabric.conf`

Available parameters to deviation application:

```
[-bwtol bwtol] [-bwdelta MBs] [-bwthres MBs] [-bwloop count] [-bwsiz size]
[-lattol latol] [-latdelta usec] [-latthres usec] [-latloop count] [-latsiz size]
[-c] [-b] [-v] [-vv] [-h reference_host]

-bwtol      Percent of bandwidth degradation allowed below Avg value
-bwbidir    Perform a bidirectional bandwidth test
-bwunidir   Perform a unidirectional bandwidth test (default)
-bwdelta    Limit in MB/s of bandwidth degradation allowed below Avg value
-bwthres    Lower Limit in MB/s of bandwidth allowed
-bwloop     Number of loops to execute each bandwidth test
-bwsiz      Size of message to use for bandwidth test
-lattol     Percent of latency degradation allowed above Avg value

-latdelta   Limit in usec of latency degradation allowed above Avg value
-latthres   Lower Limit in usec of latency allowed
-latloop    Number of loops to execute each latency test
-latsiz     Size of message to use for latency test
-c          Run test pairs concurrently instead of the default of sequential
-b          When comparing results against tolerance and delta use best
            instead of Avg
-v          verbose output
-vv         Very versbose output
-h          Reference host to use for sequential pairing
```



Both bwtol and bwdelta must be exceeded to fail bandwidth test

When bwthres is supplied, bwtol and bwdelta are ignored

Both lattol and latdelta must be exceeded to fail latency test

When latthres is supplied, lattol and latdelta are ignored

For consistency with OSU benchmarks MB/s is defined as 1000000 bytes/s

4.9 Interpreting the iba_host_admin, iba_chassis_admin and iba_switch_admin log files

Each run of iba_host_admin, iba_chassis_admin and iba_switch_admin will create test.log and test.res files in the current directory.

When iba_host_admin, iba_chassis_admin and iba_switch_admin indicates that some or all of the test cases failed, the test.res and test.log files should be reviewed. test.res will summarize which tests have failed. Using the test.res file for servers that failed can be quickly identified. If the problem is not immediately obvious, check the test.log file. The most recent results will be at the end of the file. The save_tmp/*/test.log files will be easier to read since they will represent the logs for a single test case, typically against a single chassis, switch or host.

The keyword FAILURE will be used to mark any failures. Typically due to the roll up of error messages, the first instance of FAILURE in a given sequence of failures will show what was being done. Proceeding the FAILURE, the log will also show the exact sequence of commands issued to the target host and/or chassis and the resulting output from that host and/or chassis.

For example, test.log may contain lines such as:

```
scp ./InfiniServPerf.4.1.1.0.15.tgz root@n001a:
```

```
TEST CASE FAILURE=scp ./InfiniServPerf.4.1.1.0.15.tgz root@n001a: failed: ssh:
n001a Name or service not known
```

```
lost connection
```

This indicates the scp command shown was executed but failed with the error message:

```
"ssh: n001a Name or service not known.
```

```
lost connection"
```

In this example, this was the exact output from SSH.

If there is a FAILURE message indicating time-out, it means the expected output did not occur within a reasonable time limit. The time limits used are quite generous, so such failures often indicate a host, chassis or switch is offline. It could also indicate unexpected prompts (such as a password prompt when password-less ssh is expected). Review the test.log first for such prompts. Also verify that the host can SSH to the target host or chassis with the expected password behavior.



Another common source of time-outs is incorrect host shell command prompts. Verify that both this host and the target host have their prompts set correctly. The command line prompt must end in # or \$ (make certain there is a space after either).

Yet another common source of time-outs is typographical errors in selected host or chassis names. Verify that the host, chassis or switch names in `test.log` match the intended host names. Also make sure that when IPoIB host names are used, that the correct name was formed based on the `iba_host_admin -i '<IPoIB SUFFIX>'` argument. This applies a suffix to host names to create IPoIB host names. The default is `-ib`. Use `-i ''` to indicate no suffix.

4.10 Health Check and Baselining Tools

(All) These tools help to rapidly identify if the fabric has a problem or if its configuration has changed since the last baseline. Analysis includes hardware, software, fabric topology and SM configuration. The tools are designed to permit easy manual execution or automated execution using `cron` or other mechanisms.

These tools consist of 5 commands:

`all_analysis` – Performs selected set of the below 4 analysis commands. This command is recommended as the primary tool for general analysis.

When its desired to restrict the analysis to a specific subset of components, use one of the commands below.

`fabric_analysis` – Performs fabric topology and PMA error counters analysis.

`chassis_analysis` – Performs Intel® Chassis configuration and health analysis for selected chassis.

`esm_analysis` – Performs embedded SM configuration and health analysis for selected chassis.

`hostsm_analysis` – Performs host SM configuration and health analysis for the local host.

4.10.1 Usage Model

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

- Perform initial fabric install and verification
 - Optionally run tools in “health check only” mode
 - Performs quick health check
 - Duplicates some of steps already done during verification
- Run tools in “baseline” mode
 - Takes a baseline of present HW/SW/config
- Periodically run tools in “check” mode
 - Performs quick health check
 - Compares present HW/SW/config to baseline
 - Can be scheduled in hourly cron jobs
- As needed rerun “baseline” when expected changes occur
 - Fabric upgrades



- Hardware replacements/changes
- Software configuration changes
- Etc.

4.10.2 Common Operations and Options

The Health Check and Baselining tools support the following options to select the operations to be done by the tool:

- b – Perform a baseline snapshot of the configuration
- e – Perform an error check/health analysis only

If neither option is specified, the tool performs a snapshot of the present configuration, compares it to the baseline and also perform an error check/health analysis.

Use of both -b and -e on a given run is not permitted.

The typical use of the tools is to perform an initial error check by running the -e option. Review the errors reported in the files indicated by the tools. Once all the errors are corrected, perform a baseline of the configuration using the -b option. The baseline configuration will be saved to files in `FF_ANALYSIS_DIR/baseline` (the default of `/var/opt/iba/analysis/baseline` is set through `/etc/sysconfig/fastfabric.conf`). This baseline configuration should be carefully reviewed to make sure it matches the intended configuration of the cluster. If it does not, the cluster should be corrected and a new baseline run.

4.10.2.1 Example

```
fabric_analysis -e
```

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

```
fabric_analysis -b
```

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

Note: Alternatively, the advanced topology verification capabilities of `iba_report` can be used to verify the fabric deployment against the intended design

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

```
fabric_analysis
```

Will check the present fabric links and components against the previous baseline. If there have been changes, it will report a failure and indicate which files hold the resulting snapshot and differences. It will also check the PMA error counters and link speeds for the fabric (similar to `fabric_analysis -e`). If either of these checks fail, it will return a non-zero exit status, therefore permitting higher level scripts to detect a failed condition.



The differences files are generated using the Linux command specified by `FF_DIFF_CMD` in `fastfabric.conf`. By default this is the `diff -C 1` command. It is run against the baseline and new snapshot. Therefore, lines after each `*** #, #` heading in the `diff` are from the baseline and lines after each `--- #, #` heading are from the new snapshot. If `FF_DIFF_CMD` is simply set to `diff`, lines indicated by "<" in the `diff` would be from the baseline and lines indicated by ">" in the `diff` would be from the new snapshot. Another command which can be useful is the Linux `sdiff` command. For more information about the `diff` output format, consult the Linux man page for `diff`.

If the configuration is intentionally changed, a new error analysis and baseline should be obtained using the same sequence as for the initial installation (discussed above), establishing a new baseline for future comparisons.

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

`-s` – Save history of failures.

`-d dir` – Top level directory for saving baseline, snapshots and history (default is `FF_ANALYSIS_DIR` which is set in `fastfabric.conf`).

When the `-s` option is used, each failed run will also create a directory (whose name is the date/time the analysis tool was started) containing the failing snapshot information and `diffs`. This will permit a history of failures to be tracked. Note that every run of the tools also creates a `latest` directory with the latest snapshot. The `latest` files are overwritten by each subsequent run of the tool, which means the most recent run results are always available.

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

The `-d` option allows command line control over the baseline, snapshot and history directory tree. Runs using `-d` must use the same directory as any previous baseline which is to be compared to (except when `-e` option is used). The `FF_ANALYSIS_DIR` option in `fastfabric.conf` can be changed to provide a customer specific alternate directory which will be used whenever the `-d` option is not specified. Under `FF_ANALYSIS_DIR` subdirectories will be created as follows:

- `baseline` – Baseline snapshot from each analysis tool.
- `latest` – Latest snapshot from each analysis tool.
- `YYYY-MM-DD-HH:MM:SS` – Failed analysis from analysis run with `-s`. Actual directory name will have actual date/time as the name.

4.10.3 fabric_analysis

(All) The `fabric_analysis` command performs analysis of the fabric.

4.10.3.1 Usage

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

4.10.3.2 Options

- `-b` – Baseline mode, default is compare/check mode.
- `-e` – Evaluate health only, default is compare/check mode.
- `-s` – Save history of failures (errors/differences).



`-d dir` – Top level directory for saving baseline and history of failed checks. The default is `/var/opt/iba/analysis`.

`-c file` – Error thresholds config file. The default is `/etc/sysconfig/iba/iba_mon.conf`.

`-t portsfile` – File with list of local HCA ports used to access fabric(s) for analysis. The default is `/etc/sysconfig/iba/ports`.

`-p ports` – List of local HCA ports used to access fabric(s) for analysis. The default is the first active port.

This is specified as **HCA:port**:

`0:0` – first active port in system

`0:y` – port y within system

`x:0` – first active port on HCA x

`x:y` – HCA x, port y

`-T topology_input` – Name of topology input file to use. Any `%P` markers in this filename will be replaced with the `hca:port` being operated on (such as `0:0` or `1:2`). The default is `/etc/sysconfig/iba/topology.%P.xml`. If `-T NONE` is specified, no topology input file will be used. See [“iba_report” on page 175](#) for more information.

4.10.3.3 Example

```
fabric_analysis
```

```
fabric_analysis -p '1:1 1:2 2:1 2:2'
```

The fabric analysis tool checks the following:

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

Note that the comparison includes components on the fabric. Therefore operations such as shutting down a server will cause the server to no longer appear on the fabric and will be flagged as a fabric change or failure by `fabric_analysis`.

4.10.3.4 Environment Variables

The following environment variables are also used by this command:

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

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

`FF_TOPOLOGY_FILE` – File containing `topology_input` (may have `%P` marker in filename), used in absence of `-T`.

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

`FF_CURTIME` – Timestamp to use on directory created in `FF_ANALYSIS_DIR`, default is the present date and time.

`FF_FABRIC_HEALTH` – `iba_report` options to use during a health check.

`FF_DIFF_CMD` – Linux command used to compare the baseline to the latest snapshot.



4.10.3.5 Details

For simple fabrics, the Intel® FastFabric Toolset host would be connected to a single fabric. By default the first active port on the Intel® FastFabric Toolset host will be used to analyze the fabric.

However, in more complex fabrics, the Intel® FastFabric Toolset host may be connected to more than one fabric (or subnet). In this case the specific ports and/or HCAs to use for fabric analysis may be specified. The HCA and port number specified will become part of the filenames in the `FF_ANALYSIS_DIR` such that unique status can be tracked for each fabric.

Specification of the ports to be used can be performed on the command line using the `-p` option, in a file specified using the `-t` option, through the environment variables `PORTS` or `PORTS_FILE`, or using the `PORTS_FILE` configuration option in `fastfabric.conf`. If the specified file does not exist or is empty, the first active port on the local system will be used. In more complex configurations (such as where the Intel® FastFabric Toolset host is connected to multiple True Scale Fabrics or subnets), the user will need to specify the exact ports to use such that all fabrics are analyzed. For more information, refer to [“Selection of local Ports \(subnets\)” on page 28](#).

Specification of the `topology_input` file to be used can be performed on the command line using the `-T` option, in a file specified through the environment variable `FF_TOPOLOGY_FILE`, or using the `FF_TOPOLOGY_FILE` configuration option in `fastfabric.conf`. If the specified file does not exist, no `topology_input` file will be used. Alternately the filename can be specified as `NONE` to prevent use of a `topology_input` file. For more information, refer to [“iba_report” on page 175](#).

If specified, the `topology_input` file will be used to augment the information included in reports (see [“iba_report” on page 175](#) for more information).

By default the error analysis includes PMA counters and slow links (for example, links running below enabled speeds). However this can be changed using the `FF_FABRIC_HEALTH` configuration parameter in `fastfabric.conf` (see *Appendix A* in the *Intel® True Scale Fabric Suite FastFabric User Guide* for more information). This parameter specifies the `iba_report` options and reports to be used for the health analysis. It also can specify the PMA counter clearing behavior (`-i seconds`, `-C`, or none at all). When a `topology_input` file is used, it can also be useful to extend `FF_FABRIC_HEALTH` to include fabric topology verification options such as `-o verifylinks`.

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

All files generated by `fabric_analysis` will start with fabric in their file name. This is followed by the port selection option (default of `0:0`) identifying the port used for the analysis.

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

4.10.3.6 Health Check

`latest/fabric.0:0.errors` - stdout of `iba_report` for errors encountered during fabric error analysis.

`latest/fabric.0:0.errors.stderr` - stderr of `iba_report` during fabric error analysis.



4.10.3.7 Baseline

`baseline/fabric.0:0.snapshot.xml` - `iba_report` snapshot of complete fabric components and SMA configuration.

`baseline/fabric.0:0.comps` - `iba_report` summary of fabric components and basic SMA configuration.

`baseline/fabric.0:0.links` - `iba_report` summary of internal and external links.

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

4.10.3.8 Full analysis

`latest/fabric.0:0.snapshot.xml` - `iba_report` snapshot of complete fabric components and SMA configuration.

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

`latest/fabric.0:0.errors` - `stdout` of `iba_report` for errors encountered during fabric error analysis.

`latest/fabric.0:0.errors.stderr` - `stderr` of `iba_report` during fabric error analysis.

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

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

`latest/fabric.0:0.comps.diff` - `diff` of baseline and latest fabric components.

`latest/fabric.0:0.links` - `stdout` of `iba_report` summary of internal and external links.

`latest/fabric.0:0.links.stderr` - `stderr` of `iba_report` summary of internal and external links.

`latest/fabric.0:0.links.diff` - `diff` of baseline and latest fabric internal and external links.

`latest/fabric.0:0.links.changes.stderr` - `stderr` of `iba_report` comparison of links.

`latest/fabric.0:0.links.changes` - `iba_report` comparison of links against baseline, this is typically easier to read than the `links.diff` file and will contain the same information.

`latest/fabric.0:0.comps.changes.stderr` - `stderr` of `iba_report` comparison of components.

`latest/fabric.0:0.comps.changes` - `iba_report` comparison of components against baseline, this is typically easier to read than the `comps.diff` file and will contain the same information.

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



If the `-s` option is used and failures are detected, files related to the checks that failed are also copied to the timestamped directory name under `FF_ANALYSIS_DIR`, such as:

`FF_ANALYSIS_DIR/2007-11-22-09:53:04`

4.10.3.9 True Scale Fabric items checked against the baseline

- Based on `iba_report -o links`:
 - Unconnected/down/missing cables
 - Added/moved cables
 - Changes in link width and speed
 - Changes to Node GUIDs in fabric (replacement of HCA or Switch hardware)
 - Adding/Removing Nodes (CA, Virtual CAs, Virtual Switches, Physical Switches, Physical Switch internal switching cards (leaf/spine))
 - Changes to server or switch names
- Based on `iba_report -o comps`:
 - Overlap with items above from links report
 - Changes in port MTU, LMC, number of VLS
 - Changes in port speed/width enabled or supported
 - Changes in HCA or switch device IDs/revisions/VendorID (for example, ASIC HW changes)
 - Changes in port Capability mask (which features/agents run on port/server)
 - Changes to ErrorLimits and PKey enforcement per port
 - Changes to IOUs/IOCs/IOC Services provided
- Only applicable if IOUs in fabric (9000 series Virtual IO cards, native storage, etc)
- Location (port, node) and number of SMs in fabric
 - Includes primary and backups
 - Includes configured priority for SM

4.10.3.10 True Scale Fabric Items that are also checked during health check

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

- PMA error counters on all True Scale Fabric ports (HCA, switch external and switch internal) checked against configurable thresholds.
 - Counters are cleared each time a healthcheck is run, each healthcheck reflects a counter delta since last healthcheck.
 - Typically identifies potential fabric errors (symbol errors, etc).
 - May also identify transient congestion (depends upon counters monitored).
- Link active speed/width as compared to Enabled speed.
 - Identifies links whose active speed/width is < min (enabled speed/width on each side of link).
 - This typically reflects bad cables or bad ports or poor connections.
- Side effect is the verification of SA health.



4.10.4 chassis_analysis

(Switch) The `chassis_analysis` command performs analysis of the chassis.

4.10.4.1 Usage

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

4.10.4.2 Options

- b – Baseline mode. The default is the compare/check mode.
- e – Evaluate health only, default is the compare/check mode.
- s – A save history of failures (errors/differences).
- d *dir* – The top level directory for saving baseline and history of failed checks. The default is `/var/opt/iba/analysis`.
- F *chassisfile* – The file with the chassis in the cluster. The default is `/etc/sysconfig/iba/chassis`.
- H *chassis* – A list of chassis on which to execute the command.

4.10.4.3 Example

```
chassis_analysis
```

The chassis analysis tool checks the following for Intel® Chassis:

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

4.10.4.4 Environment Variables

The following environment variables are also used by this command:

`CHASSIS`, `CHASSIS_FILE` – See the discussion on the [“Selection of Chassis” on page 24](#) above.

`FF_TIMEOUT_MULT` – Multiplier for response time-outs. The default is 2. this typically does not need to be set, but in the event of unexpected time-outs or extremely slow chassis or management network, a larger value can be used.

`FF_CHASSIS_LOGIN_METHOD` – How to login to a chassis. Can be SSH or Telnet.

`FF_CHASSIS_ADMIN_PASSWORD` – The password for the administrator on all chassis.

`FF_CURTIME` – The timestamp to use on a directory created in `ff_analysis_dir`. The default is the present date and time.

`FF_CHASSIS_CMDS` – The set of chassis CLI commands to fetch chassis configuration information.

`FF_CHASSIS_HEALTH` – The chassis CLI command to check the chassis health.

`FF_DIFF_CMD` – Linux command used to compare the baseline to the latest snapshot.



4.10.4.5 Details

Setup of ssh keys for chassis (see “[setup_ssh](#)” on page 159) is recommended. If ssh keys are not setup, all chassis must be configured with the same admin password and the password must be kept in the fastfabric.conf configuration file.

The default set of FF_CHASSIS_CMDS is:

```
showInventory fwVersion showIBNodeDesc ismShowPStatThresh showInventory fwVersion  
showIBNodeDesc ismShowPStatThresh ismChassisSet12x timeZoneConf timeDSTConf  
snmpCommunityConf snmpTargetAddr showChassisIpAddr showDefaultRoute
```

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

FF_CHASSIS_HEALTH can specify one command (with arguments) to be used to check the chassis health. For chassis with newer firmware, the hwCheck command is recommended. For chassis with older firmware a benign command, such as fruInfo, should be used. The default is hwCheck. Note that only the exit status of the FF_CHASSIS_HEALTH command is checked. The output is not captured and compared in a snapshot. However, on failure its output is saved to aid diagnosis.

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

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

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

4.10.4.6 Health Check

latest/chassis.hwCheck – Output of hwCheck command for all selected chassis

4.10.4.7 Baseline

baseline/chassis.fwVersion – Output of fwVersion command for all selected chassis.

baseline/chassis.ismChassisSet12x – Output of the ismChassisSet12x command for all selected chassis.

baseline/chassis.ismShowPStatThresh – Output of the ismShowPStatThresh command for all selected chassis.

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

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

baseline/chassis.showIBNodeDesc – Output of the showIBNodeDesc command for all selected chassis.



`baseline/chassis.showInventory` – Output of the `showInventory` command for all selected chassis.

`baseline/chassis.snmpCommunityConf` – Output of the `snmpCommunityConf` command for all selected chassis.

`baseline/chassis.snmpTargetAddr` – Output of the `snmpTargetAddr` command for all selected chassis.

`baseline/chassis.timeDSTConf` – Output of the `timeDSTConf` command for all selected chassis.

`baseline/chassis.timeZoneConf` – Output of the `timeZoneConf` command for all selected chassis.

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

4.10.4.8 Full analysis

`latest/chassis.hwCheck` – Output of the `hwCheck` command for all selected chassis.

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

`latest/chassis.fwVersion.diff` – diff of the baseline and latest `fwVersion`.

`latest/chassis.ismChassisSet12x` – Output of the `ismChassisSet12x` command for all selected chassis.

`latest/chassis.ismChassisSet12x.diff` – diff of the baseline and latest `ismChassisSet12x`.

`latest/chassis.ismShowPStatThresh` – Output of the `ismShowPStatThresh` command for all selected chassis.

`latest/chassis.ismShowPStatThresh.diff` – diff of baseline and latest `ismShowPStatThresh`.

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

`latest/chassis.showChassisIpAddr.diff` – diff of baseline and latest `showChassisIpAddr`.

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

`latest/chassis.showDefaultRoute.diff` – diff of the baseline and the latest `showDefaultRoute`.

`latest/chassis.showIBNodeDesc` – Output of the `showIBNodeDesc` command for all selected chassis.

`latest/chassis.showIBNodeDesc.diff` – diff of the baseline and latest `showIBNodeDesc`.

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

`latest/chassis.showInventory.diff` – diff of the baseline and latest `showInventory`.



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

`latest/chassis.snmpCommunityConf.diff` – diff of the baseline and latest `snmpCommunityConf`.

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

`latest/chassis.snmpTargetAddr.diff` – diff of the baseline and latest `snmpTargetAddr`.

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

`latest/chassis.timeDSTConf.diff` – diff of the baseline and latest `timeDSTConf`.

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

`latest/chassis.timeZoneConf.diff` – diff of the baseline and latest `timeZoneConf`.

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

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

`FF_ANALYSIS_DIR/2007-11-22-09:53:04`

4.10.4.9 Chassis items checked against the baseline

Based upon `showInventory`:

- Addition/removal of Chassis FRUs
 - Replacement is only checked for FRUs that `showInventory` displays the serial number. For the 9000 series, the fan and power supply replacement is not checked, just present.
- Removal of redundant FRUs (spines, power supply, fan)

Based upon `fwVersion`:

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

Based upon `showIBNodeDesc`:

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

Based upon `ismShowPStatThresh`:

- Changes to configured port thresholds for chassis port error thresholding

Based upon `ismChassisSet12x`:

- Changes to chassis link width controls. Note that changes detected here would also be detected in fabric level analysis.

Based upon `timeZoneConf` and `timeDSTConf`:

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

Based upon `snmpCommunityConf` and `snmpTargetAddr`:



- changes to SNMP persistent configuration within the chassis

The following Chassis items will not be checked against baseline:

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

4.10.4.10 Chassis Items also checked during healthcheck

Based upon `hwCheck`:

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

4.10.5 hostsm_analysis

(All) The `hostsm_analysis` command performs analysis against the local server only.

4.10.5.1 Usage

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

4.10.5.2 Options

`-b` – Baseline mode. The default is the compare/check mode.

`-e` – Evaluate health only. The default is the compare/check mode.

`-s` – Save history of failures (for example, errors/differences).

`-d dir` – Top level directory for saving baseline and history of failed checks. The default is `/var/opt/iba/analysis`.

4.10.5.3 Example

```
hostsm_analysis
```

The host SM analysis tool checks the following:

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

The `hostsm_analysis` tool performs analysis against the local server only. It is assumed that both the host SM and Intel® FastFabric Toolset are installed on the same system.



4.10.5.4 Environment Variables

The following environment variables are also used by this command:

`FF_CURTIME` – Timestamp to use on the directory created in `FF_DIFF_CMD` – Linux command to use to compare baseline to latest snapshot

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

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

4.10.5.5 Health Check

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

4.10.5.6 Baseline

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

`baseline/hostsm.smconfig` – Copy of `ifs_fm.config`.

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

4.10.5.7 Full analysis

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

`latest/hostsm.smver` – Host SM version.

`latest/hostsm.smver.diff` – diff of the baseline and latest host SM version.

`latest/hostsm.smconfig` – Copy of `ifs_fm.xml`.

`latest/hostsm.smconfig.diff` – diff of the baseline and the latest `ifs_fm.xml`.

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

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

`FF_ANALYSIS_DIR/2007-11-22-09:53:04`

4.10.5.8 Host SM items checked against the baseline

- SM configuration file
- The version of the SM rpm installed on the system

4.10.5.9 Host SM items also checked during healthcheck

- The SM is in the running state

4.10.6 esm_analysis

(Switch) The `esm_analysis` command performs analysis of the embedded SM for configuration, and health.



4.10.6.1 Usage

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

4.10.6.2 Options

- b – Baseline mode. The default is the compare/check mode.
- e – Evaluate health only. The default is the compare/check mode.
- s – Save history of failures (for example, errors/differences).
- d *dir* – Top level directory for saving baseline and history of failed checks. The default is /var/opt/iba/analysis.
- G *esmchassisfile* – File with SM chassis within the cluster. The default is /etc/sysconfig/iba/esm_chassis.
- E *esmchassis* – List of the SM chassis to be analyzed.

4.10.6.3 Example

```
esm_analysis
```

The embedded SM analysis tool checks the following:

- Embedded SM configuration (as reported by the chassis commands specified in `FF_ESM_CMDS` in `fastfabric.conf`).
- Embedded SM health (as reported by `smControl status`).
- For Intel® 12000 Chassis, the `ifs_fm.xml` file for the chassis is also checked

4.10.6.4 Environment Variables

The following environment variables are also used by this command:

`ESM_CHASSIS`, `ESM_CHASSIS_FILE` – See the discussion on the “[Selection of Chassis](#)” on page 24. These have the same format as `CHASSIS` and `CHASSIS_FILE`.

`FF_TIMEOUT_MULT` – Multiplier for response time-outs. The default is 2. This typically does not need to be set, but in the event of unexpected time-outs or extremely slow chassis or management network, a larger value can be used.

`FF_CHASSIS_LOGIN_METHOD` – How to login to a chassis. Can be SSH or Telnet.

`FF_CHASSIS_ADMIN_PASSWORD` – Password for administrator on all chassis.

`FF_CURTIME` – Time stamp to use on the directory created in `ff_analysis_dir`. The default is the current date and time.

`FF_ESM_CMDS` – Set of chassis CLI commands to get the SM configuration.

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

Setup of ssh keys for chassis (see “[setup_ssh](#)” on page 159) is recommended. If ssh keys are not setup, all chassis must be configured with the same admin password and the password must be kept in the `fastfabric.conf` configuration file.

The default set of `FF_ESM_CMDS` is:

```
smShowSMParms smShowDefBcGroup
```



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

The `esm_analysis` command performs analysis against one or more chassis in the fabric. As such it permits a chassis to be specified using the `-E`, `-G`, `ESM_CHASSIS`, `ESM_CHASSIS_FILE` or `fastfabric.conf`. The handling of these options and settings is comparable to `cmdall -C` and similar Intel® FastFabric Toolset commands against a chassis. The exception in this case is that the option and variable names are slightly different to distinguish the fact that they are specifying only the chassis that has an embedded SM running).

All files generated by `esm_analysis` start with `esm` within the file name.

The `esm_analysis` command generates files such as the following within `FF_ANALYSIS_DIR`. The actual file names reflect the individual chassis commands that have been configured using the `FF_ESM_CMDS` parameter:

4.10.6.5 Health Check

`latest/esm.smstatus` – Output of the `smControl status` command for all selected chassis.

4.10.6.6 Baseline

`baseline/esm.smShowDefBcGroup` – Output of the `smShowDefBcGroup` command for all selected chassis.

`baseline/esm.smShowSMParms` – Output of the `smShowSMParms` command for all selected chassis. Latest `/esm.CHASSIS.ifs_fm.xml` – the `ifs_fm.xml` file for the given chassis

During a baseline run, the above files are also created in `ff_analysis_dir/latest`.

4.10.6.7 Full analysis

`latest/esm.smstatus` – Output of the `smControl status` command for all selected chassis.

`latest/esm.smShowDefBcGroup` – Output of the `smShowDefBcGroup` command for all selected chassis.

`latest/esm.smShowDefBcGroup.diff` – diff of baseline and latest `smShowDefBcGroup`.

`latest/esm.smShowSMParms` – Output of the `smShowSMParms` command for all selected chassis `latest/esm`.

`latest/smShowSMParms.diff` – diff of the baseline and the latest `smShowSMParms`.

`latest/esm.CHASSIS.ifs_fm.xml` – `ifs_fm.xml` file for the given chassis

`latest/esm.CHASSIS.ifs_fm.xml.diff` – diff of the baseline and the latest `ifs_fm.xml` for the given chassis.

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



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

`FF_ANALYSIS_DIR/YYYY-MM-DD-hh:mm:ss`

4.10.6.8 Chassis SM items that are checked against the baseline

Based upon `smShowSMParms`:

- SM priority
- SM sweep rate
- SM retry and time-out
- SM fabric time-outs configured (`switchLifeTime`, `HoqLife`, `VLStall`, `PacketLifeTimes` for `PathRecords`)
- Multipath mode
 - Based on `smShowDefBcGroup`
- Default IPoIB broadcast group settings in SM (`PKey`, `MTU`, `Rate`, `SL`)
- For Intel® 12000 Chassis, the entire `ifs_fm.xml` file is also compared.

4.10.6.9 Chassis SM items also checked during healthcheck

Based upon `smControl status`:

- SM is in running state

4.10.7 all_analysis

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

4.10.7.1 Usage

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

4.10.7.2 Options

- `-b` – Baseline mode. The default is the compare/check mode.
- `-e` – Evaluate health only. The default is the compare/check mode.
- `-s` – Save history of failures (for example, errors and differences).
- `-d dir` – Top-level directory for saving baseline and a history of failed checks. The default is `/var/opt/iba/analysis`.
- `-c file` – An error thresholds configuration file. The default is `/etc/sysconfig/iba/iba_mon.conf`.
- `-t portsfile` – File with a list of local HCA ports used to access fabric(s) for analysis. The default is `/etc/sysconfig/iba/ports`.
- `-p ports` – List of local HCA ports used to access fabric(s) for analysis. The default is the first active port. This is specified as `hca:port`:
 - `0:0` – First active port in system
 - `0:y` – Port y within system



x:0 – First active port on HCA x

x:y – HCA x, port y

-T *topology_input* – Name of topology input file to use. Any %P markers in this filename will be replaced with the hca:port being operated on (such as 0:0 or 1:2). The default is /etc/sysconfig/iba/topology.%P.xml. If -T NONE is specified, no topology input file will be used. See “[iba_report](#)” on page 175 for more information.

-F *chassisfile* – File with a chassis in a cluster. The default is /etc/sysconfig/iba/chassis.

-H *chassis* – List of chassis to execute the command on.

-G *esmchassisfile* – File with the SM chassis in the cluster. The default is /etc/sysconfig/iba/esm_chassis.

-H *esmchassis* – List of SM chassis to analyze.

4.10.7.3 Example

```
all_analysis
```

```
all_analysis -p '1:1 1:2 2:1 2:2'
```

The `all_analysis` command will perform the set of analysis specified in `FF_ALL_ANALYSIS`. This can be provided by the environment or using `fastfabric.conf`. The set of analysis which can be specified are: `fabric`, `chassis`, `esm` or `hostsm`. `FF_ALL_ANALYSIS` must be a space-separated list of the values mentioned above. These correspond to the respective analysis commands previously discussed.

Note that the `all_analysis` command has options which are a superset of the options for all other analysis commands. The options will be passed along to the respective tools (for example, the `-c` file option will be passed on to `fabric_analysis` if it is specified in `FF_ALL_ANALYSIS`).

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

4.10.7.4 Environment Variables

The following environment variables are also used by this command:

`CHASSIS`, `CHASSIS_FILE` – See “[Selection of Chassis](#)” on page 24.

`ESM_CHASSIS`, `ESM_CHASSIS_FILE` – See “[Selection of Chassis](#)” on page 24. These have the same format as `chassis` and `chassis_file`.

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

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

`FF_TOPOLOGY_FILE` – File containing `topology_input` (may have %P marker in filename), used in absence of `-T`.

`FF_TIMEOUT_MULT` – Multiplier for response time-outs. The default is 2. This typically does not need to be set, but in the event of unexpected time-outs or extremely slow chassis or management network, a larger value can be SSH or Telnet.

`FF_CHASSIS_ADMIN_PASSWORD` – Password for admin on all chassis. Used in absence of `-S` option.



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

`FF_CURTIME` – Time stamp to use on the directory created in `FF_ANALYSIS_DIR`. The default is the present date and time.

`FF_FABRIC_HEALTH` – `iba_report` options to use during a health check.

`FF_CHASSIS_CMDS` – Set of chassis CLI commands to get the chassis configuration.

`FF_CHASSIS_HEALTH` – Chassis CLI command to check the chassis health.

`FF_ESM_CMDS` – Set of chassis CLI commands to get the SM configuration.

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

4.10.8 Manual and Automated Usage

There are two basic ways to use the tools:

- Manual
- Automated

In both cases the user should follow the initial setup procedure outlined above to create a good baseline of the configuration.

In the manual method, the user would run the tools manually when trying to diagnose problems, or when there is a concern or need to validate the configuration and health.

In the automated method, the user could run `all_analysis` or a specific tool in an automated script (such as a `cron` job). When run in this mode the `-s` option may prove useful (but care must be taken to avoid excessive saved failures). When run in automated mode, a frequency of no faster than hourly would be recommended. For many fabrics a run daily or perhaps every few hours would be sufficient. Since the exit code from each of the tools indicates the overall success/failure, an automated script could easily check the exit status and on failure e-mail the output from the analysis tool to the appropriate administrators for further analysis and corrective action as needed.

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

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

4.10.9 Re-establishing Health Check baseline

This is needed after changing the fabric configuration in any way. The following activities are examples of ways in which the fabric configuration may be changed:

- Repair a faulty leaf board, which leads to a new serial number for that component.



- Update switch firmware or Fabric Manager
- Change time zones in a switch
- Add or delete a new device or link to a fabric
- A link fails and its devices are removed from the Fabric Manager database.

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

1. Make sure that you have fixed all problems with the fabric, including inadvertent configuration changes, before proceeding.
2. Verify that the fabric configured is as expected. The simplest way to do this is to run `fabric_info`. This will return information for each subnet to which the fabric management server is connected. The following is an example output for a single subnet. The comments are not part of the output. They are only included to help understand the output better.

```
SM: c999f4nm02 HCA-2 Guid: 0x0008f104039908e5 State: Master
```

```
Number of CAs: 53 # one for each HCA port; including the Fabric/MS
```

```
Number of CA Ports: 53 # same as number of CAs
```

```
Number of Switch Chips: 76 # one per IBM GX HCA port + one per switch leaf + two per switch spine
```

```
Number of Links: 249 # one per HCA port + 12 per leaf
```

```
Number of 1x Ports: 0
```

3. Save the old baseline. This may be required for future debug. The old baseline is a group of files in `/var/opt/iba/analysis/baseline`.
4. Run `all_analysis -b`
5. Check the new output files in `/var/opt/iba/analysis/baseline` to verify that the configuration is as you expect it. Refer to the *Intel® True Scale Fabric Suite FastFabric User Guide* for details.

4.10.10 Interpreting the Health Check Results

When any of the health check tools are run, the overall success or failure will be indicated in the output of the tool and its exit status. The tool will also indicate which areas had problems and which files should be reviewed. The results from the latest run can be found in `FF_ANALYSIS_DIR/latest/`. Many files can be found in this directory which indicate both the latest configuration of the fabric and errors/differences found during the health check. Should the health check fail, the following paragraphs will discuss a recommended order for reviewing these files.

If the `-s` option was used when running the health check, a directory whose name is the date and time of the failing run will be created under `FF_ANALYSIS_DIR`. In which case that directory can be consulted instead of the `latest` directory shown in the examples below.

It is recommended to first review the results for any `esm` or `hostsm` health check failures. If the SM is misconfigured or not running, it can cause other health checks to fail. In which case the SM problems should be corrected first then the health check should be rerun and other problems should then be reviewed and corrected as needed.

For a `hostsm` analysis, the files should be reviewed in the following order:

```
latest/hostsm.smstatus
```

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



`latest/hostsm.smver.diff` – This indicates the SM version has changed. If this was not an expected change, the SM should be corrected before proceeding further. Once corrected the health checks should be rerun to look for further errors. If the change was expected and permanent, a baseline should be rerun once all other health check errors have been corrected.

`latest/hostsm.smconfig.diff` – This indicates that the SM configuration has changed. This file should be reviewed and as necessary the `latest/hostsm.smconfig` file should be compared to `baseline/hostsm.smconfig`. As necessary correct the SM configuration. Once corrected the health checks should be rerun to look for further errors. If the change was expected and permanent, a baseline should be rerun once all other health check errors have been corrected.

For an `esm` analysis, the `FF_ESM_CMDS` configuration setting will select which ESM commands are used for the analysis. When using the default setting for this parameter, the files should be reviewed in the following order:

`latest/esm.smstatus` – Make sure this indicates the SM is running. If no SMs are running on the fabric, that problem should be corrected before proceeding further. Once corrected the health checks should be rerun to look for further errors.

`latest/esm.CHASSIS.ifs_fm.xml` – The `ifs_fm.xml` file for the given chassis

`latest/esm.CHASSIS.ifs_fm.xml.diff` – This indicates that the SM configuration has changed. This file should be reviewed and as necessary the `latest/esm.CHASSIS.ifs_fm.xml` file should be compared to `baseline/esm.CHASSIS.ifs_fm.xml`. As necessary correct the SM configuration. Once corrected the health checks should be rerun to look for further errors. If the change was expected and permanent, a baseline should be rerun once all other health check errors have been corrected.

`latest/esm.smShowSMParms.diff` – This indicates that the SM configuration has changed. This file should be reviewed and as necessary the `latest/esm.smShowSMParms` file should be compared to `baseline/esm.smShowSMParms`. As necessary correct the SM configuration. Once corrected the health checks should be rerun to look for further errors. If the change was expected and permanent, a baseline should be rerun once all other health check errors have been corrected.

`latest/esm.smShowDefBcGroup.diff` – This indicates that the SM broadcast group for IPoIB configuration has changed. This file should be reviewed and as necessary the `latest/esm.smShowDefBcGroup` file should be compared to `baseline/esm.smShowDefBcGroup`. As necessary correct the SM configuration. Once corrected the health checks should be rerun to look for further errors. If the change was expected and permanent, a baseline should be rerun once all other health check errors have been corrected.

`latest/esm.*.diff` – If `FF_ESM_CMDS` has been modified, the changes in results for those additional commands should be reviewed. As necessary correct the SM. Once corrected the health checks should be rerun to look for further errors. If the change was expected and permanent, a baseline should be rerun once all other health check errors have been corrected.

Next, it is recommended to review the results of the fabric analysis for each configured fabric. If nodes or links are missing, the fabric analysis will detect them. Missing links or nodes can cause other health checks to fail. If such failures are expected (for example, a node or switch is offline), further review of result files can be performed, but the user must beware that the loss of the node or link can cause other analysis to also fail. The discussion below presents the analysis order for `fabric.0.0`, if other or additional fabrics are configured for analysis, it is recommended to review the files in the order shown below for each fabric. There is no specific order recommended for which fabric to review first.

`latest/fabric.0.0.errors.stderr` – If this file is not empty, it can indicate problems with `iba_report` (such as inability to access an SM) which can result in unexpected problems or inaccuracies in the related errors file. If possible problems



reported in this file should be corrected first. Once corrected the health checks should be rerun to look for further errors.

`latest/fabric.0:0.errors` – If any links with excessive error rates or incorrect link speeds are reported, they should be corrected. If there are links with errors, beware the same links may also be detected in other reports such as the `links` and `comps` files discussed below.

`latest/fabric.0:0.snapshot.stderr` – If this file is not empty, it can indicate problems with `iba_report` (such as inability to access an SM) which can result in unexpected problems or inaccuracies in the related `links` and `comps` files. If possible, problems reported in this file should be corrected first. Once corrected the health checks should be rerun to look for further errors.

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

`latest/fabric.0:0.links.changes.stderr` – If these files are not empty, it can indicate problems with `iba_report` which can result in unexpected problems or inaccuracies in the related `links` files. If possible, problems reported in these files should be corrected first. Once corrected the health checks should be rerun to look for further errors. For more information on `.changes` files refer to [“Interpreting Health Check .changes Files” on page 291](#).

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

`latest/fabric.0:0.links.changes` – These indicate that the links between components in the fabric have changed, been removed/added or that components in the fabric have disappeared. If both files are available, the `fabric.0:0.links.changes` file should be used since it will have a more concise and precise description of the fabric link changes. As necessary the `latest/fabric.0:0.links` file should be compared to `baseline/fabric.0:0.links`. If components have disappeared, review of the `latest/fabric.0:0.comps.diff` and `latest/fabric.0:0.comps.changes` files may be easier for such components. As necessary correct missing nodes and links. Once corrected the health checks should be rerun to look for further errors. If the change was expected and is permanent, a baseline should be rerun once all other health check errors have been corrected. For more information on `.changes` files refer to [“Interpreting Health Check .changes Files” on page 291](#).

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

`latest/fabric.0:0.comps.changes.stderr` – If these files are not empty, it can indicate problems with `iba_report` which can result in unexpected problems or inaccuracies in the related `comps` file. If possible, problems reported in these files should be corrected first. Once corrected the health checks should be rerun to look for further errors. For more information on `.changes` files refer to [“Interpreting Health Check .changes Files” on page 291](#).

`latest/fabric.0:0.comps.diff` and `latest/fabric.0:0.comps.changes` – These indicate that the components in the fabric or their SMA configuration have changed. If both files are available, the `fabric.0:0.comps.changes` file should be used since it will have a more concise and precise description of the fabric component changes. As necessary the `latest/fabric.0:0.comps` file should be compared to `baseline/fabric.0:0.comps`. As necessary correct missing nodes, missing SMs, ports which are down and port misconfigurations. Once corrected the health checks should be rerun to look for further errors. If the change was expected and permanent, a baseline should be rerun once all other health check errors have been corrected. For more information on `.changes` files refer to [“Interpreting Health Check .changes Files” on page 291](#).

Intel® recommends to review the results of the `chassis_analysis`. If chassis configuration has changed, the `chassis_analysis` will detect it. Previous checks should have already detected missing chassis, missing or added links and many aspects of chassis True Scale Fabric configuration. For `chassis_analysis`, the `FF_CHASSIS_CMDS` and `FF_CHASSIS_HEALTH` configuration settings will select which chassis commands are used for the analysis. When using the default setting for this parameter, the files should be reviewed in the following order:



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

`latest/chassis.fwVersion.diff` – This indicates the chassis firmware version has changed. If this was not an expected change, the chassis firmware should be corrected before proceeding further. Once corrected the health checks should be rerun to look for further errors. If the change was expected and permanent, a baseline should be rerun once all other health check errors have been corrected.

`latest/chassis/*.diff` – These files reflect other changes to chassis configuration based on checks selected by `FF_CHASSIS_CMDS`. The changes in results for these remaining commands should be reviewed. As necessary correct the chassis. Once corrected the health checks should be rerun to look for further errors. If the change was expected and permanent, a baseline should be rerun once all other health check errors have been corrected.

If any health checks failed, after correcting the related issues, another health check should be run to verify the issues were all corrected. If the failures are due to expected and permanent changes, once all other errors have been corrected, a baseline should be rerun.

4.10.11 Interpreting Health Check .changes Files

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

The format is like the following:

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

In the following example of `fabric.*:*.links.changes`, you will note that it only shows links that were "Unexpected". That means that the link was not found in the previous baseline. The issue "Unexpected Link" is listed after the link is presented.

Links Topology Verification

Links Found with incorrect configuration:

Rate MTU NodeGUID Port Type Name

60g 4096 0x00025500105baa00 1 CA IBM G2 Logical HCA

<-> 0x00025500105baa02 2 SW IBM G2 Logical Switch 1

Unexpected Link

20g 4096 0x00025500105baa02 1 SW IBM G2 Logical Switch 1

<-> 0x00066a0007000dbb 4 SW SilverStorm 9080 c938f4ql01 Leaf 2, Chip A

Unexpected Link



60g 4096 0x00025500106cd200 1 CA IBM G2 Logical HCA

<-> 0x00025500106cd202 2 SW IBM G2 Logical Switch 1

Unexpected Link

20g 4096 0x00025500106cd202 1 SW IBM G2 Logical Switch 1

<-> 0x00066a0007000dbb 5 SW SilverStorm 9080 c938f4ql01 Leaf 2, Chip A

Unexpected Link

60g 4096 0x00025500107a7200 1 CA IBM G2 Logical HCA

<-> 0x00025500107a7202 2 SW IBM G2 Logical Switch 1

Unexpected Link

20g 4096 0x00025500107a7202 1 SW IBM G2 Logical Switch 1

<-> 0x00066a0007000dbb 3 SW SilverStorm 9080 c938f4ql01 Leaf 2, Chip A

Unexpected Link

165 of 165 Fabric Links Checked

Links Expected but Missing, Duplicate in input or Incorrect:

159 of 159 Input Links Checked

Total of 6 Incorrect Links found

0 Missing, 6 Unexpected, 0 Misconnected, 0 Duplicate, 0 Different

Table 7 summarizes possible issues found in .changes files:


Table 7. Possible issues found in health check .changes files

Issue	Description and possible actions
Missing	<p>This indicates an item that is in the baseline, is not in this instance of health check output. This may indicate a broken item or a configuration change that has removed the item from the configuration.</p> <p>If you have intentionally removed this item from the configuration, save the original baseline and re-run the baseline as instructed in "Re-establishing Health Check baseline" on page 287. For example, if you've removed an HCA connection, the HCA and the link to it will be shown as Missing in <code>fabric.*:*.links.changes</code> and <code>fabric.*:*.comps.changes</code> files.</p> <p>If the item should still be part of the configuration, check for faulty connections or unintended changes to configuration files on the fabric management server.</p> <p>You should also look for any "Unexpected" or "Different" items that may correspond to this item. This would be in cases where the configuration of an item has changed in a way that makes it difficult to determine precisely how it has changed.</p>
Unexpected	<p>This indicates that an item is in this instance of health check output, but it is not in the baseline. This may indicate that an item was broken when the baseline was taken or a configuration change has added the item to the configuration.</p> <p>If you have added this item to the configuration, save the original baseline and re-run the baseline as instructed in "Re-establishing Health Check baseline" on page 287. For example, if you've added an HCA connection, will be shown as Unexpected in <code>fabric.*:*.links.changes</code> and <code>fabric.*:*.comps.changes</code> files.</p> <p>You should also look for any "Missing" or "Different" items that may correspond to this item. This would be in cases where the configuration of an item has changed in a way that makes it difficult to determine precisely how it has changed.</p>
Misconnected	<p>This only applies to links and indicates that a link is not connected properly. This should be fixed.</p> <p>It is possible to find miswires by examining all of the Misconnected links in the fabric. However, you will have to look at all of the <code>fabric.*:*.links.changes</code> files to find miswires between subnets.</p> <p>You should also look for any "Missing" or "Different" items that may correspond to this item. This would be in cases where the configuration of an item has changed in a way that makes it difficult to determine precisely how it has changed.</p> <p>Individual links which are Misconnected are reported as "Incorrect Link" (see "Incorrect Link" on page 294) and are added into the Misconnected summary count.</p>
Duplicate	<p>This indicates that an item has a duplicate in the fabric. This situation should be resolved such that there is only one instance of any particular item being discovered in the fabric.</p> <p>This error can occur if there are changes in the fabric such as addition of parallel links. It can also be reported when there enough changes to the fabric that it is difficult to properly resolve and report all the changes. It can also occur when <code>iba_report</code> is run with manually generated topology input files which may have duplicate items or incomplete specifications.</p>
Different	<p>This indicates that an item still exists in the current health check, but it is different from the baseline configuration.</p> <p>If the configuration has changed purposely since the most recent baseline, and the expected difference is reflected here, save the original baseline and re-run the baseline as instructed in "Re-establishing Health Check baseline" on page 287.</p> <p>If this difference was not intended, you must rectify the difference to prevent future health checks from reporting the same difference from the baseline.</p> <p>You should also look for any "Missing" or "Unexpected" items that may correspond to this item. This would be in cases where the configuration of an item has changed in a way that makes it difficult to determine precisely how it has changed.</p> <p>Individual items which are Different will be reported as "mismatched" or "Inconsistent" and are added into the Different summary count. See "X mismatch: expected ... found:" on page 294, "Node Attributes Inconsistent" on page 294, "Port Attributes Inconsistent" on page 294, or "SM Attributes Inconsistent" on page 294.</p>

Table 7. Possible issues found in health check .changes files (Continued)

Issue	Description and possible actions
Port Attributes Inconsistent	<p>This indicates that the attributes of a port on one side of a link have changed, such as PortGuid, Port Number, Device Type, etc. The inconsistency would be caused by connecting a different type of device or a different instance of the same device type. This would also occur after replacing a faulty device.</p> <p>If the configuration has changed purposely since the most recent baseline, and this difference is reflected here, save the original baseline and re-run the baseline as instructed in "Re-establishing Health Check baseline" on page 287. If a faulty device were replaced, this would be a reason to re-establish the baseline.</p> <p>If this difference was not intended, you must rectify the difference to prevent future health checks from reporting the same difference from the baseline.</p> <p>This is a specific case of "Different". See "Different" on page 293.</p>
Node Attributes Inconsistent	<p>This indicates that the attributes of a node in the fabric have changed, such as NodeGuid, Node Description, Device Type, etc. The inconsistency would be caused by connecting a different type of device or a different instance of the same device type. This would also occur after replacing a faulty device.</p> <p>If the configuration has changed purposely since the most recent baseline, and this difference is reflected here, save the original baseline and re-run the baseline as instructed in "Re-establishing Health Check baseline" on page 287. If a faulty device were replaced, this would be a reason to re-establish the baseline.</p> <p>If this difference was not intended, you must rectify the difference to prevent future health checks from reporting the same difference from the baseline.</p> <p>This is a specific case of "Different". See "Different" on page 293.</p>
SM Attributes Inconsistent	<p>This indicates that the attributes of the node or port running an SM in the fabric have changed, such as NodeGuid, Node Description, Port Number, Device Type, etc. The inconsistency would be caused by moving a cable, changing from host-based subnet management to embedded subnet management (or vice-versa), or by replacing the HCA in the fabric management server.</p> <p>If the configuration has changed purposely since the most recent baseline, and this difference is reflected here, save the original baseline and re-run the baseline as instructed in "Re-establishing Health Check baseline" on page 287. If the HCA in the fabric management server were replaced, this would be a reason to re-establish the baseline.</p> <p>If this difference was not intended, you must rectify the difference to prevent future health checks from reporting the same difference from the baseline.</p> <p>This is a specific case of "Different". See "Different" on page 293.</p>
X mismatch: expected ... found:	<p>This indicates an aspect of an item has changed as compared to the baseline configuration. The aspect which changed and the expected and found values will be shown. This will typically indicate configuration differences such as MTU, Speed, Node description. It can also indicate that GUIDs have changed, such as when replacing a faulty device. (perhaps due to replacement of a device with a comparable device).</p> <p>If the configuration has changed purposely since the most recent baseline, and this difference is reflected here, save the original baseline and re-run the baseline as instructed in "Re-establishing Health Check baseline" on page 287. If a faulty device were replaced, this would be a reason to re-establish the baseline.</p> <p>If this difference was not intended, you must rectify the difference to prevent future health checks from reporting the same difference from the baseline.</p> <p>This is a specific case of "Different". See "Different" on page 293.</p>
Incorrect Link	<p>This only applies to links and indicates that a link is not connected properly. This should be fixed.</p> <p>It is possible to find miswires by examining all of the Misconnected links in the fabric. However, you will have to look at all of the fabric.*:*.links.changes files to find miswires between subnets.</p> <p>You should also look for any "Missing" or "Different" items that may correspond to this item. This would be in cases where the configuration of an item has changed in a way that makes it difficult to determine precisely how it has changed.</p> <p>This is a specific case of "Misconnected". See "Misconnected" on page 293.</p>





5.0 MPI Sample Applications

As part of a Intel® FastFabric Toolset installation, some sample MPI applications and benchmarks are installed to `/opt/iba/src/mpi_apps`. These can be used to perform basic tests and performance analysis of MPI, the servers, and the fabric.

As part of this package the following sample applications are provided:

- Latency/bandwidth deviation test
- OSU latency (3 versions)
- OSU bandwidth (3 versions)
- OSU bidirectional bandwidth
- Pallas
- HPL
- NAS benchmarks

To build the applications:

1. Type `export MPICH_PREFIX=/usr/mpi/X/Y`
Where:
X is a compiler such as `gcc`
Y is an MPI variation such as `openmpi-1.2.5`.
2. Type `cd /opt/iba/src/mpi_apps`
3. Type `make clean`
4. Type `make full` (builds all of the above sample applications).

Note: The Intel® FastFabric Toolset TUI can assist in building the MPI sample applications in which case it provides a simple way to select the MPI to use for the build.

Alternatives to `full` include:

- `quick` — builds just Latency/Bandwidth Deviation, OSU, Pallas and HPL.
- `all` — builds just Latency/Bandwidth Deviation, OSU, Pallas, HPL and NAS benchmarks.

In order to run the applications an `mpi_hosts` file must be created in `/opt/iba/src/mpi_apps` that provides the names of the hosts on which processes should be run. Either IPoIB or Ethernet names can be specified. Typically, use of IPoIB names will provide faster job startup, especially on larger clusters. These run scripts allow the `mpi_hosts` filename to be specified through the environment variable `MPICH_HOSTS`. If this variable is not defined, the default of `mpi_hosts` will be used.

If a host has more than one real CPU, its name may appear in the MPI hosts file once per CPU.

Note: Intel Xeon processors support Hyperthreading. However, for floating point intensive MPI applications, such as NAS and HPL, this feature significantly impacts performance and should be disabled.

Note: When running the applications, all hosts listed in `mpi_hosts` must have a copy of the applications compiled for the same value of `MPICH_PREFIX` (for example, the same variation and version of MPI).

When the `run_*` scripts are used to execute the applications, the variation of MPI used to build the applications will be detected and the proper `mpirun` will be used to start the application.



To determine which variation of MPI the applications have been built for use the command:

```
cat /opt/iba/src/mpi_apps/.prefix
```

Note: Some variations of MPI may require that the MPD daemon be started prior to running applications. Consult the documentation on the specific variation of MPI for more information on how to start the MPD daemon.

When MPI applications are run with the `run_*` scripts provided, the results of the run will be logged to a file in `/opt/iba/src/mpi_apps/logs`. The file name will include the date and time of the run for uniqueness.

The `run_*` scripts automatically use the `ofed.openmpi.params`, `ofed.mvapich.params` or `ofed.mvapich2.params` files to setup parameters for `mpirun`. These files have various samples of setting parameters such as vFabric selection, dispersive routing, etc. These parameter files can also set the `MPI_CMD_ARGS` variable to provide additional arguments to `mpirun`.

5.1 Latency/Bandwidth Deviation Test

This is an analysis/diagnostic tool to perform assorted pairwise bandwidth and latency tests and report pairs outside an acceptable tolerance range. The tool will identify specific nodes which have problems and provide a concise summary of results.

This tool is also used by the Intel® FastFabric Toolset "Check MPI performance" TUI menu item and can also be invoked using `iba_host mpiperfdeviation`.

A script is provided to run this application:

1. Type `cd /opt/iba/src/mpi_apps`
2. Type `./run_deviation NP`

Where:

`NP` is the number of processes to run or `all`, such as:

```
./run_deviation 4
```

This will run a quick latency and bandwidth test against pairs of the hosts specified in `mpi_hosts`. By default each host is run against a single reference host and the results are analyzed. Pairs which have 20% less bandwidth or 50% more latency than the average pair will be reported as failures.

Note: For this test the `mpi_hosts` file should not list a given host more than once, regardless of how many CPUs the host has.

The tool can be run in a sequential or a concurrent mode. Sequential mode is the default and will run each host against a reference host. By default the reference host is selected based on the best performance from a quick test of the first 40 hosts.

In concurrent mode, hosts are paired up and all pairs are run concurrently. Since there may be fabric contention during such as run, any poor performing pairs are then rerun sequentially against the reference host.

Concurrent mode will run the tests in the shortest amount of time, however the results could be slightly less accurate due to switch contention. In heavily oversubscribed fabric designs, if concurrent mode is producing unexpectedly low performance, try sequential mode.

`run_deviation` supports a number of parameters that allow for more precise control over the mode, benchmark and pass/fail criteria.



'ff' When specified, the configured FF_DEVIATION_ARGS will be used

bwtol Percent of bandwidth degradation allowed below Avg value

lattel Percent of latency degradation allowed above Avg value

Other deviation arguments:

[bwbidir] [bwunidir] [-bwdelta MBs] [-bwthres MBs] [-bwloop count]

[-bwsiz size] [-latdelta usec] [-latthres usec] [-latloop count] [-latsiz size]
[-c] [-b] [-v] [-vv] [-h reference_host]

-bwbidir Perform a bidirectional bandwidth test

-bwunidir Perform a unidirectional bandwidth test (default)

-bwdelta Limit in MB/s of bandwidth degradation allowed below Avg value

-bwthres Lower Limit in MB/s of bandwidth allowed below Avg value

-bwloop Number of loops to execute each bandwidth test

-bwsiz Size of message to use for bandwidth test

-latdelta Limit in usec of latency degradation allowed above Avg value

-latthres Upper Limit in usec of latency allowed

-latloop Number of loops to execute each latency test

-latsiz Size of message to use for latency test

-c Run test pairs concurrently instead of the default of sequential

-b When comparing results against tolerance and delta use best
instead of Avg

-v verbose output

-vv Very verbose output

-h Baseline host to use for sequential pairing

Both bwtol and bwdelta must be exceeded to fail bandwidth test

When bwthres is supplied, bwtol and bwdelta are ignored

Both lattol and latdelta must be exceeded to fail latency test

When latthres is supplied, lattol and latdelta are ignored

For consistency with OSU benchmarks MB/s is defined as 1000000 bytes/s

Examples:

./run_deviation 20 ff

./run_deviation 20 ff -v



```
./run_deviation 20 20 50 -c  
./run_deviation 20 '' '' -c -v -bwthres 1200.5 -latthres 3.5  
./run_deviation 20 20 50 -c -h compute0001  
./run_deviation 20 0 0 -bwdelta 200 -latdelta 0.5
```

Example of 4 hosts with both 20% bandwidth and latency tolerances running in concurrent mode using the verbose option with a specified baseline host.

```
./run_deviation 4 20 20 -c -v -h hostname
```

5.2 OSU Latency

This is a simple benchmark of end-to-end latency for various MPI message sizes. The values reported are one-direction latency.

A script is provided to run this application that will execute an assortment of sizes:

1. Type `cd /opt/iba/src/mpi_apps`
2. Type `./run_lat`

This will run assorted latencies from 0 to 256 bytes. To run a different set of message sizes an optional argument specifying the maximum message size can be provided.

This benchmark will only use the first two nodes listed in `mpi_hosts`.

5.3 OSU Latency2

This is a simple performance test of end-to-end latency for various MPI message sizes. The values reported are one-direction latency.

A script is provided to run this application, which will execute an assortment of sizes:

1. Type `cd /opt/iba/src/mpi_apps`
2. Type `./run_lat2`

This will run assorted latencies from 0 to 4 Megabytes.

This benchmark will only use the first two nodes listed in `mpi_hosts`.

5.4 OSU Latency 3

This is a simple performance test of end-to-end latency for various MPI message sizes. The values reported are one-direction latency.

A script is provided to run this application, which will execute an assortment of sizes:

1. Type `cd /opt/iba/src/mpi_apps`
2. Type `./run_lat3`

This will run assorted latencies from 0 to 4 Megabytes.

This benchmark will only use the first two nodes listed in `mpi_hosts`.



5.5 OSU Multi Latency3

This is a simple performance test of end-to-end latency for multiple concurrent pairs of hosts for various MPI message sizes. The values reported are average one-direction latency.

A script is provided to run this application, which will execute an assortment of sizes:

1. Type `cd /opt/iba/src/mpi_apps`
2. Type `./run_multi_lat3 NP`

Where:

NP is the number of processes to run or `all`, such as:

```
./run_multi_lat3 4
```

This will run assorted latencies from 0 to 4 Megabytes.

This benchmark will only use the first *NP* nodes listed in `mpi_hosts`.

5.6 OSU Bandwidth

This is a simple benchmark of maximum unidirectional bandwidth.

A script is provided to run this application which will execute an assortment of sizes:

1. Type `cd /opt/iba/src/mpi_apps`
2. Type `./run_bw`

This will run assorted bandwidths from 4K to 4Mbytes. To run a different set of message sizes an optional argument specifying the maximum message size can be provided.

This benchmark will only use the first two nodes listed in `mpi_hosts`.

5.7 OSU Bandwidth2

This is a simple benchmark of maximum unidirectional bandwidth.

A script is provided to run this application that will execute an assortment of sizes:

1. Type `cd /opt/iba/src/mpi_apps`
2. Type `./run_bw2`

This will run assorted bandwidths from 1 byte to 4Mbytes. This benchmark will only use the first two nodes listed in `mpi_hosts`.

5.8 OSU Bandwidth3

This is a simple benchmark of maximum unidirectional bandwidth.

A script is provided to run this application that will execute an assortment of sizes:

1. Type `cd /opt/iba/src/mpi_apps`
2. Type `./run_bw3`

This will run assorted bandwidths from 1 byte to 4Mbytes. This benchmark will only use the first two nodes listed in `mpi_hosts`.



5.9 OSU Multi Bandwidth3

This is a simple benchmark of aggregate unidirectional bandwidth and messaging rate for multiple concurrent pairs of nodes.

A script is provided to run this application that will execute an assortment of sizes:

1. Type `cd /opt/iba/src/mpi_apps`
2. Type `./run_mbw_mr3 NP`

Where:

NP is the number of processes to run or `all`, such as:

```
./run_mbw_mr3 4
```

This will run assorted messaging rates from 1 byte to 4Mbytes.

5.10 OSU Bidirectional Bandwidth

This is a simple benchmark of maximum bidirectional bandwidth.

A script is provided to run this application that will execute an assortment of sizes:

1. Type `cd /opt/iba/src/mpi_apps`
2. Type `./run_bibw2`

This will run assorted bandwidths from 1 byte to 4Mbytes. This benchmark will only use the first two nodes listed in `mpi_hosts`.

5.11 OSU Bidirectional Bandwidth3

This is a simple benchmark of maximum bidirectional bandwidth.

A script is provided to run this application that will execute an assortment of sizes:

1. Type `cd /opt/iba/src/mpi_apps`
2. Type `./run_bibw3`

This will run assorted bandwidths from 1 byte to 4Mbytes. This benchmark will only use the first two nodes listed in `mpi_hosts`.

5.12 OSU All to All 3

This is a simple benchmark of AllToAll latency.

A script is provided to run this application that will execute an assortment of sizes:

1. Type `cd /opt/iba/src/mpi_apps`
2. Type `./run_alltoall3 NP`

Where:

NP is the number of processes to run or `all`, such as:

```
./run_alltoall3 4
```

This will run assorted latencies from 1 byte to 1Mbytes.



5.13 OSU Broadcast 3

This is a simple benchmark of Broadcast latency.

A script is provided to run this application that will execute an assortment of sizes:

1. Type `cd /opt/iba/src/mpi_apps`
2. Type `./run_bcast3 NP`

Where:

`NP` is the number of processes to run or `all`, such as:

```
./run_bcast3 4
```

This will run assorted latencies from 1 byte to 16K bytes.

5.14 OSU Multiple Bandwidth/Message Rate

The Multiple Bandwidth / Message Rate Test (`osu_mbw_mr`) is intended to be used with block assigned ranks. This means that all processes on the same machine are assigned ranks sequentially.

Note: All benchmarks are run using 2 processes with the exception of `osu_bcast` and `osu_mbw_mr` which can use more than 2.

If you're using `mpirun_rsh` with `mvapich` the ranks are assigned in the order they are seen in the hostfile or on the command line. If you're using `mpd` with `mvapich2` you have to specify the number of processes on each host in the hostfile otherwise `mpd` will assign ranks in a cyclic fashion. Refer to [Table 8](#) for rank assignments.

Table 8. Rank Assignment

Rank	Block	Cyclic
0	host1	host1
1	host1	host2
2	host1	host1
3	host1	host2
4	host2	host1
5	host2	host2
6	host2	host1
7	host2	host2

The following is an example of MPD HOSTFILE:

```
host1:4
host2:4

MPI-1
-----
osu_bcast          - Broadcast Latency Test
```



```
osu_bibw      - Bidirectional Bandwidth Test
osu_bw        - Bandwidth Test
osu_latency   - Latency Test
osu_mbw_mr    - Multiple Bandwidth / Message Rate Test
osu_multi_lat - Multi-pair Latency Test

MPI-2
-----

osu_acc_latency - Accumulate Latency Test
osu_get_bw      - One-Sided Get Bandwidth Test
osu_get_latency - One-Sided Get Latency Test
osu_latency_mt  - Multi-threaded Latency Test
osu_put_bibw    - One-Sided Put Bidirectional Test
osu_put_bw      - One-Sided Put Bandwidth Test
osu_put_latency - One-Sided Put Latency Test
```

5.14.1 Latency Test

The latency tests were carried out in a ping-pong fashion. The sender sends a message with a certain data size to the receiver and waits for a reply from the receiver. The receiver receives the message from the sender and sends back a reply with the same data size. Many iterations of this ping-pong test were carried out and average one-way latency numbers were obtained. Blocking version of MPI functions (MPI_Send and MPI_Recv) were used in the tests. This test is available [here](#).

5.14.2 Multi-threaded Latency Test (only applicable for MVAPICH2 with threading support enabled)

The multi-threaded latency test performs a ping-pong test with a single sender process and multiple threads on the receiving process. In this test the sending process sends a message of a given data size to the receiver and waits for a reply from the receiver process. The receiving process has a variable number of receiving threads (set by default to 2), where each thread calls MPI_Recv and upon receiving a message sends back a response of equal size. Many iterations are performed and the average one-way latency numbers are reported. This test is available [here](#).

5.14.3 Bandwidth Test

The bandwidth tests were carried out by having the sender sending out a fixed number (equal to the window size) of back-to-back messages to the receiver and then waiting for a reply from the receiver. The receiver sends the reply only after receiving all these messages. This process is repeated for several iterations and the bandwidth is calculated based on the elapsed time (from the time sender sends the first message until the time it receives the reply back from the receiver) and the number of bytes sent by the sender. The objective of this bandwidth test is to determine the maximum sustained data rate that can be achieved at the network level. Thus, non-blocking version of MPI functions (MPI_Isend and MPI_Irecv) were used in the test. This test is available [here](#).



5.14.4 Bidirectional Bandwidth Test

The bidirectional bandwidth test is similar to the bandwidth test, except that both the nodes involved send out a fixed number of back-to-back messages and wait for the reply. This test measures the maximum sustainable aggregate bandwidth by two nodes. This test is available here.

5.14.5 Multiple Bandwidth / Message Rate test

The multi-pair bandwidth and message rate test evaluates the aggregate uni-directional bandwidth and message rate between multiple pairs of processes. Each of the sending processes sends a fixed number of messages (the window size) back-to-back to the paired receiving process before waiting for a reply from the receiver. This process is repeated for several iterations. The objective of this benchmark is to determine the achieved bandwidth and message rate from one node to another node with a configurable number of processes running on each node. The test is available here.

5.14.6 Multi-pair Latency Test

This test is very similar to the latency test. However, at the same instant multiple pairs are performing the same test simultaneously. In order to perform the test across just two nodes the hostnames must be specified in block fashion.

5.14.7 Broadcast Latency Test

Broadcast Latency Test: The Broadcast latency test was carried out in the following manner. After doing a MPI_Bcast the root node waits for an acknowledgment from the last receiver. This acknowledgment is in the form of a zero byte message from the receiver to the root. This test is carried out for a large number (1000) of iterations. The Broadcast latency is obtained by subtracting the time taken for the acknowledgment from the total time. We compute the acknowledgment time initially by doing a ping-pong test. This test is available here.

5.14.8 One-Sided Put Latency Test (only applicable for MVAPICH2)

One-Sided Put Latency Test: The sender (origin process) calls MPI_Put (ping) to directly place a message of certain data size in the receiver window. The receiver (target process) calls MPI_Win_wait to make sure the message has been received. Then the receiver initiates a MPI_Put (pong) of the same data size to the sender which is now waiting on a synchronization call. Several iterations of this test is carried out and the average put latency numbers is obtained. This test is available here.

5.14.9 One-Sided Get Latency Test (only applicable for MVAPICH2)

One-Sided Get Latency Test: The origin process calls MPI_Get (ping) to directly fetch a message of certain data size from the target process window to its local window. It then waits on a synchronization call (MPI_Win_complete) for local completion. After the synchronization call the target and origin process are switched for the pong message. Several iterations of this test are carried out and the average get latency numbers is obtained. This test is available here.



5.14.10 One-Sided Put Bandwidth Test (only applicable for MVAPICH2)

One-Sided Put Bandwidth Test: The bandwidth tests were carried out by the origin process calling a fixed number of back to back Puts and then wait on a synchronization call (MPI_Win_complete) for completion. This process is repeated for several iterations and the bandwidth is calculated based on the elapsed time and the number of bytes sent by the origin process. This test is available here.

5.14.11 One-Sided Get Bandwidth Test (only applicable for MVAPICH2)

One-Sided Get Bandwidth Test: The bandwidth tests were carried out by origin process calling a fixed number of back to back Gets and then wait on a synchronization call (MPI_Win_complete) for completion. This process is repeated for several iterations and the bandwidth is calculated based on the elapsed time and the number of bytes sent by the origin process. This test is available here.

5.14.12 One-Sided Put Bidirectional Bandwidth Test (only applicable for MVAPICH2)

One-Sided Put Bidirectional Bandwidth Test: The bidirectional bandwidth test is similar to the bandwidth test, except that both the nodes involved send out a fixed number of back to back put messages and wait for the completion. This test measures the maximum sustainable aggregate bandwidth by two nodes. This test is available here.

5.14.13 Accumulate Latency Test (only applicable for MVAPICH2)

One-Sided Accumulate Latency Test: The origin process calls MPI_Accumulate to combine the data moved to the target process window with the data that resides at the remote window. The combining operation used in the test is MPI_SUM. It then waits on a synchronization call (MPI_Win_complete) for local completion. After the synchronization call, the target and origin process are switched for the pong message. Several iterations of this test are carried out and the average accumulate latency number is obtained. This test is available here.

mpi_stress Test

This can be used to place stress on the interconnect as part of verifying stability. The run_mpi_stress script can be used to run this application.

5.14.13.0.1 Usage

```
run_mpi_stress number_processes [mpi_stress arguments]
```

5.14.13.0.2 Options

mpi_stress arguments:

- a *INT*: desired alignment for buffers (must be power of 2)
- b *BYTE*: byte value to initialize non-random send buffers (otherwise 0)
- c: enable CRC checksums
- D *INT*: set max data amount per msg size (default 1073741824)
- d: enable data checksums (otherwise headers only)
- e: exercise the interconnect with random length messages
- g *INT*: use INT-dimensional grid connectivity (non-periodic)
- G *INT*: use INT-dimensional grid connectivity (periodic) (default is to use all-to-all connectivity)
- h: display this help page
- i: include local ranks as destinations (only for all-to-all)



- I *INT*: set msg size increment (default power of 2)
- l *INT*: set min msg size (default 0)
- L *INT*: set min msg count (default 100)
- m *INT*: set max msg size (default 4194304)
- M *INT*: set max msg count (default 10000)
- n *INT*: number of times to repeat (default 1)
- O: show options and parameters used for the run.
- p: show progress
- P: poison receive buffers at init and after each receive
- q: quiet mode (don't show error details)
- r: fill send buffers with random data (else 0 or -b byte)
- R: round robin destinations (default is random selection)
- s: include self as a destination (only for all-to-all)
- S: use non-blocking synchronous sends (MPI_Issend)
- t *INT*: run for INT minutes (implicitly adds -n BIGNUM)
- u: uni-directional traffic (only for grid)
- v: enable verbose mode (more -v for more verbose)
- w *INT*: number of send/recv in window (default 20)
- x: enable XOR checksums
- z: enable typical options for data integrity (-drx) (for stronger integrity checking try using -drc instead)
- Z: zero receive buffers at init and after each receive

This an MPI stress test program designed to load up an MPI interconnect with point-to-point messages while optionally checking for data integrity. By default, it runs with all-to-all traffic patterns, optionally including oneself and one's local peers. It can also be set up with multi-dimensional grid traffic patterns, and this can be parameterized to run rings, open 2D grids, closed 2D grids, cubic lattices, hypercubes, etc. Optionally, the message data can be randomized and checked using CRC checksums (strong but slow) or XOR checksums (weak but fast). The communication kernel is built out of non-blocking point-to-point calls to load up the interconnect. The program is not designed to exhaustively test out different MPI primitives. Performance metrics are displayed, but should be carefully

5.15 High Performance Linpack (HPL)

This is a standard benchmark for Floating Point Linear Algebra performance. Two versions (1.0a and 2.0) are provided by Intel and both work identically. Included in the HPL is the Dr K. Goto Linear Algebra library. If desired, the user can modify the HPL makefiles to use alternate libraries. Atlas source code and the open source math library is also provided in `/opt/iba/src/mpi_apps/ATLAS`.

Note: The Linear Algebra Library is highly optimized for a given CPU model. When running in a fabric with mixed CPU models, the HPL application will need to be rebuilt for each CPU model and that version will need to be used on all CPUs of the given type. Attempting to run a CPU with a library which is not optimized for the given CPU could result in non-optimal performance. In some cases (such as trying to run an AMD CPU optimized library on an Intel CPU) HPL may fail or produce incorrect results.

HPL is known to scale very well and is the benchmark of choice for identifying a systems ranking in the Top 500 supercomputers (<http://www.top500.org>).

Prior to running this application, a `HPL.dat` file must be installed into `/opt/iba/src/mpi_apps/hpl/bin/ICS/HPL.dat` on all nodes. The `config_hpl` script and some sample configurations are included.



The `config_hpl` script can select from one of the assorted HPL.dat files in `hpl-config`. For assorted cluster sizes (by number of CPUs). Assorted sample HPL.dat files are provided in `/opt/iba/src/mpi_apps/hpl-config`. These files are a good starting point for most clusters and should get within 10-20% of the optimal performance for the cluster. The problem sizes used assume a cluster with 1GB of physical memory per processor (for example, for a 2 processor node, 2GB of node memory is assumed). For each cluster size, 4 files are provided:

- t — a very small test run (5000 problem size)
- s — a small problem size on the low end of optimal problem sizes
- m — a medium problem size
- l — a large problem size

These can be selected using `config_hpl`. The following command displays the preconfigured problem sizes available:

```
./config_hpl
```

For example, to quickly confirm that HPL will run on the 16 nodes in the `/opt/iba/src/mpi_apps/mpi_hosts` file:

```
Type ./config_hpl 16t.
```

This will edit the HPL.dat file on the local host for a 16 host “very small” test, and copy that HPL.dat file to all hosts in the `mpi_hosts` file.

Once the HPL.dat has been configured and copied, HPL can be run using the script:

1. Type `cd /opt/iba/src/mpi_apps`
2. Type `./run_hpl NP`

Where:

NP is the number of processors for the run or all. For example:

```
./run_hpl 16
```

For more information about HPL, consult the README, TUNING and assorted HTML files in `/opt/iba/src/mpi_apps/hpl`.

5.16 Pallas

The Pallas benchmark (PMB) does exhaustive benchmarking of latency and bandwidth for assorted message sizes for many MPI primitives. This benchmark is a good tool to evaluate and tune small clusters or a subset of a large cluster.

Pallas has known scalability limitations, especially in its **AllToAll** phase. This phase can simultaneously perform up to 4MB transfers to-and-from all nodes at once. The downside is a system must have approximately $10 \times NP$ MB of memory available per process for Pallas data to run this benchmark. Therefore, for a small cluster (approximately 16 processors or less), it is modest at 160MB. However, for a larger cluster (approximately 256 processors or greater), it is rather large at 2.5GB.

As such, it is recommended that Pallas be used for smaller runs (2-32 processes) or that it be recognized that the benchmark is likely to fail (or swap Linux to death) at larger process counts. Depending upon the amount of memory in the system and the numbers of processes to run, the `VIADEV_MEM_REG_MAX` parameter in `/opt/iba/src/mpi_apps/mpi.param.pallas` may need to be edited.

To run pallas:

1. `cd /opt/iba/src/mpi_apps`
2. `./run_pmb NP`



Where:

NP is the number of processes to run or `all`, such as:

```
./run_pmb 4
```

5.17 Intel MPI Benchmark

The `run_imb` sample script has been added in `/opt/iba/src/mpi_apps`. This script can be used to run the Intel MPI Benchmark suite (IMB) application which is included in the `mpitests` rpm. The IMB is a newer version of the Pallas benchmark (PMB) which is also included in FastFabric and can be run via the existing `run_pmb` script.

1. `cd /opt/iba/src/mpi_apps`
2. `./run_imb NP`

Where:

NP is the number of processes to run or `all`, such as: A minimum of two processes is required.

```
./run_imb 4
```

5.18 MPI Fabric Stress Test

These sample applications are designed to stress parts of an True Scale Fabric, to help ensure the fabric is working properly. Although they report measurement data similar to other bandwidth applications, they are not intended to be benchmarking tools. Instead, they should be used to identify potential performance issues in the fabric, such as bad cables.

5.18.1 All HCA latency

The All HCA latency test is a specialized stress test for large fabrics. It iterates through every possible pairing of the HCAs in the fabric and performs a latency test on each pair. At the end of each combination, it reports the fastest and slowest pairs. This test has no real value as a performance benchmark but is extremely useful for checking for cabling problems in the fabric. A script is provided to run this application. It requires no arguments, but can take several options if needed. To run with no arguments, follow these steps:

1. Change directory to `/opt/iba/src/mpi_apps`.

```
cd /opt/iba/src/mpi_apps
```

2. Run the All HCA latency test

```
./run_allhcalatency
```

This test will run a 60 second test on the first two nodes listed in the `mpi_hosts` file.

To change the default behavior, specify up to three optional arguments, for example:

```
./run_allhcalatency NP MN SS
```

Where:

NP is the number of processes to run or `all`.

MN is the number of minutes the test should run.

SS is the size of the messages to use when testing (between 1 byte and 4 megabytes).

For example, to run a 30 minute test on 64 nodes with 4 kilobyte messages the following command would be used from the `/opt/iba/src/mpi_apps` directory:



```
./run_allhcalatency 64 30 4096
```

Once 30 minutes has elapsed, the test will complete as soon as the current round of testing has completed.

If you want the tests to repeat indefinitely, `infinite` is used as the duration, as shown in the following CLI command that would be used from the `/opt/iba/src/mpi_apps` directory:

```
./run_allhcalatency 64 infinite 4096
```

There are three options, `-c`, `-h` and `-v`, available:

- `-h / --help` – provides some help text then terminates.
- `-c / --csv` – prints all raw test results in CSV file format, into the application logfile. Useful for analyzing the raw results with a spreadsheet application.
- `-v / --verbose` – runs the test in a verbose mode that shows more information.

To use the results of this test, look for nodes that are often listed as the slowest at the end of the round. One of those nodes may have a cabling problem or there may be a congested interswitch link causing those nodes to experience degraded performance.

5.18.2 run cabletest

The `run cabletest` is a specialized stress test for large fabrics. It groups MPI ranks into sets which are tested against other members of the set. This has no real value as a performance benchmark but is extremely useful for checking for cabling problems in the fabric.

`./run_cabletest` requires no arguments, but does require the user to generate a `group_hosts` file. This is done with the `gen_group_hosts` script. The name of the group hosts file is specified by the `$MPI_GROUP_HOSTS` variable, and defaults to `mpi_group_hosts` for more information on `gen_group_hosts` refer to [“gen_group_hosts” on page 310](#).

By default, `run_cabletest` will run for 60 minutes and uses 4-megabyte messages, but these settings can be changed by using the three optional arguments: duration, smallest message size, and largest message size. The arguments are specified in order:

1. Change directory to `/opt/iba/src/mpi_apps`.

```
cd /opt/iba/src/mpi_apps
```

2. Run the `run_cabletest` test including the duration in minutes, the smallest message size, and the largest message size

```
./run_cabletest dd ss ll
```

Where:

- `dd` is the duration in minutes,
- `ss` is the smallest message size
- `ll` is the largest message size.

For example, to run a one minute test, with 4 megabyte messages you would enter the following CLI command:

```
./run_cabletest 1
```

Once one minute has elapsed, the test will complete when the current round of testing completes.

If you want the tests to repeat indefinitely, you would use `infinite` as the duration, as shown in the following CLI command:



```
./run_cabletest infinite
```

In addition to the duration, you can specify the smallest and largest messages to send. The messages must be between 16384 and 4194304 (4-megabytes). This example will test message sizes between 1- and 4-megabytes, and will run for 24-hours:

```
./run_cabletest 1440 1048576 4194304
```

There are two options, `-h` and `-v`, available:

- `-h / --help` – provides this help text.
- `-v / --verbose` – runs the test in a verbose mode that shows you how the nodes were grouped.

5.18.3 run batch cabletest

The `run batch cabletest` in `/opt/iba/src/mpi_apps` makes it easier to run the `run_cabletest stress test` (see [run cabletest](#)). The `run_batch_cabletest` script runs separate jobs for each `BATCH_SIZE` hosts and can generate the `mpi_group_hosts` files needed using a single `mpi_hosts` file which lists each host to test once in topology order. For many clusters, `iba_sorthosts` may help put a list of hosts in topology order or `iba_findgood` may be used to identify candidate hosts. By using many small jobs the impact of any individual host issues (host crash, hang, etc) during the test is limited to one batch of hosts.

Note: When using `run_batch_cabletest`, the log files are now separated with each individual job getting its own log file with a suffix to the log filename indicating the run number within the set of batches. Such as: `cabletest.04Jan12165901.1`
`cabletest.04Jan12165901.2` This avoids any previous intermingling of output from multiple runs into a single log file.

By default, `run_batch_cabletest` will run for 60 minutes and uses 4-megabyte messages, but these settings can be changed by using the three optional arguments: duration, smallest message size, and largest message size. The arguments are specified in order:

1. Change directory to `/opt/iba/src/mpi_apps`.

```
cd /opt/iba/src/mpi_apps
```

2. Run the `run_batch_cabletest` test including the duration in minutes, the smallest message size, and the largest message size

```
./run_batch_cabletest [duration [minmsg [maxmsg]]]
```

Where:

- `duration` is the duration in minutes and can be `infinite`
- `minmsg` is the smallest message size. Must be between 16384 and 4194304
- `maxmsg` is the largest message size. Must be between 16384 and 4194304.

This will build a set of `mpi_hosts.#` and `mpi_group_hosts.#` files with no more than `BATCH_SIZE` hosts each. If an odd number of hosts appear in `mpi_hosts`, the last one is skipped

For example, to run a one minute batch test, with 4 megabyte messages you would enter the following CLI command:

```
./run_batch_cabletest 1
```

Once one minute has elapsed, the batch test will complete when the current round of testing completes.



If you want the tests to repeat indefinitely, you would use `infinite` as the duration, as shown in the following CLI command:

```
./run_batch_cabletest infinite
```

In addition to the duration, you can specify the smallest and largest messages to send. The messages must be between 16384 and 4194304 (4-megabytes). This example will batch test message sizes between 1- and 4-megabytes, and will run for 24-hours:

```
./run_batch_cabletest 1440 1048576 4194304
```

The following options are available:

- h / --help - provides this help text.
- v / --verbose - runs the test in a verbose mode that shows you how the nodes were grouped.
- n - specifies the number of processes to run per host.
- duration* - how many minutes to run. Can be 'infinite'. Default is 60
- minmsg* - smallest message to use. Must be between 16384 and 4194304.
- maxmsg* - largest message to use. Must be between 16384 and 4194304.
- Default minmsg and maxmsg is 4 Megabytes

This will build a set of `mpi_hosts.#` and `mpi_group_hosts.#` files with no more than `BATCH_SIZE` hosts each. If an odd number of hosts appear in `mpi_hosts`, the last one is skipped.

Each `run_cabletest` MPI job will have its output saved to a corresponding `/tmp/nohup.#.out` file

5.18.3.1 Environment

MPI_HOSTS - `mpi_hosts` file to use, the default is `mpi_hosts`. This should list the hosts in topology order, one entry per host. The hosts will be paired sequentially (first and second, third and fourth, and so on).

BATCH_SIZE - The maximum hosts per MPI job, The default is 18, and the number must be even.

5.18.3.2 Examples

```
./run_batch_cabletest  
  
MPI_HOSTS=good ./run_batch_cabletest 1440  
  
BATCH_SIZE=16 MPI_HOSTS=good ./run_batch_cabletest infinite
```

5.18.4 gen_group_hosts

This tool generates an `mpi_group_test` file for use with `run_cabletest`. The `gen_group_hosts` tool asks three questions that need to be answered in order for it to generate the `mpi_group_hosts` file.

The first question asks for the name of your hosts file. The hosts must be listed in group order and one hosts per line, The hosts should not be listed more than once and should be listed in their physical order. The default hosts file is `/opt/iba/src/mpi_apps/mpi_hosts`.



The second question asks how big your groups are. For example, if you want to test each node against the node next to it, you would use 2 as the group size. If you want to test the nodes connected to one leaf switch against the nodes on another leaf switch, and you have 16 nodes per leaf, you would use 32 as the group size. The default group size is 2.

The third question asks how many processes you want to run per node. The higher the number the higher the link utilization will be. The number should be between 1 and the number of processors per node. The default number of processes per node is 3. Using more processes than needed to saturate the link will not improve testing.

Once the above questions are answered the `/opt/iba/src/mpi_apps/mpi_group_hosts` file is generated.

If the number of the hosts is not a multiple of the group size, a warning will be shown.

5.18.5 **run_multibw**

`run_multibw` runs `mpi_multibw`, which performs a multi-core pairwise bandwidth test. `mpi_multibw` is based on OSU bw and multi-lat.

1. Change directory to `/opt/iba/src/mpi_apps`.

```
cd /opt/iba/src/mpi_apps
```
2. Run the `run_multibw` test including the number of processes on which to run the test

```
./run_multibw processes
```

Where `processes` is the number of processes on which to run the test ('all' indicates the test should be run for every process in the `mpi_hosts` file).

5.18.6 **run_nxnlatbw**

`run_nxnlatbw` runs `mpi_nxnlatbw`, which is an NxN latency bandwidth test.

1. Change directory to `/opt/iba/src/mpi_apps`.

```
cd /opt/iba/src/mpi_apps
```
2. Run the `run_nxnlatbw` test including the number of processes on which to run the test

```
./run_nxnlatbw processes
```

Where `processes` is the number of processes on which to run the test ('all' indicates the test should be run for every process in the `mpi_hosts` file).

5.19 **MPI Batch `run_*` scripts**

The `run_batch_script`, in `/opt/iba/src/mpi_apps` makes it easier to run other `run_*` scripts as many smaller jobs. This script runs separate jobs for each `BATCH_SIZE` hosts. By using many small jobs the impact of any individual host issues (host crash, hang, etc) during the test is limited to one batch of hosts.

Note: When using `run_batch_script`, the log files are now separated with each individual job getting its own log file with a suffix to the log filename indicating the run number within the set of batches. Such as: `mpi_groupstress.04Jan12165901.1`
`mpi_groupstress.04Jan12165901.2` This avoids any previous intermingling of output from multiple runs into a single log file.



5.19.0.1 Usage

```
./run_batch_script [-e] run_script [args]
```

or

```
./run_batch_script --help
```

5.19.0.2 Options

-e – force an even number of hosts in final batch by skipping the last one.

run_script – a run_* script from this directory

args – arguments for the run_script. if the first argument is NP it will be replaced with the process count

This will build a set of mpi_hosts.# files with no more than BATCH_SIZE hosts each. If -e is specified and an odd number of hosts appear in mpi_hosts, the last one is skipped. Each run_script MPI job will have its output saved to a corresponding ./tmp/nohup.#.out file

This script is only used for scripts which use MPI_HOSTS.

To run run_cabletest use run_batch_cabletest.

5.19.0.3 Environment

MPI_HOSTS - mpi_hosts file to use, default is mpi_hosts

BATCH_SIZE - max hosts per MPI job, The default is 18, if -e must be even

MIN_BATCH_SIZE - min hosts per MPI job, default is 2, if -e must be even

The following Environment variables are supported in individual run_* scripts:

SHOW_MPI_HOSTS - set to "y" if MPI_HOSTS contents should be output prior to starting job

SHOW_MPI_HOSTS_LINES - set to the maximum number of lines in

5.19.0.4 Examples

```
./run_batch_script run_deviation NP ff
```

```
BATCH_SIZE=2 MPI_HOSTS=good ./run_batch_script run_lat2
```

```
BATCH_SIZE=16 MPI_HOSTS=good ./run_batch_script run_deviation ff
```

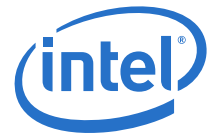
```
MIN_BATCH_SIZE=16 BATCH_SIZE=16 ./run_batch_script run_hpl2 16
```

5.19.1 SHMEM Batch run_* scripts

Scripts for various SHMEM benchmarks included with SHMEM are contained in /opt/iba/src/shmem_apps. The behavior of these scripts is very similar to those in mpi_apps.

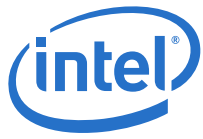
Each SHMEM application/benchmark has an accompanying run_* script, which assumes the existence of a local mpi_hosts file. The provided run_* scripts include the following:

```
run_alltoall
```

```
run_barrier
run_reduce
run_get[put]_bw
run_get[put]_bibw
```

§ §





Appendix A Port Counters Overview

Each Port in an True Scale Fabric maintains a set of Port Counters to indicate both traffic and error counts. These counters can be grouped into a few categories outlined below.

A.1 Link Integrity

These counters reflect link errors and the Physical (Phy) and Link Layers. The typical cause is a hardware problem such as a poor connection, marginal cable, incorrect length/model cable for signal rate (DDR, SDR, etc) or damaged/broken hardware (bad connectors, etc). It should be noted that IBTA standard defines an acceptable bit error rate as 10^{-12} , hence no link is guaranteed to be totally error free. Intel strives for a 10^{-15} -bit error rate in our designs and internal testing such that manufacturing tolerances will still place the error rate much better than 10^{-12} in the field. In practice, when using Intel products and qualified cables, links observe much better signal quality and often observe less than one error per day.

When a bad packet is detected, one of these counters will be incremented and the Link Layer will discard the packet. In the face of lost packets, transport protocols such as the RC transport layer will detect the lost packet and request a retransmission by the remote endpoint. Hence lost packets do not cause application failure and will only cause a minor latency hit for the given message.

During the link training sequence, assorted errors may be observed (especially Symbol Errors). This is a normal part of the link training and clock synchronization process. Hence errors observed as part of rebooting nodes or moving cables should not be considered a problem.

With the exception of ExcessiveBufferOverrunErrors, the counters below are all reported on the receive side of the link. However they could indicate a problem on either side of the link.

A.2 SymbolError Counter

Total number of minor link errors detected on one or more physical lanes.

IBTA-compliant fabrics use an 8B10B encoding mechanism (for example, 10-bits on the wire are used to represent an 8-bit byte). As such there are quite a few 10B (10-bit) values which are invalid. In most cases a single bit flip in a byte will be discovered at the serdes level of the Phy/Link Layer and will result in a Symbol Error. Symbol Errors are typically the first indication of a link quality problem and focus is on this parameter first, as an indication of link quality.

The Symbol Error counter is 16-bits, so the value will stop increasing once 65535 is reached.

A.2.1 LinkErrorRecovery Counter

Total number of times the Port Training state machine has successfully completed the link error recovery process.

If a link is having a significant number of problems, the link will retrain through the Link Error Recovery process. Such retraining can be done without SM intervention. Link Error Recovery is typically caused by link integrity problems. However in rare cases it can be caused by incorrect SM or link configuration (such as inconsistent number of VLS or MTU configured for each side of a link).



It is the responsibility of the SM to avoid such inconsistencies, however if user's manually override the SM's configuration of an Active link (using tools such as `iba_portconfig`) errors can be observed until the SM discovers and corrects the configuration inconsistency. When changing configuration of a link it is recommended to Down the link (which `iba_portconfig` will do by default) so the SM can reconfigure both sides of the link consistently.

The LinkErrorRecovery counter is 8 bits, so the value will stop increasing once 255 is reached.

A.2.2 LinkDowned Counter

Total number of times the Port Training state machine has failed the link error recovery process and downed the link.

If link Error Recovery fails, the Link State machine will take the Link Down and restart the link training sequence from scratch. If the link training is successful, the SM will need to re initialize the link to return it to the Active state and bring it back into normal use. As LinkDowned errors can cause disruptions to fabric traffic that can last for a longer period of time (milliseconds or seconds).

As for other errors, LinkDowned can also happen during normal link training and is part of the link speed negotiation process.

The LinkDowned counter is 8-bits, so the value will stop increasing once 255 is reached.

A.2.3 PortRcvErrors

Total number of packets containing an error that were received on the port. These errors include:

- Local physical errors (ICRC, VCRC, LPCRC, and all physical errors that cause entry into the BAD PACKET or BAD PACKET DISCARD states of the packet receiver state machine)
- Malformed data packet errors (LVer, length, VL)
- Malformed link packet errors (operand, length, VL)
- Packets discarded due to buffer overrun

This error will catch multiple bit flips in a given packet and other causes for CRC errors. In addition to the 8B10B encoding mechanism (discussed in SymbolErrors above), the IBTA standard maintains two CRCs per packet. These CRCs provide stronger detection of multiple bit flips and other forms of packet corruption.

The typical cause for PortRcvErrors are CRC errors and indicate a link integrity problem. However in rare cases it can be caused by incorrect SM or link configuration (such as inconsistent number of VLS or MTU configured for each side of a link).

It is the responsibility of the SM to avoid such inconsistencies, however if user's manually override the SM's configuration of an Active link (using tools such as `iba_portconfig`) errors can be observed until the SM discovers and corrects the configuration inconsistency. Hence when changing configuration of a link it is recommended to Down the link (which `iba_portconfig` will do by default) so the SM can reconfigure both sides of the link consistently.

The PortRcvErrors counter is 16-bits, so the value will stop increasing once 65535 is reached



A.2.4 LocalLinkIntegrityErrors

The number of times that the count of local physical errors exceeded the threshold specified by LocalPhyErrors

The SM configures a value for LocalPhyErrors per link. This value is a threshold on the number of physical layer errors observed. Once this is hit, the link automatically initiates link error recovery and retrains at which point this counter is incremented.

This counter is 4 bits, so the value will stop increasing once 15 is reached

A.2.5 ExcessiveBufferOverflowErrors

The number of times that OverflowErrors consecutive flow control update periods occurred, each having at least one overflow error

This error indicates a loss of Link Layer flow control update packets. It can occur on extremely poor links (in which case the other error counters would also occur). However in rare cases it can be caused by incorrect SM or link configuration (such as inconsistent number of VLs configured for each side of a link).

It is the responsibility of the SM to avoid such inconsistencies, however if user's manually override the SM's configuration of an Active link (using tools such as `iba_portconfig`) errors can be observed until the SM discovers and corrects the configuration inconsistency. Hence when changing configuration of a link it is recommended to Down the link (which `iba_portconfig` will do by default) so the SM can reconfigure both sides of the link consistently.

This counter is 4-bits, so the value will stop increasing once 15 is reached

A.3 Sma Congestion

A.3.1 VL15Dropped

Number of incoming VL15 packets dropped due to resource limitations (for example, lack of buffers) in the port

VL15 is exclusively used for SMA packets sent by the SM to query and configure each device in the fabric. By definition VL15 is not flow controlled, so SMAs which are unable to locally buffer a given SMA packet will discard it and increment this counter.

The SM has no way to identify the buffering nor performance capabilities in the SMAs in the fabric. However to achieve the best performance for fabric discovery and programming the SM must issue more than one SMA packet at a time, especially to switches (which have many ports and attributes to be configured).

As such the Intel® SM employs a heuristic approach which issues a modest number of concurrent SMA packets to each device, especially switches. In some cases the number of concurrent packets to or through a given device may cause packet loss and this counter will increment. All SMA packets require a positive response to the SM, so if a packet is lost, the SM will notice the loss and retry it, hence causing no impact on fabric operation and only a minor hit in SM performance.

As a result, when using the Intel® SM, it can be expected to see increases in this counter per SM sweep. This is normal and is not cause for concern. If the SM has problems communicating with the fabric which retries cannot handle, the SM log will reflect such problems.

This counter is 16 bits, so the value will stop increasing once 65535 is reached



A.4 Congestion

A.4.1 PortXmitDiscards

Total number of outbound packets discarded by the port because the port is down or congested. Reasons for this include:

- Output port is not in the active state
- Packet length exceeded NeighborMTU
- Switch Lifetime Limit exceeded
- Switch HOQ Lifetime Limit exceeded

This may also include packets discarded while in VLStalled State.

If an output port cannot make reasonable progress, the packet will be discarded and this counter is incremented. The typical cause for this is in switches when a packet is queued in the switch for an amount of time exceeding the Switch HOQ Lifetime or Switch Lifetime limits. These two limits are configured by the SM (and the settings are adjusted through the SM configuration file).

Other causes for this can include packets queued in a switch when a link goes down or upstream failures such as a hung CA or server whose link remains Active but is not accepting packets.

However in rare cases it can be caused by incorrect SM or link configuration (such as inconsistent MTU used in a path through the fabric).

This counter is 16 bits, so the value will stop increasing once 65535 is reached.

A.4.2 PortXmitWait

The number of link layer “ticks” for which the transmitter had data but was unable to transmit (no credits available or link was busy sending non data packets such as link layer retraining or flow control). The definition of a tick is defined as the Xmit Wait Tick. This value is only available for Intel® True Scale HCAs.

A.4.3 PortXmitCongestion

The number of times the switch checked its queue depth and found an excessive outbound queue depth. The PortCheckRate defines the number of checks per second. This value is only available for Intel® True Scale switches.

A.5 Security

A.5.1 PortXmitConstraintErrors

Total number of packets not transmitted from the switch physical port for the following reasons:

- FilterRawOutbound is true and packet is raw
- PartitionEnforcementOutbound is true and packet fails partition key check or IP version check

The IBTA standard defined a raw packet capability which was never used by any application. Therefore, the main cause for this counter is Partitioning violations.



In IFS 4.4 and later releases Partitioning is supported as part of Virtual Fabrics in which case this counter can indicate applications which are not complying with the partitioning configuration of the fabric as defined by the administrator through the SM configuration file. Such applications may have bugs, attempting to violate security or merely have incorrect configuration in the application or the SM.

This counter is 8 bits, so the value will stop increasing once 255 is reached.

A.5.2 PortRcvConstraintErrors

Total number of packets received on the switch physical port that are discarded for the following reasons:

- FilterRawInbound is true and packet is raw
- PartitionEnforcementInbound is true and packet fails partition key check or IP version check

This counter is identical in function to PortXmitConstraintErrors except it indicates invalid packets detected by the receiver rather than the transmitter.

This counter is 8 bits, so the value will stop increasing once 255 is reached.

A.6 Routing

A.6.1 PortRcvSwitchRelayErrors

Total number of packets received on the port that were discarded because they could not be forwarded by the switch relay.

Reasons for this include:

- DLID mapping
- VL mapping
- Looping (output port = input port)

This counter is intended to indicate packets which could not be routed. Typical causes would be applications sending packets to non-existent DLIDs or SM bugs which caused looped packets or use of invalid VLs. This counter can also be incremented if there are packets in the fabric while the SM is reconfiguring routing, such as when the SM is removing an offline node from the fabric.

This counter is 16 bits, so the value will stop increasing once 65535 is reached.

A.6.2 Data Movement

These counters are optional and reflect traffic movement through a given port.

These counters are all 32- or 64-bits and hence have a raw limit of 4294967295 or 1.8×10^{19} . However various tools may present these values in other formats (such as MB/s) which may have lower limits.

Beware that at full DDR speeds, 32-bit counters could saturate in 10 seconds or less (5 seconds or less at QDR speeds). 64-bit versions of these counters are available in all Intel® True Scale HCAs and Switches. 64-bit counters will take over 100 years before they will overflow at full QDR speeds.



A.6.3 PortXmitData

Optional; shall be zero if not implemented. Total number of data octets, divided by 4, transmitted on all VLs from the port. This includes all octets between (and not including) the start of packet delimiter and the VCRC, and may include packets containing errors. It excludes all link packets.

Implementers may choose to count data octets in groups larger than four but are encouraged to choose the smallest group possible.

Results are still reported as a multiple of four octets.

All Intel® True Scale HCAs and Switches support this counter. The Intel® True Scale products support the 64-bit option.

A.6.4 PortRcvData

Optional; shall be zero if not implemented. Total number of data octets, divided by 4, received on all VLs at the port. This includes all octets between (and not including) the start of packet delimiter and the VCRC, and may include packets containing errors. It excludes all link packets.

When the received packet length exceeds the maximum allowed packet length specified, the counter may include all data octets exceeding this limit.

Implementers may choose to count data octets in groups larger than four but are encouraged to choose the smallest group possible.

Results are still reported as a multiple of four octets.

All Intel® True Scale HCAs and Switches support this counter. The Intel® True Scale products support the 64-bit option.

A.6.5 PortXmitPkts

Optional; shall be zero if not implemented. Total number of packets transmitted on all VLs from the port. This may include packets with errors, and excludes link packets.

All Intel® True Scale HCAs and Switches support this counter. The Intel® True Scale products support the 64-bit option.

A.6.6 PortRcvPkts

Optional; shall be zero if not implemented. Total number of packets, including packets containing, and excluding link packets, received from all VLs on the port.

All Intel® True Scale HCAs and Switches support this counter. The Intel® True Scale products support the 64-bit option.

A.6.7 PortXmitWait

If ClassPortInfo:CapabilityMask.PortCountersXmitWaitSupported is set to 1, the number of ticks during which the port selected by PortSelect had data to transmit but no data was sent during the entire tick either because of insufficient credits or because of lack of arbitration. Otherwise, undefined.

Unfortunately, while the IBTA standard defines this counter, it defines no way to clear it. As such, the sampling mechanism is the only way to access it, which greatly limits its usefulness.



To overcome this limitation, Intel® True Scale HCAs and Switches support the PortXmitWait and PortXmitCongestion counters discussed in the Congestion section above. Those counters are implemented in a manner which permits scalable monitoring and clearing of the values. Permitting the Intel® FM and its PM/PA to utilize them for advanced congestion analysis.

A.7 Other

A.7.1 PortRcvRemotePhysicalErrors

Total number of packets marked with the EBP delimiter received on the port.

IBTA Architecture was designed for low latency. As such cut-through switching is an integral part of the spec. If a switch supporting cut-through detects an error in a packet which it has already begun to transmit, it will continue transmitting the packet but will mark it with a bad CRC and an EBP delimiter at the end. Each subsequent port along the path which receives the packet will increment the PortRcvRemotePhysicalErrors counter. End nodes, such as CAs, will discard the packet. However switches doing cut-through will not see the EBP delimiter and bad CRC till the end of the packet so they will end up passing along the bad packet with the EBP delimiter.

Because this counter indicates a “victim” of an error earlier in the path, use of this counter is of limited value. At best it can indicate the amount of wasted bandwidth or which ports are being victims of errors elsewhere in the fabric.

Due to the limited value of this counter and difficulty associating it with the exact location of the faulty link, it's recommended to ignore this counter and instead focus on the counters in the “Integrity” section to identify the faulty link.

This counter is 16 bits, so the value will stop increasing once 65535 is reached

