39XX/51XX Switches and Platforms

YANG Model Reference

SAOS 6.21.2

What's inside...

New in this release
Introduction
Ciena YANG model schemas
Standard YANG model schemas

009-3458-047 - Revision A
March 2022

Copyright© 2022 Ciena® Corporation. All rights reserved.

LEGAL NOTICES

THIS DOCUMENT CONTAINS CONFIDENTIAL AND TRADE SECRET INFORMATION OF CIENA CORPORATION AND ITS RECEIPT OR POSSESSION DOES NOT CONVEY ANY RIGHTS TO REPRODUCE OR DISCLOSE ITS CONTENTS, OR TO MANUFACTURE, USE, OR SELL ANYTHING THAT IT MAY DESCRIBE. REPRODUCTION, DISCLOSURE, OR USE IN WHOLE OR IN PART WITHOUT THE SPECIFIC WRITTEN AUTHORIZATION OF CIENA CORPORATION IS STRICTLY FORBIDDEN.

EVERY EFFORT HAS BEEN MADE TO ENSURE THAT THE INFORMATION IN THIS DOCUMENT IS COMPLETE AND ACCURATE AT THE TIME OF PUBLISHING; HOWEVER, THE INFORMATION CONTAINED IN THIS DOCUMENT IS SUBJECT TO CHANGE.

While the information in this document is believed to be accurate and reliable, except as otherwise expressly agreed to in writing CIENA PROVIDES THIS DOCUMENT “AS IS” WITHOUT WARRANTY OR CONDITION OF ANY KIND, EITHER EXPRESS OR IMPLIED. The information and/or products described in this document are subject to change without notice. For the most up-to-date technical publications, visit www.ciena.com.

Copyright© 2022 Ciena® Corporation. All Rights Reserved

Use or disclosure of data contained in this document is subject to the Legal Notices and restrictions in this section and, unless governed by a valid license agreement signed between you and Ciena, the Licensing Agreement that follows.

The material contained in this document is also protected by copyright laws of the United States of America and other countries. It may not be reproduced or distributed in any form by any means, altered in any fashion, or stored in a data base or retrieval system, without express written permission of the Ciena Corporation.

Security

Ciena® cannot be responsible for unauthorized use of equipment and will not make allowance or credit for unauthorized use or access.

Contacting Ciena

Corporate Headquarters	410-694-5700 or 800-921-1144	www.ciena.com
Customer Technical Support/Warranty		www.ciena.com/support/
Sales and General Information	North America: 1-800-207-3714 International: +44 20 7012 5555	E-mail: sales@ciena.com
In North America	410-694-5700 or 800-207-3714	E-mail: sales@ciena.com
In Europe	+44-207-012-5500 (UK)	E-mail: sales@ciena.com
In Asia	+81-3-3248-4680 (Japan)	E-mail: sales@ciena.com
In India	+91-22-42419600	E-mail: sales@ciena.com
In Latin America	011-5255-1719-0220 (Mexico City)	E-mail: sales@ciena.com
Training		E-mail: learning@ciena.com

For additional office locations and phone numbers, please visit the Ciena web site at www.ciena.com.

READ THIS LICENSE AGREEMENT ("LICENSE") CAREFULLY BEFORE INSTALLING OR USING CIENA SOFTWARE OR DOCUMENTATION. THIS LICENSE IS AN AGREEMENT BETWEEN YOU AND CIENA COMMUNICATIONS, INC. (OR, AS APPLICABLE, SUCH OTHER CIENA CORPORATION AFFILIATE LICENSOR) ("CIENA") GOVERNING YOUR RIGHTS TO USE THE SOFTWARE. BY INSTALLING OR USING THE SOFTWARE, YOU ACKNOWLEDGE THAT YOU HAVE READ THIS LICENSE AND AGREE TO BE BOUND BY IT.

1. License Grant. Ciena may provide "Software" to you either (1) embedded within or running on a hardware product or (2) as a standalone application, and Software includes upgrades acquired by you from Ciena or a Ciena authorized reseller. Subject to these terms, and payment of all applicable License fees including any usage-based fees, Ciena grants you, as end user, a non-exclusive, non-transferable, personal License to use the Software only in object code form and only for its intended use as evidenced by the applicable product documentation. Unless the context does not permit, Software also includes associated documentation.

2. Open Source and Third Party Licenses. Software excludes any open source or third-party programs supplied by Ciena under a separate license, and you agree to be bound by the terms of any such license. If a separate license is not provided, any open source and third party programs are considered "Software" and their use governed by the terms of this License.

3. Title. You are granted no title or ownership rights in or to the Software. Unless specifically authorized by Ciena in writing, you are not authorized to create any derivative works based upon the Software. Title to the Software, including any copies or derivative works based thereon, and to all copyrights, patents, trade secrets and other intellectual property rights in or to the Software, are and shall remain the property of Ciena and/or its licensors. Ciena's licensors are third party beneficiaries of this License. Ciena reserves to itself and its licensors all rights in the Software not expressly granted to you.

4. Confidentiality. The Software contains trade secrets of Ciena. Such trade secrets include, without limitation, the design, structure and logic of individual Software programs, their interactions with other portions of the Software, internal and external interfaces, and the programming techniques employed. The Software and related technical and commercial information, and other information received in connection with the purchase and use of the Software that a reasonable person would recognize as being confidential, are all confidential information of Ciena ("Confidential Information").

5. Obligations. You shall:

i) Hold the Software and Confidential Information in strict confidence for the benefit of Ciena using your best efforts to protect the Software and Confidential Information from unauthorized disclosure or use, and treat the Software and Confidential Information with the same degree of care as you do your own similar information, but no less than reasonable care;

ii) Keep a current record of the location of each copy of the Software you make;

iii) Use the Software only in accordance with the authorized usage level;

iv) Preserve intact any copyright, trademark, logo, legend or other notice of ownership on any original or copies of the Software, and affix to each copy of the Software you make, in the same form and location, a reproduction of the copyright notices, trademarks, and all other proprietary legends and/or logos appearing on the original copy of the Software delivered to you; and

v) Issue instructions to your authorized personnel to whom Software is disclosed, advising them of the confidential nature of the Software and provide them with a summary of the requirements of this License.

6. Restrictions. You shall not:

i) Use the Software or Confidential Information a) for any purpose other than your own internal business purposes; and b) other than as expressly permitted by this License;

ii) Allow anyone other than your authorized personnel who need to use the Software in connection with your rights or obligations under this License to have access to the Software;

iii) Make any copies of the Software except such limited number of copies, in machine readable form only, as may be reasonably necessary for execution in accordance with the authorized usage level or for archival purposes only;

iv) Make any modifications, enhancements, adaptations, derivative works, or translations to or of the Software;

v) Reverse engineer, disassemble, reverse translate, decompile, or in any other manner decode the Software;

vi) Make full or partial copies of the associated documentation or other printed or machine-readable matter provided with the Software unless it was supplied by Ciena in a form intended for reproduction;

vii) Export or re-export the Software from the country in which it was received from Ciena or its authorized reseller unless authorized by Ciena in writing; or

Copyright© 2022 Ciena® Corporation

viii) Publish the results of any benchmark tests run on the Software.

7. Audit: Upon Ciena's reasonable request you shall permit Ciena to audit the use of the Software to ensure compliance with this License.

8. U.S. Government Use. The Software is provided to the Government only with restricted rights and limited rights. Use, duplication, or disclosure by the Government is subject to restrictions set forth in FAR Sections 52-227-14 and 52-227-19 or DFARS Section 52.227-7013(C)(1)(ii), as applicable. The Software and any accompanying technical data (collectively "Materials") are commercial within the meaning of applicable Federal acquisition regulations. The Materials were developed fully at private expense. U.S. Government use of the Materials is restricted by this License, and all other U.S. Government use is prohibited. In accordance with FAR 12.212 and DFAR Supplement 227.7202, the Software is commercial computer software and the use of the Software is further restricted by this License.

9. Term of License. This License is effective until the applicable subscription period expires or the License is terminated. You may terminate this License by giving written notice to Ciena. This License will terminate immediately if (i) you breach any term or condition of this License or (ii) you become insolvent, cease to carry on business in the ordinary course, have a receiver appointed, enter into liquidation or bankruptcy, or any analogous process in your home country. Termination shall be without prejudice to any other rights or remedies Ciena may have. Upon any termination of this License you shall destroy and erase all copies of the Software in your possession or control, and forward written certification to Ciena that all such copies of Software have been destroyed or erased. Your obligations to hold the Confidential Information in confidence, as provided in this License, shall survive the termination of this License.

10. Compliance with laws. You agree to comply with all laws related to your installation and use of the Software. Software is subject to U.S. export control laws, and may be subject to export or import regulations in other countries. If Ciena authorizes you to import or export the Software in writing, you shall obtain all necessary licenses or permits and comply with all applicable laws.

11. Limitation of Liability. ANY LIABILITY OF CIENA SHALL BE LIMITED IN THE AGGREGATE TO THE AMOUNTS PAID BY YOU TO CIENA OR ITS AUTHORIZED RESELLER FOR THE SOFTWARE. THIS LIMITATION APPLIES TO ALL CAUSES OF ACTION, INCLUDING WITHOUT LIMITATION BREACH OF CONTRACT, BREACH OF WARRANTY, NEGLIGENCE, STRICT LIABILITY, MISREPRESENTATION AND OTHER TORTS. THE LIMITATIONS OF LIABILITY DESCRIBED IN THIS SECTION ALSO APPLY TO ANY LICENSOR OF CIENA. NEITHER CIENA NOR ANY OF ITS LICENSORS SHALL BE LIABLE FOR ANY INJURY, LOSS OR DAMAGE, WHETHER INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL INCLUDING WITHOUT LIMITATION ANY LOST PROFITS, CONTRACTS, DATA OR PROGRAMS, AND THE COST OF RECOVERING SUCH DATA OR PROGRAMS, EVEN IF INFORMED OF THE POSSIBILITY OF SUCH DAMAGES IN ADVANCE.

12. General. Ciena may assign this License to an affiliate or to a purchaser of the intellectual property rights in the Software. You shall not assign or transfer this License or any rights hereunder, and any attempt to do so will be void. This License shall be governed by the laws of the State of New York without regard to conflict of laws provisions. The U.N. Convention on Contracts for the International Sale of Goods shall not apply hereto. This License constitutes the complete and exclusive agreement between the parties relating to the license for the Software and supersedes all proposals, communications, purchase orders, and prior agreements, verbal or written, between the parties. If any portion hereof is found to be void or unenforceable, the remaining provisions shall remain in full force and effect.

About this document vii

Documents in the 39XX/51XX documentation suite 2-vii

New in this release 1-1

Introduction 2-1

Ciena YANG model schemas 3-1

ciena-access-flow-deviation-sds.yang 3-1
ciena-access-flow-sds.yang 3-4
ciena-alarm.yang 3-10
ciena-bw-calculation-mode.yang 3-25
ciena-cfm-sds.yang 3-27
ciena-chassis-sds.yang 3-36
ciena-classifier.yang 3-58
ciena-command-file.yang 3-61
ciena-command-log.yang 3-63
ciena-configuration-management.yang 3-69
ciena-cos-to-frame-map.yang 3-79
ciena-cpu-rate-limit-sds.yang 3-81
ciena-cpu-sds.yang 3-90
ciena-dhcp.yang 3-92
ciena-dhcpv6.yang 3-108
ciena-dns-client.yang 3-120
ciena-dot1x.yang 3-125
ciena-egress-qos.yang 3-127
ciena-eoam.yang 3-131
ciena-ethernet-port-deviation.yang 3-133
ciena-ethernet-port.yang 3-134
ciena-event.yang 3-137
ciena-ext-lag.yang 3-141
ciena-fd-sds.yang 3-154
ciena-fd.yang 3-156
ciena-file-transfer-types.yang 3-159
ciena-flood-containment-profile.yang 3-162
ciena-fp-sds.yang 3-164
ciena-g8032.yang 3-167
ciena-icl.yang 3-170
ciena-ieee-ethernet-port.yang 3-176
ciena-ieee-lag-deviation.yang 3-180
ciena-ieee-lag.yang 3-184
ciena-ieee-lldp-deviation.yang 3-205
ciena-ieee-lldp.yang 3-206
ciena-ieee-mstp.yang 3-229
ciena-ieee-port-dot1x-deviation.yang 3-231
ciena-ieee-port-dot1x.yang 3-232
ciena-ieee-port-eoam-deviation.yang 3-241
ciena-ieee-port-eoam.yang 3-243
ciena-ieee-rstp.yang 3-270

Copyright© 2022 Ciena® Corporation

ciena-ietf-igmp-deviation.yang 3-272
ciena-ietf-igmp.yang 3-273
ciena-ietf-ip-deviation-sds.yang 3-292
ciena-ietf-snmp-deviation.yang 3-294
ciena-ietf-syslog-deviation.yang 3-295
ciena-ietf-system-deviation.yang 3-297
ciena-ietf-twamp-deviation.yang 3-299
ciena-ietf-twamp.yang 3-301
ciena-ip-acl-sds.yang 3-312
ciena-itu-g8032-deviation.yang 3-318
ciena-itu-g8032.yang 3-321
ciena-l2cp-profile.yang 3-340
ciena-log-sds.yang 3-345
ciena-logical-port.yang 3-360
ciena-mac-management-deviation-sds.yang 3-365
ciena-management-interfaces-sds.yang 3-366
ciena-mef-access-flow.yang 3-388
ciena-mef-classifier-deviation.yang 3-395
ciena-mef-classifier.yang 3-396
ciena-mef-cos-to-frame-map-deviation.yang 3-405
ciena-mef-cos-to-frame-map.yang 3-406
ciena-mef-egress-qos-deviation.yang 3-408
ciena-mef-egress-qos.yang 3-410
ciena-mef-fd-deviation.yang 3-420
ciena-mef-fd.yang 3-421
ciena-mef-flood-containment-profile-deviation.yang 3-424
ciena-mef-flood-containment-profile.yang 3-425
ciena-mef-fp-deviation.yang 3-427
ciena-mef-fp.yang 3-430
ciena-mef-frame-to-cos-map-deviation.yang 3-439
ciena-mef-frame-to-cos-map.yang 3-440
ciena-mef-l2cp-profile-deviation.yang 3-443
ciena-mef-l2cp-profile.yang 3-444
ciena-mef-logical-port-deviation.yang 3-447
ciena-mef-logical-port.yang 3-448
ciena-mef-mac-management.yang 3-453
ciena-mef-meter-profile-deviation.yang 3-461
ciena-mef-meter-profile.yang 3-462
ciena-mef-pfg-profile.yang 3-464
ciena-mef-qos-flow-deviation.yang 3-467
ciena-mef-qos-flow.yang 3-468
ciena-netch-yuma-api.yang 3-473
ciena-ntp.yang 3-475
ciena-ocsp.yang 3-490
ciena-perfmon.yang 3-492
ciena-pfg-profile-deviation-sds.yang 3-504
ciena-pfg-profile-sds.yang 3-505
ciena-pkix-types.yang 3-508
ciena-pkix.yang 3-513
ciena-qos-flow.yang 3-517

Copyright© 2022 Ciena® Corporation

ciena-radius.yang 3-522
ciena-removable-storage.yang 3-533
ciena-reserved-vlan-sds.yang 3-538
ciena-resource-manager-sds.yang 3-539
ciena-resource-manager.yang 3-541
ciena-rmon.yang 3-544
ciena-sat-deviation.yang 3-560
ciena-sat.yang 3-562
ciena-security-certs.yang 3-628
ciena-security-log.yang 3-635
ciena-snmp.yang 3-643
ciena-soam-pm-sds.yang 3-650
ciena-software-licensing.yang 3-666
ciena-software-mgmt.yang 3-671
ciena-ssh-types.yang 3-683
ciena-ssh.yang 3-688
ciena-syslog.yang 3-700
ciena-system.yang 3-702
ciena-tacacs.yang 3-706
ciena-telnet.yang 3-715
ciena-test-1.yang 3-717
ciena-test-rollback.yang 3-724
ciena-test-types-deviation.yang 3-725
ciena-test-types.yang 3-726
ciena-time.yang 3-745
ciena-twamp.yang 3-748
ciena-types.yang 3-751
ciena-users.yang 3-763
ciena-vlli.yang 3-773
ciena-xftp.yang 3-783

Standard YANG model schemas 4-1

iana-crypt-hash@2014-08-06.yang 4-1
iana-if-type.yang 4-4
ieee-dot1q-types.yang 4-30
ieee-port-lldp.yang 4-42
ieee-types.yang 4-46
ietf-inet-types.yang 4-48
ietf-inet-types@2010-09-24.yang 4-55
ietf-interfaces.yang 4-63
ietf-ip.yang 4-76
ietf-netconf-acm@2012-02-22.yang 4-89
ietf-netconf-monitoring.yang 4-97
ietf-snmp-common.yang 4-107
ietf-snmp-community.yang 4-111
ietf-snmp-engine.yang 4-116
ietf-snmp-notification.yang 4-119
ietf-snmp-proxy.yang 4-123
ietf-snmp-ssh.yang 4-126
ietf-snmp-target.yang 4-129

Copyright© 2022 Ciena® Corporation

ietf-snmp-tls.yang 4-133
ietf-snmp-tsm.yang 4-137
ietf-snmp-usm.yang 4-139
ietf-snmp-vacm.yang 4-143
ietf-snmp.yang 4-148
ietf-syslog-types.yang 4-150
ietf-syslog.yang 4-155
ietf-system.yang 4-167
ietf-x509-cert-to-name.yang 4-182
ietf-yang-smiv2.yang 4-188
ietf-yang-types.yang 4-191
ietf-yang-types@2010-09-24.yang 4-199
mef-cfm-deviation.yang 4-206
mef-cfm.yang 4-210
mef-soam-fm-deviation.yang 4-247
mef-soam-fm.yang 4-250
mef-soam-pm-deviation.yang 4-279
mef-soam-pm-sds-deviation.yang 4-299
mef-soam-pm.yang 4-301

Copyright© 2022 Ciena® Corporation

Publication history

March 2022

Standard Revision A

First standard release of this document for SAOS 6.21.2.

Copyright© 2022 Ciena® Corporation

Copyright© 2022 Ciena® Corporation

About this document

This document describes the YANG model schemas for 39XX/51XX switches. YANG is the data modeling language used by Network Configuration Protocol (NETCONF).NETCONF provides a way to install, manipulate and delete the configuration of a 39XX/51XX switch. NETCONF is discussed in *39XX/51XX Switches and Platforms Base Configuration.*

Documents in the 39XX/51XX documentation suite

For descriptions of documents in the 39XX/51XX *Switches and Platforms* documentation suite, see *39XX/51XX Switches and Platforms Product Fundamentals*.

Copyright© 2022 Ciena® Corporation

Copyright© 2022 Ciena® Corporation

New in this release

This document reflects the YANG model schemas used in SAOS 6.21.2.

Copyright© 2022 Ciena® Corporation

Copyright© 2022 Ciena® Corporation

Introduction

YANG models are divided into two main groups:

- standardized YANG models, which conform to standards set by regulating bodies. Standard YANG models are:
  - — ietf-interfaces, which conforms to RFC 7223. For details about ietf-interfaces, refer to www.ietf.org.
  - — mef-cfm, which conforms to MEF 38. For details about mef-cfm, refer to www.mef.net. Note that 39XX/51XX switches do not fully support mef-cfm.
- pre-standards YANG models, which were developed before YANG models were standardized

  Note: Some YANG models differ significantly from other SAOS management interfaces. There is not always a one-to-one correspondence between CLI commands and YANG models.

  Table 2-1 lists the pre-standards YANG models supported on 39XX/51XX switches. For each pre-standards YANG model, Table 2-1 provides the SAOS license dependency and notes any unsupported features.

Table 2-1
Supported pre-standards YANG models

YANG model	SAOS feature	SAOS license dependency	Unsupported features
ietf-interfaces.yang	Port Aggregation	none	arbitrary-names pre-provisioning if-mib
ciena-mef-logical-port.yang	Port Aggregation	none	none
ciena-ieee-ethernet-port.yang	Port	none	none
ciena-ethernet-port.yang	Port	none	none

Copyright© 2022 Ciena® Corporation

Table 2-1
Supported pre-standards YANG models

YANG model	SAOS feature	SAOS license dependency	Unsupported features
ciena-ieee-lag.yang	Aggregation	none	none
ciena-chassis-sds.yang	Chassis management	none	Operational data
mef-cfm.yang	CFM	none	none
ciena-cpu-rate-limit-sds.yang	CPU Rate Limit	none	Operational Data
ciena-configuration-managmenet.yang	Configuration management	none	Direct manipulation of the configuration file
ciena-dhcpv6.yang	DHCP	none	Operational data
ciena-dhcp.yang	DHCP	none	Operational data
ciena-dns-client.yang	DNS	None	operational data
ciena-ieee-port-dot1x.yang	dot1x	Security	none
ciena-ieee-port-eoam.yang	EOAM	Advanced OAM	none
ciena-fd.yang	FD Model	none	Operational Data
ciena-bw-calculation-mode.yang	Frame bandwidth calculation mode	none	
ciena-ip-acl-sds.yang	IP-ACL	none	Operational Data
ciena-ieee-lldp.yang	LLDP	none	dot1x-extension dot3x-extension
ciena-command-log.yang	Logging	none	Operational data Viewing logs
ciena-log-sds.yang	Logging	none	Operational data Viewing logs
ciena-security-log.yang	Logging	none	Operational data Viewing logs
ciena-management-interfaces-sds.yang	Management interfaces	none	Operational data IP interfaces Some interface settings
ciena-ieee-mstp.yang	MSTP	none	none

Copyright© 2022 Ciena® Corporation

Table 2-1
Supported pre-standards YANG models

YANG model	SAOS feature	SAOS license dependency	Unsupported features
ciena-ntp.yang	NTP	none	Operational data
ciena-perform.yang	Perform	none	Operational data
ciena-radius.yang	RADIUS	none	Operational data
ciena-reserved-vlan-sds.yang	Reserved VLANs	none	
ciena-ieee-rstp.yang	RSTP	none	none
ciena-snmp.yang ietf-snmp-common.yang ietf-snmp-community.yang ietf-snmp-engine.yang ietf-snmp-notification.yang ietf-snmp-proxy.yang ietf-snmp-ssh.yang ietf-snmp-target.yang ietf-snmp-tls.yang ietf-snmp-tsm.yang ietf-snmp-usm.yang ietf-snmp-vacm.yang ietf.snmp.yang ciena-ietf-snmp-deviation.yang	SNMP	Security license for SNMPv3 access	
ciena-software-licensing.yang	Software licensing		
ciena-command-file.yang	Software management	none	
ciena-software-mgmt.yang	Software management	none	

Copyright© 2022 Ciena® Corporation

Table 2-1
Supported pre-standards YANG models

YANG model	SAOS feature	SAOS license dependency	Unsupported features
ietf-syslog-types.yang ietf-syslog.yang ciena-ietf-syslog-deviation.yang ciena-syslog.yang	Syslog	none	Operational data
ciena-system.yang	System management	none	Operational data
ietf-system.yang ciena-ietf-system-deviation.yang	System management RADIUS User management User authentication NTP Time System Security DNS	Security license for security features	Operational data
ciena-tacacs.yang	TACACS	none	Operational data
ciena-telnet.yang	Telnet	none	
ciena-time.yang	Time	none	Operational data
ciena-mef-meter-profile.yang	Traffic Profiling	Advanced Ethernet	none
ciena-mef-qos-flow.yang	Traffic Profiling	Advanced Ethernet	none
ciena-ietf-twamp.yang	TWAMP	none	controlClient server sessionSender sessionReflector
ciena-users.yang	User management	none	Operational data
mef-cfm.yang	VLAN Virtual Switch Virtual Circuit	none	none
ciena-mef-fd.yang	VLAN Virtual Switch Virtual Circuit	none	none

Copyright© 2022 Ciena® Corporation

Table 2-1
Supported pre-standards YANG models

YANG model	SAOS feature	SAOS license dependency	Unsupported features
ciena-mef-fp.yang	VLAN Virtual Switch Virtual Circuit	none	none
ciena-mef-classifier.yang	VLAN Virtual Switch Virtual Circuit	none	filter-parameters-type vlan-stack-feature
ciena-mef-cos-to-frame-map.yang	VLAN Virtual Switch Virtual Circuit	none	none
ciena-mef-egress-qos.yang	VLAN Virtual Switch Virtual Circuit	none	none
ciena-mef-flood-containment-profile.yang	VLAN Virtual Switch Virtual Circuit	none	none
ciena-mef-frame-to-cos-map.yang	VLAN Virtual Switch Virtual Circuit	none	none
ciena-cfm-sds.yang	VS based CFM service	none	
ciena-soam-pm-sds.yang	Y.1731 PM (Legacy mode)	none	MEF mode SLM/DM sessions

YANG models can be augmented with extensions that add additional attributes or tables. YANG also supports deviation statements that document platform-specific or product-specific non-conformance for a particular YANG model, such as different default values and ranges, or unsupported tables or attributes.

Copyright© 2022 Ciena® Corporation

Table 2-2 lists the deviation files and augmentations for each supported pre-standards YANG model.

Table 2-2
Deviation files and augmentations for pre-standards YANG models

YANG model	Deviation file	Augments	Augmented by
ciena-ethernet-port.yang	ciena-ethernet-port-deviation.yang	ietf-interfaces.yang	none
ciena-ieee-ethernet-port.yang	none	ietf-interfaces.yang	none
ciena-ieee-lag.yang	none	ietf-interfaces.yang	none
ciena-ieee-lldp.yang	ciena-ieee-lldp-deviation.yang	none	none
ciena-ieee-mstp.yang	none	ietf-interfaces.yang	none
ciena-ieee-port-dot1x.yang	ciena-ieee-port-dot1x-deviation.yang	ietf-interfaces.yang	ciena-dot1x.yang
ciena-ieee-port-eoam.yang	ciena-ieee-port-eoam-deviation.yang	ietf-interfaces.yang	ciena-eoam.yang
ciena-ieee-rstp.yang	none	ietf-interfaces.yang	none
ciena-ietf-twamp.yang	ciena-ietf-twamp-deviation.yang	none	ciena-twamp.yang
ciena-mef-classifier.yang	ciena-mef-classifier-deviation.yang	none	none
ciena-mef-cos-to-frame-map.yang	ciena-mef-cos-to-frame-map-deviation.yang	none	none
ciena-mef-egress-qos.yang	ciena-mef-egress-qos-deviation.yang	none	ciena-egress-qos.yang
ciena-mef-fd.yang	ciena-mef-fd-deviation.yang	none	none
ciena-mef-flood-containment-profile.yang	ciena-mef-flood-containment-profile-deviation.yang	none	none
ciena-mef-fp.yang	ciena-mef-fp-deviation.yang	none	none
ciena-mef-frame-to-cos-map.yang	ciena-mef-frame-to-cos-map-deviation.yang	none	none

Copyright© 2022 Ciena® Corporation

Table 2-2
Deviation files and augmentations for pre-standards YANG models

YANG model	Deviation file	Augments	Augmented by
ciena-mef-logical-port.yang	ciena-mef-logical-port-deviation.yang	none	ciena-logical-port.yang
ciena-mef-meter-profile.yang	ciena-mef-meter-profile-deviation.yang	none	none
ciena-mef-qos-flow.yang	ciena-mef-qos-flow-deviation.yang	none	ciena-qos-flow.yang
ietf-interfaces.yang	none	none	when type=ieee8023adLag: ciena-ieee-lag.yang when type=ethernetCsmacd: ciena-ieee-ethernet-port.yang ciena-ieee-lldp.yang ciena-ieee-mstp.yang ciena-ieee-port-dot1x.yang ciena-ieee-port-eoam.yang ciena-ieee-rstp.yang
mef-cfm.yang	mef-cfm-deviation.yang	none	none

Copyright© 2022 Ciena® Corporation

Copyright© 2022 Ciena® Corporation

Ciena YANG model schemas

This chapter provides the Ciena schemas for YANG models used with 39XX/51XX switches.

ciena-access-flow-deviation-sds.yang

```
module ciena-access-flow-deviation-sds {
   namespace "http://www.ciena.com/ns/yang/ciena-access-flow-deviation-sds";
   prefix "ciena-acl-deviation-sds";

   import ciena-mef-access-flow {
      prefix "mef-acl";
   }

   organization
      "Ciena Corporation";

   contact
       "Web URL: http://www.ciena.com/
       E-mail:  yang@ciena.com
       Postal:  7035 Ridge Road
                Hanover, Maryland 21076
                U.S.A.
       Phone:   +1 800-921-1144
       Fax:     +1 410-694-5750";

   description
      "This YANG module defines the deviations for Access Flows and
       Access Profiles for 6x platforms.

       Copyright (c) 2016 Ciena Corporation.  All rights reserved.

       All information contained herein is, and remains
       the property of Ciena Corporation. Dissemination of this
       information or reproduction of this material is strictly
       forbidden unless prior written permission is obtained from
       Ciena Corporation.";

   revision 2017-06-20 {
      description "Initial revision";
   }

   deviation /mef-acl:access-profiles/mef-acl:access-profile/mef-acl:stats-
collection
   {
      deviate not-supported;
```

```
   }

   deviation /mef-acl:access-profiles/mef-acl:access-profile/mef-
acl:description
   {
      deviate not-supported;
   }

   deviation /mef-acl:access-flows/mef-acl:access-flow/mef-acl:stats-
collection
   {
      deviate not-supported;
   }

   deviation /mef-acl:access-flows/mef-acl:access-flow/mef-acl:description
   {
      deviate not-supported;
   }

   deviation /mef-acl:access-flows/mef-acl:access-flow/mef-acl:parent-
interface
   {
      deviate not-supported;
   }

   deviation /mef-acl:acl-actions/mef-acl:acl-action/mef-acl:description
   {
      deviate not-supported;
   }

   deviation /mef-acl:access-flows/mef-acl:access-flow/mef-acl:classifier-
precedence {
      deviate replace {
         type uint32 {
            range "0..255";
         }
      }
   }

   deviation /mef-acl:acl-actions/mef-acl:acl-action/mef-acl:name {
      deviate replace {
         type string {
            length "1..31";
         }
      }
   }

   deviation /mef-acl:access-flows/mef-acl:access-flow/mef-acl:name {
      deviate replace {
         type string {
            length "1..63";
         }
      }
   }

   deviation /mef-acl:access-profiles/mef-acl:access-profile/mef-acl:name {
      deviate replace {
         type string {
            length "1..31";
         }
```


Copyright© 2022 Ciena® Corporation

```
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-access-flow-sds.yang

```
module ciena-access-flow-sds {
   namespace "http://www.ciena.com/ns/yang/ciena-access-flow-sds";
   prefix "ciena-access-flow-sds";

   import ietf-yang-types {
      prefix yang;
   }

   import ciena-mef-access-flow {
      prefix "mef-acl";
   }

   import ciena-mef-fd {
      prefix mef-fd;
   }

   import ciena-fd {
      prefix ciena-fd;
   }

   import ciena-types {
      prefix "ciena-type";
   }

   import ciena-mef-logical-port {
      prefix "mef-logical-port";
   }

   import ciena-management-interfaces-sds {
      prefix "ciena-mgmt-interfaces";
   }

   organization
      "Ciena Corporation";

   contact
       "Web URL: http://www.ciena.com/
       E-mail:  yang@ciena.com
       Postal:  7035 Ridge Road
                Hanover, Maryland 21076
                U.S.A.
       Phone:   +1 800-921-1144
       Fax:     +1 410-694-5750";

   description
      "This YANG module defines the Augments for Access Flows and
       Access Profiles for 6x platforms.

       Copyright (c) 2016 Ciena Corporation.  All rights reserved.

       All information contained herein is, and remains
       the property of Ciena Corporation. Dissemination of this
       information or reproduction of this material is strictly
       forbidden unless prior written permission is obtained from
       Ciena Corporation.";

   revision "2017-09-18" {
      description
```


Copyright© 2022 Ciena® Corporation

```
        "Augmented logical-port-status for ACL Flow stats";
  }

  revision 2017-06-20 {
     description "Initial revision";
  }

  grouping access-flow-statistics {
     description
        "packet and byte match statistics.";

     leaf hitFrames {
        type yang:counter64;
        description
           "Number of pkts matching this entry.";
     }

     leaf hitBytes {
        type yang:counter64;
        description
           "Number of bytes in pkts matching this entry.";
     }
  }

// data nodes

augment "/mef-acl:access-profiles" {
   description
     "Augment Access Profiles global data for 6.x.";

   container ciena-access-profiles-global-config-data {
     description
       "The global config for ACL Profiles.";

     leaf admin-state {
       type ciena-type:admin-state;

       default enabled;

       description
         "The global administrative state.";
     }

     leaf filter-mode {
       type enumeration {
         enum l2-l3-combo {
           description
             "Support L2, L3 match criteria.";
         }
         enum l3-only {
           description
             "Support L3 match criteria only.";
         }
       }

       default l2-l3-combo;

       description
         "The device wide filter mode.";
     }
```


Copyright© 2022 Ciena® Corporation

```
    }
  }

  augment "/mef-acl:access-profiles/mef-acl:access-profile" {
     description
        "Augment ACL Profile for 6.x.";

     container ciena-access-profiles {
        description
           "Augment ACL Profile for 6.x.";

        leaf admin-state {
           type ciena-type:admin-state;

           default enabled;

           description
              "Administrative state of ACL Profile.";
        }
     }
  }

  container access-profiles-state {
     description
        "ACL Profile oper data for 6.x.";

     config false;

     list access-profile {
        key "name";
        description
           "The operational data for this Access-Profile.";

        leaf name {
           type string;
           description
              "A string that identifies the Access-Profile.";
        }

        leaf number-of-flows {
           type uint16 {
              range "0 .. 255";
           }
           description
              "Number of Flows in this ACL profile.";
        }

        leaf number-of-attachments {
           type uint16;
           description
              "Number of entities that this profile is attached to.";
        }
     }

     leaf defined-access-profiles {
       type uint32;
       description
         "Number of ACL Profiles Defined.";
     }
     leaf remaining-access-profiles-available {
```


Copyright© 2022 Ciena® Corporation

```
        type uint32;
        description
          "Remaining ACL Profile Definitions.";
      }
   }

   augment "/mef-acl:access-flows-state" {
      description
         "Augment Access Flow oper data.";

    container ciena-access-flows-state {
      config false;

      description
        "Operational data for ACL Profile for 6.x.";

      leaf defined-access-flows {
        type uint32;
        description
          "Number of ACL Flows Defined.";
      }
      leaf remaining-access-flows-available {
        type uint32;
        description
          "Remaining ACL Flows Definitions.";
      }
      leaf active-access-flows-used {
        type uint32;
        description
          "Number of ACL Active Flows Used.";
      }
      leaf active-access-flows-free {
        type uint32;
        description
          "Number of ACL Active Flows Free.";
      }
    }
  }

   // add ingress ACL Profile to L2/L3 logical interfaces
   // vlan/VS  -> fd
   // logical-port -> Logical port
   // ip-interface - not supported yet

   augment "/mef-fd:fds/mef-fd:fd/ciena-fd:ciena-fd" {
      description
         "Apply ingress ACL Profile to VLAN/VS.";

      leaf ingress-acl {
         type mef-acl:access-profile-ref;
         description
            "Ingress ACL Profile name.";
      }
   }

   augment "/mef-fd:fds-state/mef-fd:fd" {
      description
         "Access Flow statistics for this FD.";

      container access-flow-statistics {
```


Copyright© 2022 Ciena® Corporation

```
         description
            "Per FD per Access Flow statistics.";

         list access-flows {
            key "access-flow-name";

            description
               "List of Access Flow Entries.";

            leaf access-flow-name {
               type string;

               description
                  "A unique name identifying this Access Flow Entry.";
            }

            uses access-flow-statistics;
         }
      }
   }

   augment "/mef-logical-port:logical-ports/mef-logical-port:logical-port" {

      description
         "Apply ingress ACL Profile to logical port.";

      leaf ingress-acl {
         type mef-acl:access-profile-ref;
         description
           "Ingress ACL Profile name.";
      }
   }

   augment "/mef-logical-port:logical-port-oper-status/mef-logical-
port:logical-port-status" {
      description
         "Access Flow statistics for this logical port.";

      container access-flow-statistics {
         description
            "Per logical port per Access Flow statistics.";

         list access-flows {
            key "access-flow-name";

            description
               "List of Access Flow Entries.";

            leaf access-flow-name {
               type string;

               description
                  "A unique name identifying this Access Flow Entry.";
            }

            uses access-flow-statistics;
         }
      }
   }
```


Copyright© 2022 Ciena® Corporation

```
   /*
    * RPC
    */

   rpc clear-statistics {
      description "An RPC operation to clear the statistics for an ACL
profile,FD,Interface and Logical port.";

      input {
         choice type {

            case acl-profile {
               leaf acl-name {
                  type mef-acl:access-profile-ref;
                  description
                     "The ACL Profile to clear statistics on.";
               }
            }
            case fd {
               leaf fd-name {
                  type leafref {
                     path "/mef-fd:fds/mef-fd:fd/mef-fd:name";
                  }
                  description
                     "The FD to clear ACL Profile statistics on.";
               }
            }
            case logical-port {
               leaf logical-port {
                  type mef-logical-port:logical-port-ref;
                  description
                    "The logical port to clear ACL Profile statistics on.";
               }
            }
            case mgmt-interface {
               leaf mgmt-interface {
                  type leafref {
                    path "/ciena-mgmt-interfaces:mgmt-interfaces/ciena-mgmt-
interfaces:interface/ciena-mgmt-interfaces:name";
                  }
                  description
                   "The managament interface to clear ACL Profile statistics
on.";
               }
            }
         } // end type choice
      }
      output {
         leaf status {
            type string;

            description
               "Status of the clear statistics operation.";
         }
      }
   } // end rpc clear-statistics
}
```


Copyright© 2022 Ciena® Corporation

ciena-alarm.yang

```
module ciena-alarm {
  namespace "http://www.ciena.com/ns/yang/ciena-alarm";
  prefix ciena-alarm;

  import ietf-yang-types {
    prefix "yang";
  }

  import ciena-types {
    prefix ciena;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module defines nodes for the management of alarm
     indications through the NETCONF protocol.

     Alarms are generated when a failure or fault is detected in the
     network elements. Events are correlated into alarms based on
     their significance. Alarms are messages that call for an
     operator or administrator's attention. When an alarm is created,
     the problem associated with the alarm must be resolved.

     Copyright (c) 2017 Ciena Corporation.  All rights reserved.

     All information contained herein is, and remains the property of
     Ciena Corporation. Dissemination of this information or
     reproduction of this material is strictly forbidden unless prior
     written permission is obtained from Ciena Corporation.";

  revision 2017-08-25 {
    description
      "Initial revision";
    reference
      "39XX/51XX Service Delivery, Aggregation and Virtualization
       Switches, Fault and Performance Management, SAOS 6.16
       (009-3273-009 - Standard Revision A)
       (Ciena Corporation, December 2016)

       8700 Packetwave Platform, Fault and Performance Management,
       SAOS 8.4 (380-1874-520 - Standard Revision A)
       (Ciena Corporation, May 2016)";
  }

  /*
   * Features
   */
```


Copyright© 2022 Ciena® Corporation

```
feature alarm-cutoff {
  description
    "This feature is enabled when a system supports an external
     audible alarm output.";
}

feature external-alarms {
  description
    "This feature is enabled when a system supports external alarm
     inputs for monitoring external equipment such as Intrusion,
     Fire, and Flood sensors.";
}

/*
 * Type definitions
 */

typedef alarm-indication {
  type enumeration {
    enum warning {
      description
        "A virtual Warning indication. This indication has no
         physical expression.";
    }
    enum minor-visible {
      description
        "The visual indication corresponding to Minor severity.";
    }
    enum major-visible {
      description
        "The visual indication corresponding to Major severity.";
    }
    enum critical-visible {
      description
        "The visual indication corresponding to Critical
         severity.";
    }
    enum minor-audible {
      description
        "The audible indication corresponding to Minor severity.";
    }
    enum major-audible {
      description
        "The audible indication corresponding to Major severity.";
    }
    enum critical-audible {
      description
        "The audible indication corresponding to Critical
         severity.";
    }
  }
  description
    "The physical manifestation of a Set alarm.";
}

typedef alarm-sense {
  type enumeration {
    enum closed {
      description
```


Copyright© 2022 Ciena® Corporation

```
          "Indicates the trigger polarity of the external alarm
           input pin is closed.";
      }
      enum open {
        description
          "Indicates the trigger polarity of the external alarm
           input pin is open.";
      }
    }
    description
      "The trigger polarity of the external alarm input pin.";
  }

  typedef alarm-severity {
    type enumeration {
      enum cleared {
        description
          "Indicates the clearing of one or more previously reported
           alarms.";
      }
      enum indeterminate {
        description
          "Indicates that the severity level cannot be determined.";
      }
      enum critical {
        description
          "Indicates that a service affecting condition has occurred
           and an immediate corrective action is required. Such a
           severity can be reported, for example, when a managed
           object becomes totally out of service and its capability
           must be restored.";
      }
      enum major {
        description
          "Indicates that a service affecting condition has developed
           and an urgent corrective action is required. Such a
           severity can be reported, for example, when there is a
           severe degradation in the capability of the managed object
           and its full capability must be restored.";
      }
      enum minor {
        description
          "Indicates the existence of a non-service affecting fault
           condition and that corrective action should be taken in
           order to prevent a more serious (for example, service
           affecting) fault. Such a severity can be reported, for
           example, when the detected alarm condition is not
           currently degrading the capacity of the managed object.";
      }
      enum warning {
        description
          "Indicates the detection of a potential or impending
           service affecting fault, before any significant effects
           have been felt. Action should be taken to further diagnose
           (if necessary) and correct the problem in order to prevent
           it from becoming a more serious service affecting fault.";
      }
    }
    description
      "Provides an indication of how it is perceived that the
```


Copyright© 2022 Ciena® Corporation

```
      capability of the managed object has been affected. Those
      severity levels which represent service affecting conditions
      ordered from most severe to least severe are Critical, Major,
      Minor, and Warning.";
    reference
      "International Telecommunication Union (ITU)
      Recommendation X.733 Sec. 8.1.2.3 Perceived Severity";
  }

  /*
   * Groupings
   */

  grouping count-escalation {
    description
      "Defines the attributes used in count-based severity
      escalation. This form of escalation can be used with events
      that are sent by the system multiple times at various levels
      of severity. All levels of severity must be defined within
      the range of severities used, lowest to highest. The system
      will use the highest severity at a given threshold.";

    leaf minor-at {
      type uint16 {
        range "0..9999";
      }
      must "../../severity > 'minor'" {
        error-message
          "Cannot escalate to minor from current base severity.";
        description
          "Cannot escalate to a severity that is equal to or lower
           than the previous severity.";
      }
      description
        "The number of triggering events at which the alarm will be
         escalated to minor severity. Escalation requires that both a
         base threshold and severity be set.";
    }
    leaf major-at {
      type uint16 {
        range "0..9999";
      }
      must "../minor-at or ../../severity = 'minor'" {
        error-message
          "Must also specify minor escalation to specify major
           escalation";
        description
          "All levels of severity must be defined within the range of
           severities used.";
      }
      must "../../severity > 'major'" {
        error-message
          "Cannot escalate to major from current base severity.";
        description
          "Cannot escalate to a severity that is equal to or lower
           than the previous severity.";
      }
      description
        "The number of triggering events at which the alarm will be
         escalated to major severity. Escalation requires that both a
```


Copyright© 2022 Ciena® Corporation

```
        base threshold and severity be set.";
    }
    leaf critical-at {
      type uint16 {
        range "0..9999";
      }
      must "../major-at or ../../severity = 'major'" {
        error-message
          "Must also specify minor and major escalations to specify
           critical escalation";
        description
          "All levels of severity must be defined within the range of
           severities used.";
      }
      must "../../severity > 'critical'" {
        error-message
          "Cannot escalate to critical from current base severity.";
        description
          "Cannot escalate to a severity that is equal to or lower
           than the previous severity.";
      }
      description
        "The number of triggering events at which the alarm will be
         escalated to critical severity. Escalation requires that
         both a base threshold and severity be set.";
    }
  } // end grouping count-escalation

  grouping time-escalation {
    description
      "Defines the attributes used in time-based severity escalation.
       This form of escalation can be used with single events that
       become more severe over time when not addressed.";

    leaf to-minor {
      type uint32;
      units seconds;
      must "../../severity > 'minor'" {
        error-message
          "Cannot escalate to minor from current base severity.";
        description
          "Cannot escalate to a severity that is equal to or lower
           than the previous severity.";
      }
      description
        "The duration of time after which the alarm will be escalated
         to minor severity. Escalation requires that a base severity
         be set.";
    }
    leaf to-major {
      type uint32;
      units seconds;
      must "../to-minor or ../../severity = 'minor'" {
        error-message
          "Must also specify minor escalation to specify major
           escalation";
        description
          "All levels of severity must be defined within the range of
           severities used.";
      }
```


Copyright© 2022 Ciena® Corporation

```
      must "../../severity > 'major'" {
        error-message
          "Cannot escalate to major from current base severity.";
        description
          "Cannot escalate to a severity that is equal to or lower
           than the previous severity.";
      }
      description
        "The duration of time after which the alarm will be escalated
         to major severity. Escalation requires that a base severity
         be set.";
    }
    leaf to-critical {
      type uint32;
      units seconds;
      must "../to-major or ../../severity = 'major'" {
        error-message
          "Must also specify minor and major escalations to specify
           critical escalation";
        description
          "All levels of severity must be defined within the range of
           severities used.";
      }
      must "../../severity > 'critical'" {
        error-message
          "Cannot escalate to critical from current base severity.";
        description
          "Cannot escalate to a severity that is equal to or lower
           than the previous severity.";
      }
      description
        "The duration of time after which the alarm will be escalated
         to critical severity. Escalation requires that a base
         severity be set.";
    }
  } // end time-escalation

  /*
   * Configuration data nodes
   */

  container alarm {
    description
      "Configuration parameters for the management of system
       alarms.";

    list defined-alarm {
      key table-id;
      description
        "A list containing the configurable attributes of each alarm
         defined on the device.";

      leaf table-id {
        type uint32;
        description
          "The Alarm Table Identifier (ATID) is a unique index
           associated with each defined alarm.";
      }
      leaf enabled {
        type boolean;
```


Copyright© 2022 Ciena® Corporation

```
    description
      "Enables or disables an alarm. Disabling an alarm means
       that no future instances of the alarm class will occur,
       regardless of triggering events, until it is enabled. Some
       alarms can't be disabled.";
  }
  leaf threshold {
    type uint16 {
      range "1..9999";
    }
    default 1;
    description
      "A threshold that prevents the setting of an alarm until a
       certain number of triggering events have occurred for that
       instance. This threshold applies to the base severity of
       the alarm.";
  }
  leaf severity {
    type alarm-severity;
    description
      "The base severity of the alarm.";
  }

  container count-escalation {
    must "../severity" {
      error-message
        "Must also specify base severity to specify a count
         based escalation.";
      description
        "Prevent count escalation without a base severity.";
    }
    must "../threshold" {
      error-message
        "Must also specify base threshold to specify minor
         escalation";
      description
        "Prevent count escalation without a base threshold.";
    }
    description
      "Configurable parameters for defining count-based severity
       escalation.";
    uses count-escalation;
  }

  container time-escalation {
    must "../severity" {
      error-message
        "Must also specify base severity to specify a time based
         escalation.";
      description
        "Prevent time escalation without a base severity.";
    }
    description
      "Configurable parameters for defining time-based severity
       escalation.";
    uses time-escalation;
  }

  container rate-alarm {
    description
```


Copyright© 2022 Ciena® Corporation

```
        "Configurable parameters for defining rate-based alarms.";

      leaf cap {
        type uint16 {
          range "0..9999";
        }
        description
          "A defined limit to the number of Set events per instance
           that will be counted. The occurrence cap can be used in
           a circumstance where multiple Set events might flow
           through the system, yet a single Clear event is expected
           to cancel the alarm (this would be a Cap of 1).";
      }
      leaf leak {
        type uint16 {
          range "0..9999";
        }
        must "../leak-interval > 0" {
          error-message
            "Must also specify leak-interval to specify leak";
          description
            "Prevent the setting of leak without a leak-interval.";
        }
        description
          "The number by which the event counter will be
           decremented at every leak-interval.";
      }
      leaf leak-interval {
        type uint32;
        units seconds;
        description
          "The duration of time that will pass before the event
           counter is decremented by the specified leak amount.";
      }
    }

    leaf description {
      type string {
        length "1..25";
      }
      description
        "A textual description of the alarm.";
    }
    leaf sense {
      if-feature external-alarms;
      type alarm-sense;
      description
        "The trigger polarity of the external alarm input pin.";
    }
    leaf forced {
      if-feature external-alarms;
      type boolean;
      description
        "Indicates whether the alarm has been force-triggered.
         Setting this leaf to 'true' will force-trigger the alarm
         to an active state. Setting this leaf to 'false' will
         clear the alarm.";
    }
  } // end list defined-alarm
```


Copyright© 2022 Ciena® Corporation

```
    list alarm-status {
      key description;
      description
        "A list containing the configurable attributes of each type
         of alarm indication on the device.";

      leaf description {
        type alarm-indication;
        description
          "Describes the type of alarm indication.";
      }
      leaf enabled {
        type boolean;
        description
          "Enables or disables the alarm indication.";
      }
    } // end list alarm-status
  } // end container alarm

  /*
   * Operational state data nodes
   */

  container alarm-state {
    config false;
    description
      "Operational state parameters of system alarms.";

    list active-alarm {
      key instance-id;
      description
        "A list containing the operational parameters of alarms that
         are currently active on the device.";

      leaf instance-id {
        type uint32;
        description
          "The Instance Identifier (IID) is a unique index
           associated with each active instance of an alarm.";
      }
      leaf acknowledged {
        type boolean;
        description
          "Indicates whether the alarm has been acknowledged. An
           acknowledged alarm will no longer manifest a physical
           indication.";
      }
      leaf table-id {
        type uint32;
        description
          "The Alarm Table Identifier (ATID) is a unique index
           associated with each defined alarm.";
      }
      leaf severity {
        type alarm-severity;
        description
          "The severity of the active alarm instance.";
      }
      leaf timestamp {
        type yang:date-and-time;
```


Copyright© 2022 Ciena® Corporation

```
      description
        "The date and time this alarm instance became active,
         expressed in local time.";
    }
    leaf instance {
      type uint32;
      description
        "The instance value used to distinguish one alarm source
         from another. This value corresponds with the
         description of the instance in the defined-alarm list.";
    }
    leaf description {
      type string {
        length "1..25";
      }
      description
        "A textual description of the alarm.";
    }
  }

  list defined-alarm {
    key table-id;
    description
      "A list containing the operational parameters of each alarm
       defined on the device.";

    leaf table-id {
      type uint32;
      description
        "The Alarm Table Identifier (ATID) is a unique index
         associated with each defined alarm.";
    }
    leaf active {
      type boolean;
      description
        "True if the alarm is currently active, false if it's
         inactive. Active alarms will appear in the active-alarm
         list.";
    }
    leaf count {
      type uint32;
      description
        "The number of currently active instances of the alarm.";
    }
    leaf instance {
      type string;
      description
        "A description of the instance value used to distinguish
         one alarm source from another, e.g. Port ID. This
         corresponds with the value of the instance in the
         active-alarm list.";
    }
    leaf probable-cause {
      type string;
      description
        "A description of the probable cause of the alarm.";
      reference
        "International Telecommunication Union (ITU) Recommendation
         X.733 Sec. 8.1.2.1 Probable Cause";
    }
```


Copyright© 2022 Ciena® Corporation

```
    } // end list defined-alarm

    list alarm-statistics {
      key description;
      description
        "Operational statistics associated with each type of alarm
         indication on the device.";

      leaf description {
        type alarm-indication;
        description
          "Describes the type of alarm indication.";
      }
      leaf count {
        type uint32;
        description
          "The number of currently active alarms for which the
           indication is set.";
      }
      leaf cumulative {
        type uint32;
        description
          "The cumulative number of alarms for which the indication
           has been set.";
      }
    } // end list alarm-statistics

    list alarm-status {
      key description;
      description
        "Operational state parameters of each type of alarm
         indication on the device.";

      leaf description {
        type alarm-indication;
        description
          "Describes the type of alarm indication.";
      }
      leaf on {
        type boolean;
        description
          "Whether the alarm indication is currently on or off.";
      }
    } // end list alarm-status
  } // end container alarm-state

  /*
   * RPCs
   */

  rpc alarm-acknowledge {
    description
      "Cut-off the audible or visual effect for a specific alarm.";

    input {
      leaf-list instance-id {
        type uint32;
        description
          "The Instance Identifiers of specific alarms in the
           active-alarm list to be acknowledged.";
```


Copyright© 2022 Ciena® Corporation

```
    }
  }
}

rpc alarm-clear-history {
  description
    "Clear the alarm history.";
}

rpc alarm-cutoff {
  if-feature alarm-cutoff;
  description
    "Place the system into alarm cutoff state and turn OFF the
     external audible alarm output. When the system is in alarm
     cutoff state, a newly raised alarm (critical, major, or minor)
     will clear the alarm cutoff state and turn ON the external
     audible alarm output.";
}

rpc alarm-delete {
  description
    "Remove a specific alarm instance from the active-alarm list.";

  input {
    leaf-list instance-id {
      type uint32;
      description
        "The Instance Identifiers of specific alarms in the
         active-alarm list to be deleted.";
    }
  }
}

rpc alarm-history {
  description
    "Return all or a specified range of alarm history.";

  input {
    choice filter-attribute {
      description
        "A set of mutually exclusive options for filtering the
         history log output.";

      leaf recent {
        type empty;
        description
          "Recent entries in the history log.";
      }
      leaf since-activated {
        type empty;
        description
          "Entries in the history log from system start or
           failover.";
      }
      leaf today {
        type empty;
        description
          "Today's entries in the history log.";
      }
      leaf yesterday {
```


Copyright© 2022 Ciena® Corporation

```
      type empty;
      description
        "Yesterday's entries in the history log.";
    }
    leaf from-date {
      type ciena:simple-date;
      description
        "History entries from date (inclusive).";
    }
    leaf last {
      type uint32;
      units seconds;
      must "not(../to-date) and not(../from-time)" {
        error-message
          "Cannot specify last with to-date or from-time.";
        description
          "Prevent mutually exclusive filter options.";
      }
      description
        "Entries added to the history log during the last
         specified time period.";
    }
  } // end choice filter-attribute

  leaf to-date {
    type ciena:simple-date;
    must "not(../recent) and not(../today) and not(../yesterday)" {
      error-message
        "Cannot specify to-date with recent, today, or
         yesterday.";
      description
        "Prevent mutually exclusive filter options.";
    }
    description
      "History entries to date (inclusive).";
  }
  leaf from-time {
    type ciena:simple-time;
    must "not(../recent) and not(../last)" {
      error-message
        "Cannot specify from-time with recent or last.";
      description
        "Prevent mutually exclusive filter options.";
    }
    description
      "History entries from time (inclusive).";
  }
  leaf to-time {
    type ciena:simple-time;
    description
      "History entries to time (inclusive).";
  }
  leaf severity {
    type ciena:event-severity;
    description
      "The severity of the history log event.";
  }
  leaf table-id {
    type uint32;
    description
```


Copyright© 2022 Ciena® Corporation

```
        "The Alarm Table Identifier (ATID) is a unique index
         associated with each defined alarm.";
    }
    leaf tail {
      type uint32;
      description
        "Specified number of latest entries in the history log.";
    }
  } // end input

  output {
    list alarm-history {
      key history-id;
      description
        "A log of alarm history. Alarm history includes set alarm
         instances, acknowledgement or deletion of alarm instances,
         and configuration changes to defined alarms.";

      leaf history-id {
        type uint32;
        description
          "The History Identifier (HID) is a unique index
           associated with each entry in the alarm-history list.";
      }
      leaf reason {
        type string;
        description
          "The reason the history event appears in the log.";
      }
      leaf instance-id {
        type uint32;
        description
          "The Instance Identifier (IID) is a unique index
           associated with each active instance of an alarm.";
      }
      leaf table-id {
        type uint32;
        description
          "The Alarm Table Identifier (ATID) is a unique index
           associated with each defined alarm.";
      }
      leaf severity {
        type ciena:event-severity;
        description
          "The severity of the history log event.";
      }
      leaf timestamp {
        type yang:date-and-time;
        description
          "The date and time of the history event.";
      }
      leaf instance {
        type uint32;
        description
          "The instance value used to distinguish one alarm source
           from another. This value corresponds with the
           description of the instance in the defined-alarm list.";
      }
      leaf description {
        type string {
```


Copyright© 2022 Ciena® Corporation

```
            length "1..25";
          }
          description
            "A textual description of the alarm.";
        }
      } // end list alarm-history
    } // end output
  } // end rpc alarm-history
} // end module ciena-alarm
```


Copyright© 2022 Ciena® Corporation

ciena-bw-calculation-mode.yang

```
module ciena-bw-calculation-mode {
  namespace "http://www.ciena.com/ns/yang/ciena-bw-calculation-mode";
  prefix "ciena-bw-calculation-mode";

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
    E-mail:  yang@ciena.com
    Postal:  7035 Ridge Road
             Hanover, Maryland 21076
             U.S.A.
    Phone:   +1 800-921-1144
    Fax:     +1 410-694-5750";

  description
    "This YANG module defines the nodes for
     BW Calculation Mode.

     Copyright (c) 2017 Ciena Corporation.  All rights
     reserved.

     All information contained herein is, and remains
     the property of Ciena Corporation. Dissemination of this
     information or reproduction of this material is strictly
     forbidden unless prior written permission is obtained from
     Ciena Corporation.";

  revision "2016-12-08"
  {
      description
          "Initial version";
  }

  /*
   * Type definitions
   */

   identity bw-calculation-modes {
      description
         "Modes of bandwidth calculation.";
   }

   identity  transport {
       base bw-calculation-modes;
       description
          "Bandwidth calculation mode is transport.";
   }

   identity payload {
      base bw-calculation-modes;
      description
         "Bandwidth calculation mode is payload.";
   }

  /*
   * Configuration data nodes
```

```
   */

   container bw-calculation-mode {
       description
           "Mode configuration for bandwidth calculations.";

           leaf eqos-bw-calculation-mode {
               type identityref {
                   base bw-calculation-modes;
               }
               default transport;

               description
                  "Egress QoS Bandwidth calculation mode selection. (Egress
Shaping)";
           }
           leaf iqos-bw-calculation-mode {
               type identityref {
                   base bw-calculation-modes;
               }
               default transport;

               description
                 "Ingress QoS Bandwidth calculation mode selection. (Ingress
Metering)";
           }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-cfm-sds.yang

```
module ciena-cfm-sds {

  namespace "http://www.ciena.com/ns/yang/ciena-cfm-sds";
  prefix "ciena-cfm-sds";

  import mef-cfm {
    prefix "cfm";
  }

  import ciena-mef-fd {
    prefix "mef-fd";
  }

  organization
   "Ciena Corporation";

  contact
   "Web URL: http://www.ciena.com/
    E-mail:  yang@ciena.com
    Postal:  7035 Ridge Road
             Hanover, Maryland 21076
             U.S.A.
    Phone:   +1 800-921-1144
    Fax:     +1 410-694-5750";

  description
   "This YANG module contains the Ciena's addition to management data
    definitions for the management of Ethernet Services Operations,
    Administration and Maintenance for Connectivity Fault Management
    for the NETCONF protocol.";

  revision 2018-08-10 {
    description
     "Added support to configure CFM over LAG member ports.";
  }

  revision 2018-05-25 {
    description
     "Aligned the default value of alarm-time with SAOS";
  }

  revision 2018-02-01 {
    description
     "Added leaf admin-state for maintenance-association";
  }

  revision 2017-02-21 {
    description
     "Initial revision";
  }

  //
  // Type definitions related to CFM
  //
  typedef cfm-admin-state {
    type enumeration {
      enum disable {
          description "CFM global administrative state is
```


Copyright© 2022 Ciena® Corporation

```
            disabled";
      }
      enum enable {
          description "CFM global administrative state is
            enabled";
      }
    }
    description "CFM global admininistrative state";
  }

  typedef message-interval {
    type enumeration {
      enum invalid {
          description "No messages are sent (disabled).";
      }
      enum 3.3ms {
          description "Messages are sent every 3 1/3 milliseconds (300Hz).";
      }
      enum 10ms {
          description "Messages are sent every 10 milliseconds.";
      }
      enum 100ms {
          description "Messages are sent every 100 milliseconds.";
      }
      enum 1s {
          description "Messages are sent every 1 second.";
      }
      enum 10s {
          description "Messages are sent every 10 seconds.";
      }
      enum 1min {
          description "Messages are sent every minute.";
      }
      enum 10min {
          description "Messages are sent every 10 minutes.";
      }
    }

    description
      "Indicates the message interval type being used for CFM message
transmissions.";
  }

//
// Config Data related to ciena proprietary extension of CFM model
//

  container cfm-global-config {
    description
     "Contains configurable parameters applicable to all maintenance
      associations in the system";

    leaf admin-state {
      type cfm-admin-state;
      default disable;
      description
       "Enable or Disable cfm on this device";
    }

    leaf dot1ad-strict-enforcement {
```


Copyright© 2022 Ciena® Corporation

```
      type enumeration {
        enum on {
          description
            "Set Strict 802.1ad compliance to on.";
        }
        enum off {
          description
            "Set Strict 802.1ad compliance to off.";
        }
      }
      default on;
      description
        "Enable/Disable Strict 802.1ad compliance
         When enabled, only CFM frames that contain a single VLAN
         tag or are untagged are recognized as CFM frames. A
         double-tagged CFM frame will be forwarded as a normal
         data frame on interfaces containing MEPs and/or MIPs, and
         CFM processing is bypassed.";
    }

    leaf l2-loop-detection {
      type enumeration {
        enum on {
          description
            "Set L2 Loop Detection to on.";
        }
        enum off {
          description
            "Set L2 Loop Detection to off.";
        }
      }
      default off;
      description
        "Enables or disables Layer 2 loop detection.";
    }

    leaf mip-level-enforcement {
      type enumeration {
        enum on {
          description
            "Set MIP MD Level enforcement to on.";
        }
        enum off {
          description
            "Set MIP MD Level enforcement to off.";
        }
      }
      default on;
      description
        "Set MIP MD Level enforcement on/off.";
    }

    leaf automatic-meps {
      type enumeration {
        enum on {
          description
            "Set Automatic Up MEP creation to on for UNI members of Virtual-
switch.";
        }
        enum off {
```


Copyright© 2022 Ciena® Corporation

```
          description
            "Set Automatic Up MEP creation to off for UNI members of Virtual-
switch.";
          }
        }
        default on;
        description
          "Set Automatic UNI Up MEP creation on/off for UNI Members of Virtual
Switch.
           Feature is not supported on VLAN forwarding domain.";
      }

      leaf dmm-default-count {
        type uint32 {
          range "2..65535";
        }
        default 100;
        description
         "An integer value which specifies the default DMM count of frames that
need to be
          transmitted within a given iteration of a DMM session.";
      }

      leaf dmm-default-interval {
        type message-interval;
        default 100ms;
        description
          "This object specifies the default interval between DMM Performance
Measurement OAM message
           transmissions.";
      }

      leaf dmm-default-timeout {
        type uint16 {
          range "2000..10000";
        }
        units milliseconds;
        default 5000;
        description
          "Indicates the default DMM timeout value.";
      }

      leaf lmm-default-count {
        type uint32 {
          range "2..65535";
        }
        default 100;
        description
         "An integer value which specifies the default LMM count of frames that
need to be
          transmitted within a given iteration of a LMM session.";
      }

      leaf lmm-default-interval {
        type message-interval;
        default 100ms;
        description
          "This object specifies the default interval between LMM Performance
Measurement OAM message
           transmissions.";
```


Copyright© 2022 Ciena® Corporation

```
    }

    leaf lmm-default-timeout {
      type uint16 {
        range "2000..10000";
      }
      units milliseconds;
      default 5000;
      description
        "Indicates the default LMM timeout value.";
    }

    leaf slm-default-count {
      type uint32 {
        range "2..65535";
      }
      default 100;
      description
       "An integer value which specifies the default SLM count of frames that
need to be
        transmitted within a given iteration of a SLM session.";
    }

    leaf slm-default-interval {
      type message-interval;
      default 100ms;
      description
        "This object specifies the default interval between SLM Performance
Measurement OAM message
         transmissions.";
    }

    leaf slm-default-timeout {
      type uint16 {
        range "2000..10000";
      }
      units milliseconds;
      default 5000;
      description
        "Indicates the default SLM timeout value.";
    }
  }

  augment "/cfm:maintenance-domain/cfm:maintenance-association/
cfm:component-list" {
    leaf fd-name {
      type leafref {
        path "/mef-fd:fds/mef-fd:fd/mef-fd:name";
      }

      description
      "This gives reference to the flow-domain monitored by this MA, including
the
         direct VID or the VC service VID which is being monitored.";
    }
  }

  augment "/cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point" {
    leaf interface-type {
```


Copyright© 2022 Ciena® Corporation

```
      type enumeration {
        enum ieee8023adLag {
          description
            "Interface is a LAG port.";
        }
        enum ieee8023adLagMember {
          description
            "Interface is a LAG member port.";
        }
        enum ettp {
          description
            "Interface is a physical port and not member of any LAG.";
        }
      }
      description
        "The type of the interface on which MEP is being created.";
      reference
        "NA";
    }
    description
     "This object shall be used to specify type of the interface on which MEP
is being created.
       Purpose of interface-type leaf is to allow user to create MEP on
different type of interfaces.";
  }

  augment "/cfm:maintenance-domain/cfm:maintenance-association" {
    leaf accelerate {
      type boolean;
      default false;
      description
        "Indicates whether the service created is Hardware assisted or not.";
    }

    leaf alarm-priority {
      type uint8 {
        range "1..5";
      }
      default 3;
      description
        "Indicates the service fault alarm priority.";
    }

    leaf alarm-time {
      type uint16 {
        range "0..36000";
      }
      units milliseconds;
      default 3000;
      description
        "Indicates the service fault alarm time in milliseconds.";
    }

    leaf csf-interval {
      type message-interval;
      default 1s;
      description
        "Specifies the CSF transmission interval index or time.";
    }
```


Copyright© 2022 Ciena® Corporation

```
    leaf csf-loss-num {
      type uint8 {
        range "1..255";
      }
      default 3;
      description
        "Indicates the CSF loss number.";
    }

    leaf csf-dci-num {
      type uint8 {
        range "1..255";
      }
      default 3;
      description
        "Indicates the CSF Dci number.";
    }

    leaf ccm-loss-bucket-size {
      type uint8 {
        range "1..60";
      }
      default 15;
      description
        "Indicates the CCM loss bucket size in minutes.";
    }

    leaf ccm-loss-num {
      type uint8 {
        range "1..255";
      }
      default 3;
      description
        "Indicates the CCM loss number.";
    }

    leaf ccm-loss-stats {
      type enumeration {
        enum enable {
          description
            "Enables CCM Loss Statistics per Remote MEP.";
        }
        enum disable {
          description
            "Disables CCM Loss Statistics per Remote MEP.";
        }
      }
      default disable;
      description
        "Enables/Disables the CCM Loss Statistics per Remote MEP.";
    }

    leaf reset-time {
      type uint16 {
        range "10..36000";
      }
      units milliseconds;
      description
        "Indicates the service fault reset time.";
    }
```


Copyright© 2022 Ciena® Corporation

```
leaf remote-mep-aging-time {
  type uint32 {
    range "0..3600000";
  }
  units milliseconds;
  default 0;
  description
    "Indicates the remote mep aging time.";
}

leaf remote-mep-discovery {
  type enumeration {
    enum enable {
      description
        "Enables automatic discovery of remote MEPs.";
    }
    enum disable {
      description
        "Disables automatic discovery of remote MEPs.";
    }
  }
  description
    "Enables/Disables the automatic discovery of remote MEPs.";
}

leaf sender-id-type {
  type enumeration {
    enum none {
      description
        "No Sender ID TLV type to be set.";
    }
    enum chassis {
      description
        "Set the value to: chassis";
    }
    enum manage {
      description
        "Set the value to: manage";
    }
    enum chassis-manage {
      description
        "Set the value to: chassis-manage";
    }
  }
  description
    "Specifies the sender ID TLV type.";
}

leaf admin-state {
  type enumeration {
    enum enable {
      description
        "Enable cfm service.";
    }
    enum disable {
      description
        "Disable cfm service.";
    }
  }
```


Copyright© 2022 Ciena® Corporation

```
      default disable;
      description
        "Enable/Disable cfm service.";
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-chassis-sds.yang

```
module ciena-chassis-sds {
  namespace "http://www.ciena.com/ns/yang/ciena-chassis";
  prefix "ciena-chassis";

  import ietf-system {
    prefix sys;
  }

  import ietf-yang-types {
    prefix yang;
  }

  import ciena-types {
    prefix ciena;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module defines Ciena's global configuration for system chassis
settings.

     Copyright (c) 2016 Ciena Corporation.  All rights
     reserved.

     All information contained herein is, and remains
     the property of Ciena Corporation. Dissemination of this
     information or reproduction of this material is strictly
     forbidden unless prior written permission is obtained from
     Ciena Corporation.";

  revision 2017-05-25 {
    description
      "Initial version";
    reference
      "RFC 6020: YANG - A Data Modeling Language for
       the Network Configuration Protocol (NETCONF).
       No specific reference; standard not available.";
  }

  /*
   * typedefs
   */

  typedef post-subsystem {
    type enumeration {
      enum main {
        description
```


Copyright© 2022 Ciena® Corporation

```
          "Main subsystem error.";
      }

      enum psa {
        description
          "Power supply A subsystem error.";
      }

      enum psb {
        description
          "Power supply B subsystem error.";
      }

      enum fas {
        description
          "Fan subsystem error.";
      }

      enum module {
        description
          "Module subsystem error.";
      }
    }
    description
      "Subsystem of the POST result.";
  }

  typedef oam-service {
    type enumeration {
      enum oam-services {
        description
          "Set OAM service port role to OAM services.";
      }

      enum customer-traffic {
        description
          "Set OAM service port role to customer traffic.";
      }
    }
    description
      "OAM service roles.";
  }

  typedef post-severity {
    type enumeration {
      enum fatal {
        description
          "Fatal POST severity detected.";
      }

      enum severe {
        description
          "Severe POST severity detected.";
      }

      enum warning {
        description
          "Warning POST severity detected.";
      }
    }
```


Copyright© 2022 Ciena® Corporation

```
    description
      "Severity of the POST result.";
  }

  typedef hardware-state {
    type enumeration {
      enum offline {
        description
          "Hardware is offline.";
      }
      enum online {
        description
          "Hardware is online.";
      }
      enum faulted {
        description
          "Hardware is faulted.";
      }
      enum uninstalled {
        description
        "Hardware is uninstalled.";
      }
      enum unknown {
        description
          "Hardware state is unknown.";
      }
    }
    description
      "State of the a hardware component.";
  }

  typedef power-supply-type {
    type enumeration {
      enum ac {
        description
          "AC power supply.";
      }
      enum dc {
        description
          "DC power supply.";
      }
      enum unequipped {
        description
          "Power supply is unequipped.";
      }
      enum unknown {
        description
          "Unknown power supply.";
      }
    }
    description
      "Type of power supply installed.";
  }

  typedef fan-speed {
    type uint32;
    units rpm;
    description
      "Speed of the fan in revolutions per minute.";
  }
```


Copyright© 2022 Ciena® Corporation

```
/*
 * groupings
 */

grouping temperature-attributes {
  leaf current {
    type int32;
    units degrees-celsius;
    description
      "The current temperature.";
  }

  leaf high {
    type int32;
    units degrees-celsius;
    description
      "The high temperature.";
  }

  leaf low {
    type int32;
    units degrees-celsius;
    description
      "The low temperature.";
  }
  description
    "Common temperature attributes.";
}

grouping temperature-threshold-attributes {
  leaf low-threshold {
    type int32;
    units degrees-celsius;
    description
      "Low temperature threshold.";
  }

  leaf high-threshold {
    type int32;
    units degrees-celsius;
    description
      "High temperature threshold.";
  }
  description
    "Common temperature threshold attributes.";
}

grouping mac-partition {
  leaf base-mac {
    type yang:mac-address;
    description
      "Base MAC of the partition.";
  }

  leaf block-size {
    type uint16;
    description
      "Block size of the MAC partition.";
  }
```


Copyright© 2022 Ciena® Corporation

```
    description
      "Common MAC partition information.";
  }

  /*
   * Identities
   */

  /*
   * Features
   */

  /*
   * Configuration data nodes
   */
  augment /sys:system {
    description
      "Configuration of the system's chassis settings.";

    container chassis {
      description
        "System chassis administrative settings.";

      // Dying gasp
      container dying-gasp {
        description
          "Set the dying gasp states.";

        leaf state {
          type ciena:admin-state;
          description
            "The admin state of the dying gasp feature.";
        }

        leaf inform {
          type ciena:on-off-state;
          description
            "The SNMP inform state. When set to 'on', the dying gasp packet
structure is set to 'inform'.
             When set to 'off', the dying gasp packet structure is set to
'trap'.";
        }
      }

      // OAM services port role
      leaf oam-services-port-role {
        type oam-service;
        description
          "Set the OAM service states.";
      }

      // Temperature
      container temperature {
        uses temperature-threshold-attributes;

        description
          "Chassis temperature settings.";
      }
    }
  }
```


Copyright© 2022 Ciena® Corporation

```
/*
 * Operational state data nodes
 */
augment /sys:system-state {
  description
    "Operational data for the system's chassis settings.";

  container chassis {
    config false;

    description
      "Chassis operational data.";

    // Post
    leaf post-state {
      type ciena:on-off-state;
      description
        "The state of the post feature.";
    }

    leaf reboot-time {
      type yang:date-and-time;
      description
        "Date and time that the chassis will reboot.";
    }

    list post-results {
      key index;
      description
        "Results of the POST check.";

      leaf index {
        type uint32;
        description
          "Internal index of the post result.";
      }

      leaf subsystem {
        type post-subsystem;
        description
          "Subsystem of the POST error.";
      }

      leaf severity {
        type post-severity;
        description
          "Severity of the POST error.";
      }

      leaf message {
        type string;
        description
          "Information about the POST error.";
      }
    }

    // capablilites
    container capabilities {
      description
```


Copyright© 2022 Ciena® Corporation

```
    "Platform capabilities.";

  // chassis
  container chassis {
    description
      "Chassis related platform capabilities.";

    leaf capabilities-class {
      type uint32;
      description
        "Chassis capabilities class.";
    }

    leaf platform-type {
      type uint32;
      description
        "Platform type";
    }

    leaf platform-name {
      type string;
      description
        "Name of the platform.";
    }

    leaf platform-description {
      type string;
      description
        "Description of the platform.";
    }

    leaf num-slots {
      type uint32;
      description
        "Number of slots.";
    }

    leaf primary-ctrl-blade {
      type uint32;
      description
        "Primy control blade.";
    }

    leaf num-fan-trays {
      type uint32;
      description
        "Number of fan trays.";
    }

    leaf num-fans-per-tray {
      type uint32;
      description
        "Number of fans per tray.";
    }

    leaf fixed-dc-power {
      type boolean;
      description
        "Fixed DC power.";
    }
```


Copyright© 2022 Ciena® Corporation

```
    leaf redundant-power {
      type boolean;
      description
        "Redundant power.";
    }

    leaf max-physical-ports {
      type uint32;
      description
        "Max number of physical ports.";
    }

    leaf max-lag-ports {
      type uint32;
      description
        "Max number of LAG ports.";
    }

    leaf max-ip-interfaces {
      type uint32;
      description
        "Max number of IP interfaces.";
    }
  }

  // vlan
  container vlan {
    description
      "VLAN related platform capabilities.";
    leaf max-supported {
      type uint32;
      description
        "Max number of VLANs supported.";
    }

    leaf translation-supported {
      type boolean;
      description
        "True if VLAN translation is supported.";
    }

    leaf max-cross-connections {
      type uint32;
      description
        "Max number of VLAN cross connections.";
    }
  }

  // multicast services
  container multicast-services {
    description
      "Multicast service related platform capabilities.";

    leaf max-igmp-snoop-service-instances {
      type uint32;
      description
        "Max IGMP snoop service instances.";
    }
```


Copyright© 2022 Ciena® Corporation

```
    leaf max-mcast-groups {
      type uint32;
      description
        "Max Mcast groups.";
    }

    leaf max-mcast-source-addrs {
      type uint32;
      description
        "Max Mcast source addresses.";
    }

    leaf max-mcast-source-addrs-per-grp {
      type uint32;
      description
        "Max Mcast source addresses per group.";
    }

    leaf max-mcast-logical-interfaces {
      type uint32;
      description
        "Max Mcast logical interfaces.";
    }

    leaf max-mcast-member-interfaces {
      type uint32;
      description
        "Max Mcast member interfaces.";
    }
  }

  // spanning tree
  container spanning-tree {
    description
      "Spanning tree related platform capabilities.";

    leaf max-mstis {
      type uint32;
      description
        "Max MSTIs.";
    }

    leaf max-rstp-domains {
      type uint32;
      description
        "Max RSTP domains.";
    }
  }

  // pm
  container pm {
    if-feature ciena:pm;
    description
      "PM related platform capabilities.";

    leaf max-instances {
      type uint32;
      description
        "Max PM instances.";
    }
```


Copyright© 2022 Ciena® Corporation

```
          leaf max-bins {
            type uint32;
            description
              "Max PM bins.";
          }
        }

       // Protocol filters feature support should be returned by the netconf
server

        // vpls
        container vpls {
          if-feature ciena:vpls-fpga;
          description
            "VPLS related platform capabilities.";

          leaf max-tunnels-non-native-pbt {
            type uint32;
            description
              "Max tunnels non-native PBT";
          }

          leaf max-decap-tunnels-non-native-pbt {
            type uint32;
            description
              "Max decap tunnels non-native PBT";
          }

          leaf max-encap-tunnels-non-native-pbt {
            type uint32;
            description
              "Max encap tunnels non-native PBT";
          }

          leaf max-pbt-vcs {
            type uint32;
            description
              "Max PBT VCs";
          }
        }

        // mpls
        container mpls {
          if-feature ciena:mpls;
          description
            "MPLS related platform capabilities.";

          leaf max-ingress-tunnels {
            type uint32;
            description
              "Max MPLS ingress tunnels.";
          }

          leaf max-egress-tunnels {
            type uint32;
            description
              "Max MPLS egress tunnels.";
          }
```


Copyright© 2022 Ciena® Corporation

```
    leaf max-transit-tunnels {
      type uint32;
      description
        "Max MPLS transit tunnels.";
    }

    leaf max-vcs {
      type uint32;
      description
        "Max MPLS VCs.";
    }

    leaf max-l2vpns {
      type uint32;
      description
        "Max MPLS L2VPNs.";
    }

    leaf max-vss {
      type uint32;
      description
        "Max MPLS VSs.";
    }

    leaf max-vpws-vss {
      type uint32;
      description
        "Max MPLS VPWS VSs.";
    }

    leaf max-vpls-vss {
      type uint32;
      description
        "Max MPLS VPLS VSs.";
    }
  }

  // evc
  container evc {
    description
      "Ethernet virtual circuit related platform capabilities.";

    leaf max-evcs {
      type uint32;
      description
        "MAX Ethernet virtual circuits.";
    }
  }

  // vs
  container vs {
    description
      "Virtual switch related platform capabilities.";

    leaf max-default-vss {
      type uint32;
      description
        "Max default VSs.";
    }
```


Copyright© 2022 Ciena® Corporation

```
          leaf max-vss {
            type uint32;
            description
              "Max VSs.";
          }

          leaf max-vs-subs {
            type uint32;
            description
              "Max VS subs.";
          }
        }

        // l2cft
        container l2cft {
          description
            "L2-CFT related platform capabilities.";

          leaf max-l2cft-profiles {
            type uint32;
            description
              "Max L2-CFT profiles.";
          }
        }

        // vc l2 xforms
        container vc-l2-xforms {
          if-feature ciena:max-vc-l2xforms;
          description
            "VC L2 transform related platform capabilities.";

          leaf max-vc-l2-transforms {
            type uint32;
            description
              "Max VC L2 transforms.";
          }

          leaf max-vcs-with-l2-transform {
            type uint32;
            description
              "Max VCs with L2 transforms.";
          }
        }


       // egress bandwidth shaping support should be returned by the netconf
server

        container queue-groups {
          if-feature ciena:max-queue-groups;
          description
            "Queue group related platform capabilities.";

          leaf max-queue-groups {
            type uint32;
            description
              "Max queue groups.";
          }

          leaf max-queue-group-profiles {
```


Copyright© 2022 Ciena® Corporation

```
            type uint32;
            description
              "Max queue group profiles.";
          }

          leaf max-queue-group-instances {
            type uint32;
            description
              "Max queue group instances.";
          }

          leaf max-custom-queues {
            type uint32;
            description
              "Max custom queues.";
          }
        }

       // multi subs per port support should be returned by the netconf server

        // redundancy groups
        container redundancy-groups {
          description
            "Redundancy group related platform capabilities.";

          leaf max-redundancy-groups {
            type uint32;
            description
              "Max redundancy groups.";
          }

          leaf max-links-per-redundancy-group {
            type uint32;
            description
              "Max links per redundancy group.";
          }
        }

        // VPLS FPGA support should be returned by the netconf server
        // PBT FPGA support should be returned by the netconf server
        // L2 MAC table expansion support should be returned by the netconf
server

        container l2-learning {
          description
            "L2 learning related platform capabilities.";

          leaf max-dyna-sa-entries {
            type uint32;
            description
              "Max DYNA-SA entries.";
          }
        }

        container flow {
          description
            "Flow (Qos and Cos) related platform capabilities.";

          leaf max-fsmt-entries {
            type uint32;
```


Copyright© 2022 Ciena® Corporation

```
            description
              "Max FSMT entries.";
          }

          leaf max-srvc-lvl-entries {
            type uint32;
            description
              "Max SRVC-LVL entries.";
          }
        }

        container l2-access {
          description
          "L2 access (SAC, STATIC MAC, AGING) related platform capabilities.";

          leaf max-l2-sac-entries {
            type uint32;
            description
              "Max L2-SAC entries.";
          }

          leaf max-smt-entries {
            type uint32;
            description
              "Max SMT entries.";
          }
        }

        container network {
          description
            "Network related platform capabilities.";

          leaf max-epr-snids {
            type uint32;
            description
              "Max EPR snids.";
          }
        }

        container raps {
          if-feature ciena:raps;
          description
            "RAPS related platform capabilities.";

          leaf max-logical-rings {
            type uint32;
            description
              "Max logical rings.";
          }

          leaf max-virtual-rings {
            type uint32;
            description
              "Max virtual rings.";
          }
        }

        // Network sync, bits timing, gps timing, ptp timing, and benchmark
timing support should be returned
        // by the netconf server
```


Copyright© 2022 Ciena® Corporation

```
        container benchmark {
          if-feature ciena:benchmark;
          description
            "Benchmark related platform capabilities.";

          leaf max-bandwith {
            type uint32;
            units Mbps;
            description
              "Max benchmark bandwith in Mbps.";
          }
         // Benchmark reflector and generator support should be returned by
the netconf server

          leaf max-concurrent-tests {
            type uint32;
            description
              "Max benchmark concurrent tests.";
          }

          leaf max-profiles-config {
            type uint32;
            description
              "Max benchmark profiles config.";
          }

          leaf max-unique-active-cvids {
            type uint32;
            description
              "Max benchmark unique active CVIDs.";
          }
        }

        container acl-profiles {
          description
            "ACP profile related platform capabilities.";

          leaf max-profile-definitions {
            type uint32;
            description
              "Max ACL profile definitions.";
          }

          leaf max-rule-definitions {
            type uint32;
            description
              "Max ACL rule definitions.";
          }
        }

        container dhcp-relay {
          description
            "DHCP relay related platform capabilities.";

          leaf max-agents {
            type uint32;
            description
              "MAX DHCP relay agents.";
          }
```


Copyright© 2022 Ciena® Corporation

```
  }
}

// device-id
container device-id {
  description
    "Device ID information of the chassis.";

  leaf device-type {
    type uint32;
    description
      "Device type value.";
  }

  leaf part-number {
    type uint32;
    description
      "Chassis part number.";
  }

  leaf part-number-revision {
    type uint32;
    description
      "Chassis part number revision.";
  }

  leaf serial-number {
    type string;
    description
      "Chassis serial number.";
  }

  leaf manufactured-date {
    type yang:date-and-time;
    description
      "Date the chassis was manufactured.";
  }

  leaf mac-address {
    type yang:mac-address;
    description
      "Chassis MAC address.";
  }

  leaf param-version {
    type uint32;
    description
      "Chassis param version.";
  }
}

// fans
list fan-status {
  key fan-id;
  description
    "Fan speed status.";

  leaf fan-id {
    type uint32;
    description
```


Copyright© 2022 Ciena® Corporation

```
            "ID of the fan.";
        }

        leaf current-speed {
          type fan-speed;
          description
            "Current speed of the fan.";
        }

        leaf average-speed {
          type fan-speed;
          description
            "Average speed of the fan.";
        }

        leaf minimum-speed {
          type fan-speed;
          description
            "Minimum speed of the fan.";
        }
      }

    // mac
      container mac-partitions {
        description
          "Chassis MAC partitions.";

        container chassis {
          description
            "Chassis MAC partition.";
          uses mac-partition;
        }

        container local-mgmt {
          description
            "Local management MAC partition.";
          uses mac-partition;
        }

        container l3-interface {
          description
            "Layer 3 interface MAC partition.";
          uses mac-partition;
        }

        container port-base {
          description
            "Port base MAC partition.";
          uses mac-partition;
        }

        container benchmark {
          description
            "Benchmark MAC partition.";
          uses mac-partition;
        }

        container remote-mgmt {
          description
            "Remote management MAC partition.";
```


Copyright© 2022 Ciena® Corporation

```
      uses mac-partition;
    }
  }

  // power
  container power {
    description
      "Chassis power supply information.";

    // power supply status
    list status {
      key module;
      description
        "Power supply operational status.";

      leaf module {
        type string;
        description
          "Power supply module.";
      }

      leaf part-number {
        type string;
        description
          "Power supply part number.";
      }

      leaf type {
        type string;
        description
          "Type of power supply.";
      }

      leaf state {
        type hardware-state;
        description
          "State of the power supply.";
      }

      leaf fan {
        type hardware-state;
        description
          "State of the power supply's fan.";
      }

    }

    // power supply info
    list info {
      key module;
      description
        "Power supply related information.";

      leaf module {
        type string;
        description
          "Power supply module.";
      }

      leaf model-number {
```


Copyright© 2022 Ciena® Corporation

```
    type string;
    description
      "Model number of the power supply.";
  }

  leaf serial-number {
    type string;
    description
      "Serial number of the power supply.";
  }

  leaf revision {
    type string;
    description
      "Revision of the power supply.";
  }

  leaf clei-code {
    type string;
    description
      "CLEI code of the power supply.";
  }

  leaf manufactured-date {
    type string;
    description
      "Manufactured date of the power supply.";
  }

  leaf input {
    type power-supply-type;
    description
      "Type of power supply.";
  }

  leaf input-voltage {
    type string;
    units volts;
    description
      "Input voltage range of the power supply.";
  }

  leaf output-voltage {
    type string;
    units volts;
    description
      "Output voltage range of the power supply.";
  }

  leaf manufacturing-location {
    type string;
    description
      "Manufacturing location of the power supply.";
  }

  leaf checksum {
    type string;
    description
      "Checksum of the power supply.";
  }
```


Copyright© 2022 Ciena® Corporation

```
      }
    }

    // temperature
    container temperature {
      description
        "Chassis temperature information.";

      uses temperature-attributes;
      uses temperature-threshold-attributes;
    }
  }
}

/*
 * RPCs
 */

rpc post {
  description
    "Set the post state.";

  input {
    leaf state {
      type ciena:on-off-state;
      description
        "Set the post state.";
    }
  }

  output {
    leaf status {
      type string;
      description
        "Status of the reboot RPC.";
    }
  }
}

rpc force-reset {
  description
    "Force a reset of the chassis.";

  output {
    leaf status {
      type string;
      description
        "Status of the force reset.";
    }
  }
}

rpc reboot {
  description
    "Reboot the chassis.";

  input {
    choice schedule {
      description
```


Copyright© 2022 Ciena® Corporation

```
          "When the chassis is scheduled to reboot.";

        leaf now {
          type empty;
          description
            "Request the chassis to reboot now.";
        }
        leaf delay {
          type uint32;
          units seconds;
          description
            "Request the chassis to reboot after a delay.";
        }
        leaf time {
          type yang:date-and-time;
          description
            "Request the chassis to reboot at the specified date.
             The date may be at most one month away.";
        }
      }

      choice options {
        description
          "Additional reboot options.";

        leaf no-post {
          type empty;
          description
            "Disable POST at reboot.";
        }
        leaf cold-start {
          type empty;
          description
            "Start the system fresh, including POST as well as
             resetting all the HW components.";
        }
      }
    }

    output {
      leaf status {
        type string;
        description
          "Status of the reboot RPC.";
      }
    }
  }

  rpc reboot-cancel {
    description
      "Cancel a scheduled reboot.";

    output {
      leaf status {
        type string;
        description
          "Status of the reboot RPC.";
      }
    }
  }
```


Copyright© 2022 Ciena® Corporation

```
  rpc clear-post {
    description
      "Clear the POST errors;";

    output {
      leaf status {
        type string;
        description
          "Status of the clear post RPC.";
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-classifier.yang

```
module ciena-classifier {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-classifier";
    prefix "ciena-classifier";

    import ciena-mef-classifier {
        prefix classifier;
    }

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
                 Hanover, Maryland 21076
                 U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
        "This YANG module defines Ciena's management data definition for the
         management of a classifier.";

    revision "2017-05-16" {
        description
            "Added correct augment path";
    }

    revision "2017-02-03" {
        description
            "Added vlan ID list";
    }

    revision "2016-05-09" {
        description
            "Initial version";
    }

   augment "/classifier:classifiers/classifier:classifier/classifier:filter-
entry/classifier:filter-parameters/classifier:vtag-stack/classifier:vtag-
stack-type/classifier:tagged/classifier:vtags" {

        container ciena-vtags {

            leaf pcp-bits {
                type bits {
                    bit pcp-0;
                    bit pcp-1;
                    bit pcp-2;
                    bit pcp-3;
                    bit pcp-4;
                    bit pcp-5;
                    bit pcp-6;
                    bit pcp-7;
                }
                description
                    "A bit list of PCP values for the classifier.";
```


Copyright© 2022 Ciena® Corporation

```
            }

            leaf-list vlan-id {
                type uint16 {
                    range "0..4094";
                }
                max-elements 4;
                description
                    "A list of vlan-id values for the classifier.";
            }
        }
    }

   augment "/classifier:classifiers/classifier:classifier/classifier:filter-
entry" {

        container ciena-filter-entry {

            leaf dscp-bits {
                type bits {
                    bit dscp-0;
                    bit dscp-1;
                    bit dscp-2;
                    bit dscp-3;
                    bit dscp-4;
                    bit dscp-5;
                    bit dscp-6;
                    bit dscp-7;
                    bit dscp-8;
                    bit dscp-9;
                    bit dscp-10;
                    bit dscp-11;
                    bit dscp-12;
                    bit dscp-13;
                    bit dscp-14;
                    bit dscp-15;
                    bit dscp-16;
                    bit dscp-17;
                    bit dscp-18;
                    bit dscp-19;
                    bit dscp-20;
                    bit dscp-21;
                    bit dscp-22;
                    bit dscp-23;
                    bit dscp-24;
                    bit dscp-25;
                    bit dscp-26;
                    bit dscp-27;
                    bit dscp-28;
                    bit dscp-29;
                    bit dscp-30;
                    bit dscp-31;
                    bit dscp-32;
                    bit dscp-33;
                    bit dscp-34;
                    bit dscp-35;
                    bit dscp-36;
                    bit dscp-37;
                    bit dscp-38;
                    bit dscp-39;
```


Copyright© 2022 Ciena® Corporation

```
                    bit dscp-40;
                    bit dscp-41;
                    bit dscp-42;
                    bit dscp-43;
                    bit dscp-44;
                    bit dscp-45;
                    bit dscp-46;
                    bit dscp-47;
                    bit dscp-48;
                    bit dscp-49;
                    bit dscp-50;
                    bit dscp-51;
                    bit dscp-52;
                    bit dscp-53;
                    bit dscp-54;
                    bit dscp-55;
                    bit dscp-56;
                    bit dscp-57;
                    bit dscp-58;
                    bit dscp-59;
                    bit dscp-60;
                    bit dscp-61;
                    bit dscp-62;
                    bit dscp-63;
                }
                description
                    "A bit list of DSCP values for the classifier.";
            }
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-command-file.yang

```
module ciena-command-file {
  namespace "http://www.ciena.com/ns/yang/ciena-command-file";
  prefix ciena-command-file;

  import ciena-file-transfer-types {
     prefix ciena-ftt;
   }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module defines the nodes for
     software command file operations
     for the NETCONF protocol.

     Copyright (c) 2016 Ciena Corporation.  All rights reserved.

     All information contained herein is, and remains the property of
     Ciena Corporation. Dissemination of this information or
     reproduction of this material is strictly forbidden unless prior
     written permission is obtained from Ciena Corporation.";

  revision 2017-03-08 {
    description
      "Initial revision";
    reference
      "39XX/51XX Service Delivery, Aggregation and Virtualization
       Switches, Software Management and Licensing (009-3273-018)
       (Ciena Corporation, December 2016);
       RFC 6020: YANG - A Data Modeling Language for the Network
       Configuration Protocol (NETCONF).";
  }

  /*
   * Operational state data nodes
   */
  container command-file-state {
    config false;
    description
      "Information pertaining to the operational state of software
       command files.";

    leaf last-command-file {
      type string;
      description
        "The last command file successfully executed.";
    }
```


Copyright© 2022 Ciena® Corporation

```
    leaf last-configuration-file {
      type string;
      description
        "The last configuration file successfully processed via a
         command file.";
    }

    leaf file-transfer-status {
      type ciena-ftt:file-transfer-status;
      description
        "Status of the latest download request coming from NETCONF
         when downloading a command-file using an RPC. This does
         not apply to file transfers initiated by other sources.";
    }
  }

  /*
   * RPCs
   */
  rpc command-file-reset {
    description
      "Reset (clear) the name of the last command or configuration file
       successfully processed.  On success, returns <ok/>.  On failure,
       returns an <rpc-error> containing an <error-message> describing
       the reason for the failure.";

    input {
      leaf file {
        type enumeration {
          enum last-command-file {
            description
              "The last command file";
          }
          enum last-config-file {
            description
              "The last configuration file";
          }
        }
        mandatory true;
        description
          "Which file to be reset: the last command file or the last
           configuration file";
      }
    }
  }

  rpc command-file-run {
    description
      "Run a command file.  On success, returns <ok/>.  On
       failure, returns an <rpc-error> containing an <error-message>
       describing the reason for the failure.";

    input {
      uses ciena-ftt:file-transfer-parameters;
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-command-log.yang

```
module ciena-command-log {
  namespace "http://www.ciena.com/ns/yang/ciena-command-log";
  prefix "ciena-command-log";

  import ciena-types {
    prefix ciena;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
    E-mail:  yang@ciena.com
    Postal:  7035 Ridge Road
             Hanover, Maryland 21076
             U.S.A.
    Phone:   +1 800-921-1144
    Fax:     +1 410-694-5750";

  description
    "This YANG module defines the nodes for
     Command Logging for the NETCONF protocol.

    Copyright (c) 2016 Ciena Corporation.  All rights
         reserved.

         All information contained herein is, and remains
         the property of Ciena Corporation. Dissemination of this
         information or reproduction of this material is strictly
         forbidden unless prior written permission is obtained from
         Ciena Corporation.";

  revision "2016-02-22"
  {
      description
          "Initial version";
      reference
          "RFC 6020: YANG - A Data Modeling Language for
              the Network Configuration Protocol (NETCONF).
              No specific reference; standard not available.";
  }

  /*
   * Nodes with unresovled questions:
   * 1. config/destination
   * 2. oper/destination
   * 3. oper/log-file
   * 4.
   *
   */

  /*
   * Configuration data nodes
   */
  container command-log {
    description "";
```


Copyright© 2022 Ciena® Corporation

```
    leaf admin-state {
      type ciena:admin-state;
      description
        "The command log admin state.";
    }

    // PN8700-A13.1*> command-log show destination
    // Command Log Destination: flash
    // setable in 8.x; not in 6.x? oper in 6.x and config in 8.x?
    //
    leaf destination {
      if-feature ciena:saos-8x;

      type enumeration {
        enum flash {
          description "Use to chose destination as flash.";
        }
      }
      description "An enumeration of destination options. 8x allows setting
the destination.
                   Using an enum; as there is the possibility of more options
in future releases.
                    The memory/device where the command log will be stored,
i.e, its destination.";
    }

  } // end command-log container

  /*
   * Operational state data nodes
   */
  container command-log-state {
    config false;
    description "The command log state container.";

    leaf destination {
      if-feature ciena:saos-6x;

      type enumeration {
        enum flash {
          description "Use to chose destination as flash.";
        }
      }
      description "An enumeration of destination options. 6x allows viewing
the destination.
                   Using an enum; as there is the possibility of more options
in future releases.
                    The memory/device where the command log will be stored,
i.e, its destination.";
    }


    container log-file {
      description "";

      // command log show / command log file
      leaf command-log-contents { // or command-log-file?
        type binary;
        description
          "The command log file. Not tested.
```


Copyright© 2022 Ciena® Corporation

```
           The assumption is that this leaf will be able to send an entire
file to the user.
          This leaf is attempting to send the entire command log output from
'command-log show'. OR
             -- what files does the 'command-log show' command read? when
displayed on the CLI?
           Are we sending the command log files in /flash0/log:?
           cmdLog.0    cmdLog.1    cmdLog.2    cmdLog.3    eventLog.4

           The command log files can be extreamly long.
          Are there concerns about netconf server excessive load / possible
processing delay.
             -- can one request block other netconf server requests?


         The binary built-in type represents any binary data, i.e., a sequence
of octets.
          Binary values are encoded with the base64 encoding scheme. Base64
binary-to-text encoding schemes.
           Base 64 Encoding https://tools.ietf.org/html/rfc4648#section-4.";

      } // end command-log-... leaf

      leaf log-argument-type-6x {
        if-feature ciena:saos-6x;
        type enumeration {
          enum containing { // containing <String[1..128]>
            description "";
          }
          enum exit-status { // exit-status <NUMBER: 0..4294967295>
            description "";
          }
          enum failed {
            description "";
          }
         enum from-date { // from-date <date: yyyy-mm-dd or yy-mm-dd or mm-dd>
            description "";
          }
          enum from-time { // from-time <time: hh:mm:ss or hh:mm>
            description "";
          }
          enum last { //  last <duration: {N[yMwdhms]}* e.g. 1h10m3s>
            description "";
          }
          enum mine {
            description "";
          }
          enum on-terminal { // on-terminal <String[1..32]>
            description "";
          }
          enum privilege { // limited      admin        super        diag
            description "";
          }
          enum recent {
            description "";
          }
          enum sequence-id { //sequence-id <NUMBER: 1..4294967295[-
<range>][,...]>
            description "";
          }
```


Copyright© 2022 Ciena® Corporation

```
          enum since-activated {
            description "";
          }
          enum successful {
            description "";
          }
          enum tail { // tail <NUMBER: 1..4294967295>
            description "";
          }
          enum today {
            description "";
          }
          enum to-date { // to-date <date: yyyy-mm-dd or yy-mm-dd or mm-dd>
            description "";
          }
          enum to-time {  // to-time <time: hh:mm:ss or hh:mm>
            description "";
          }
          enum user {  //  user <String[1..32]>
            description "";
          }
          enum verbose {
            description "";
          }
          enum yesterday {
            description "";
          }
        }
        description "";

      } // end log-argument-type-6x leaf

      leaf log-argument-type-8x {
        if-feature ciena:saos-8x;
        type enumeration {
          enum containing { //containing <String[1..128]>
            description "";
          }
          enum exit-status { // exit-status <NUMBER: 0..4294967295>
            description "";
          }
         enum from-date { // from-date <date: yyyy-mm-dd or yy-mm-dd or mm-dd>
            description "";
          }
          enum from-time { //  from-time <time: hh:mm:ss or hh:mm>
            description "";
          }
         enum last {  //  last <duration: number/time format N[yMwdhms] e.g.
1h10m3s  for 1 hour, 10 minutes and 3 seconds>
            description "";
          }
          enum mine {
            description "";
          }
          enum on-terminal { // on-terminal <String[1..32]>
            description "";
          }
         enum privilege { //   limited      admin        super        diag
            description "";
          }
```


Copyright© 2022 Ciena® Corporation

```
          enum recent {
            description "";
          }
          enum sequence-id { //  sequence-id <NUMBER: 1..4294967295[,...]>
            description "";
          }
          enum since-activated {
            description "";
          }
          enum tail { // tail <NUMBER: 1..4294967295>
            description "";
          }
          enum today {
            description "";
          }
          enum to-date { //  to-date <date: yyyy-mm-dd or yy-mm-dd or mm-dd>
            description "";
          }
         enum to-time { // to-time <time: hh:mm:ss or hh:mm>
            description "";
          }
          enum user { // user <String[1..32]>
            description "";
          }
          enum verbose {
            description "";
          }
          enum yesterday {
            description "";
          }
        }
        description "";
      } // end log-argument-type-8x

      leaf augument-value {
        type string;
        description "The argument value.
                     This value will be taken as a string, and converted to
the type value.
                   The value will be checked for the format expected by the
argument type.";
      }

    } // end log-file container

   } // end command-log-state container

  /*
   * RPC
   */
  rpc command-log-clear {
    description "";
    output {
      leaf status {
        type string;
          description "";
      }
    }
  }
```


Copyright© 2022 Ciena® Corporation

```
} // end module ciena-command-log
```


Copyright© 2022 Ciena® Corporation

ciena-configuration-management.yang

```
module ciena-configuration-management {
  namespace
    "http://www.ciena.com/ns/yang/ciena-configuration-management";

  prefix "config";

  import ietf-system {
    prefix sys;
  }

  import ietf-yang-types {
    prefix yang;
  }

  import ciena-types {
    prefix ciena;
  }

  import ietf-inet-types {
    prefix inet;
  }

  import ciena-file-transfer-types {
    prefix ciena-ftt;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module defines Ciena's CLI configuration file options.

     Copyright (c) 2017 Ciena Corporation.  All rights
     reserved.

     All information contained herein is, and remains
     the property of Ciena Corporation. Dissemination of this
     information or reproduction of this material is strictly
     forbidden unless prior written permission is obtained from
     Ciena Corporation.";

  revision 2017-05-25 {
    description
      "Initial version";
    reference
      "39XX/51XX, SAOS 6.16, Base Configuration, Revision A  Standard,
       009-3273-008 - Configuration management document for SAOS 6.x
       base configuration.
```


Copyright© 2022 Ciena® Corporation

```
      8700 Packetwave Platform Base Configuration, Standard Revision
      A, 380-1874-310 - Configuration management document for SAOS
      8.x base configuration.";
}

/*
 * typedefs
 */

typedef config-filename {
  type string {
    length 1..128;
  }
  description
    "Standard configuration filename type.";
}

typedef config-system-status {
  type enumeration {
    enum idle {
      description
        "The configuration system is currently idle and ready to
         accept command.";
    }
    enum loading {
      description
        "The system is busy reading/writing the configuration file.";
    }
    enum augmenting {
      description
        "The system is busy augmenting a configuration file.";
    }
    enum checking {
      description
        "The system is busy checking a configuration file.";
    }
    enum unknown {
      description
        "The configuration system state is unknown.";
    }
  }
  description
    "Values for the configuration system status.";
}

typedef config-xftp-status {
  type enumeration {
    enum idle {
      description
        "No files are being downloaded at this time.";
    }
    enum busy {
      description
        "A file is currently being downloaded.";
    }
  }
  description
    "Configuration XFTP status values.";
}
```


Copyright© 2022 Ciena® Corporation

```
typedef config-check-status {
  type enumeration {
    enum none {
      description
        "No file has been checked.";
    }

    enum no-error {
      description
        "No error found while checking the config.";
    }

    enum error {
      description
        "Errors found while checking the config.";
    }
  }
  description
    "Configuration check status values.";
}

/*
 * Groupings
 */

/*
 * Identities
 */

/*
 * Features
 */

feature config-revert-timeout {
  description
    "This feature enables the use of configuration revert timeouts.";
}

feature config-reset-to-defaults {
  description
    "This feature enables the user to reset to defaults.";
}

feature config-strip-user-comments {
  description
    "This feature enables the user to strip user comments before
     saving a configuration file.";
}

feature config-backup {
  description
    "This feature enables the user to use configuration backup
     features.";
}

feature config-zero-touch {
  description
    "This feature enables the user to use config zero touch
     features.";
}
```


Copyright© 2022 Ciena® Corporation

```
  feature config-cancel-revert {
    description
      "This feature enables the user to cancel configuration
       reversion.";
  }

  /*
   * Configuration data nodes
   */
  augment /sys:system {
    description
      "SAOS configuration options.";

    container configuration {
      description
        "Configuration of the system configuration options.";

      leaf default-save-filename {
        type config-filename;
        description
          "Configuration default save filename.";
      }

      leaf default-load-filename {
        type config-filename;
        description
          "Configuration default load filename.";
      }

      leaf backup-load-filename {
        if-feature config-backup;
        type config-filename;
        description
          "Configuration backup load filename.";
      }

      container ztp {
        if-feature config-zero-touch;
        description
          "Configuration related ZTP information.";

        leaf admin-state {
          type ciena:admin-state;
          description
            "Admin zero-touch provisioning state.";
        }
      }
    }
  }

  /*
   * Operational state data nodes
   */
  augment /sys:system-state {
    description
      "SAOS operational data.";

    container configuration {
      config false;
```


Copyright© 2022 Ciena® Corporation

```
description
  "Operational data for this systems configuration options.";

leaf status {
  type config-system-status;
  description
    "Status of the configuration system.";
}

leaf loaded {
  type config-filename;
  description
    "Name of the configuration file loaded.";
}

leaf dirty {
  type boolean;
  description
    "True when the configuration is dirty.";
}

leaf-list config-files {
  type config-filename;
  description
    "List of configuration files on the system.";
}

container check-status {
  description
    "Status of the last config-check RPC operation.";

  leaf checked-file {
    type inet:uri;
    description
      "File information of the last checked config.";
  }

  leaf checked-status {
    type config-check-status;
    description
      "Status of the last checked config.";
  }
}

container ztp {
  if-feature config-zero-touch;
  description
    "Operational related ZTP information.";

  leaf oper-state {
    type ciena:oper-state;
    description
      "Operational zero-touch provisioning state";
  }

  leaf ztp-configured {
    type boolean;
    description
      "True when ZTP is configured.";
  }
```


Copyright© 2022 Ciena® Corporation

```
      }

      leaf xftp-status {
        type config-xftp-status;
        description
          "Status of the latest download request coming from NETCONF
           when downloading a config using an RPC. This does not
           apply to files transferred using the CLI or SNMP.";
      }
    }
  }

  /*
   * RPCs
   */

  rpc config-install {
    description
      "Download and install a new configuration file.";

    input {
      uses ciena-ftt:file-transfer-parameters;
    }
  }

  rpc config-augment {
    description
      "Augment the current configuration settings with the contents of
       the specified file.  This RPC will return immediately and the
       status of the augment will be indicated by the status operational
       leaf.";

    input {
      uses ciena-ftt:file-transfer-parameters;

      leaf revert-timeout {
        if-feature config-revert-timeout;
        type yang:date-and-time;
        description
          "Revert timeout.";
      }
    }
  }

  rpc config-check {
    description
      "Check a configuration file for errors.";

    input {
      uses ciena-ftt:file-transfer-parameters;
    }
  }


  rpc config-reset {
    description
      "Reset the configuration.  This RPC will reboot the device after
       issued.";

    input {
```


Copyright© 2022 Ciena® Corporation

```
      choice reset-type {
        mandatory true;
        case factory-defaults {
          leaf factory-defaults {
            type empty;
            description
              "Reset configuration to factory defaults.";
          }
        }
        case defaults {
          if-feature config-reset-to-defaults;
          leaf defaults {
            type empty;
            description
              "Reset configuration to defaults.";
          }
        }
        case user-config {
          leaf filename {
            type config-filename;
            description
              "Reset configuration to a user defined filename.";
          }
        }
        description
          "Configuration action to perform after the device resets.";
      }
    }
  }

  rpc config-save {
    description
      "Save the current running configuration";

    input {
      leaf filename {
        type config-filename;
        description
          "Save the current running configuration to a file. If
           omitted, the current running configuration will save to
           the default save configuration file.";
      }

      leaf include-default-settings {
        type empty;
        description
          "Include the default settings in the configuration file.";
      }

      leaf strip-logical-ids {
        type empty;
        description
          "Strip the logical IDs from the configuration before saving.";
      }

      leaf strip-user-comments {
        if-feature config-strip-user-comments;
        type empty;
        description
          "Strip user comments from the configuration before saving.";
```


Copyright© 2022 Ciena® Corporation

```
      }
    }
  }

  rpc config-notification-clear {
    description
      "Clear configuration notifications when logging into the CLI.";

    input {
      leaf backup-config-warning {
        if-feature config-backup;
        type empty;
        description
          "Cancel 'backup configuration file restored' warning";
      }

      leaf config-errors-warning {
        type empty;
        description
          "Cancel 'configuration errors during restore' warning";
      }

      leaf no-config-warning {
        type empty;
        description
          "Cancel 'no configuration file restored' warning";
      }

      leaf zero-touch-warning {
        if-feature config-zero-touch;
        type empty;
        description
          "Cancel 'Zero-Touch Provisioning configuration' warning";
      }
    }
  }

  rpc config-copy {
    description
      "Copy (duplicate) a configuration file";

    input {
      leaf source {
        type config-filename;
        mandatory true;
        description
          "Name of the source configuration file to copy.";
      }

      leaf dest {
        type config-filename;
        mandatory true;
        description
          "Name of the destination configuration file copied from
           source.";
      }
    }
  }

  rpc config-delete {
```


Copyright© 2022 Ciena® Corporation

```
    description
      "Delete a configuration file.";

    input {
      leaf filename {
        type config-filename;
        mandatory true;
        description
          "Name of a configuration file.";
      }
    }
  }

  rpc config-rename {
    description
      "Rename a configuration file";

    input {
      leaf from-filename {
        type config-filename;
        mandatory true;
        description
          "Name of the configuration file to rename.";
      }

      leaf to-filename {
        type config-filename;
        mandatory true;
        description
          "New name of the configuration file.";
      }
    }
  }

  rpc config-backup {
    if-feature config-backup;

    description
      "Backup current configuration to specified file on the local
       filesystem.";

    input {
      leaf filename {
        type config-filename;
        mandatory true;
        description
          "Backup filename.";
      }

      leaf strip-user-comments {
        type empty;
        description
          "Remove user comments before saving.";
      }

      leaf include-default-settings {
        type empty;
        description
          "Include the default settings when saving.";
      }
```


Copyright© 2022 Ciena® Corporation

```
    }
  }

  rpc config-cancel-revert {
    if-feature config-cancel-revert;
    description
      "Cancel automatic configuration reversion.";
  }

  rpc config-ztp-clear {
    if-feature config-zero-touch;
    description
      "Remove zero-touch configuration.";
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-cos-to-frame-map.yang

```
module ciena-cos-to-frame-map {
  namespace "http://www.ciena.com/ns/yang/ciena-cos-to-frame-map";
  prefix "ciena-ctf";

  import ciena-mef-cos-to-frame-map {
     prefix "ctf";
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
    E-mail:  yang@ciena.com
    Postal:  7035 Ridge Road
             Hanover, Maryland 21076
             U.S.A.
    Phone:   +1 800-921-1144
    Fax:     +1 410-694-5750";

  description
    "This YANG module defines augments for CoS to Frame Map.

    Copyright (c) 2017 Ciena Corporation.  All rights
         reserved.

         All information contained herein is, and remains
         the property of Ciena Corporation. Dissemination of this
         information or reproduction of this material is strictly
         forbidden unless prior written permission is obtained from
         Ciena Corporation.";

  revision "2017-06-30"
  {
      description
          "Initial version";
      reference
          "RFC 6020: YANG - A Data Modeling Language for
              the Network Configuration Protocol (NETCONF).
              No specific reference; standard not available.";
  }

  /*
   * Typedefs
   */

  typedef ctfMapTypes {
    type enumeration {
      enum global {
        description
          "Indicates that a map assigned this type will be able to modify
           any applicable and supported header fields based on the assigned
CoS.";
      }
      enum l2-only {
        description
         "Indicates that a map assigned this type will only be able to modify
```


Copyright© 2022 Ciena® Corporation

```
           L2 (i.e. Ethernet) header fields based on the assigned CoS for a
L2 Network Only.";
      }
    }
  }

  augment "/ctf:cos-to-frame-maps/ctf:cos-to-frame-map" {
    description
      "Augment Cos to Frame Map.";

    container ciena-cos-to-frame-map {
      leaf map-type {
        type ctfMapTypes;

        default global;

        description
          "Cos To Frame Map Type.";
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-cpu-rate-limit-sds.yang

```
module ciena-cpu-rate-limit-sds {
  namespace "http://www.ciena.com/ns/yang/ciena-sds-cpu-rate-limit";
  prefix "ciena-cpu-rate-limit-sds";

  import ietf-interfaces {
    prefix "if";
  }

  import ietf-yang-types {
    prefix yang;
  }

  import iana-if-type {
    prefix ianaift;
  }

  import ciena-types {
    prefix ciena;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
    E-mail:  yang@ciena.com
    Postal:  7035 Ridge Road
             Hanover, Maryland 21076
             U.S.A.
    Phone:   +1 800-921-1144
    Fax:     +1 410-694-5750";

  description
    "This YANG module defines the nodes for
     cpu rate limiting for the NETCONF protocol for 6x.

     Single Blade.

    Copyright (c) 2016 Ciena Corporation.  All rights
         reserved.

         All information contained herein is, and remains
         the property of Ciena Corporation. Dissemination of this
         information or reproduction of this material is strictly
         forbidden unless prior written permission is obtained from
         Ciena Corporation.";

  revision "2019-09-25"
  {
      description
          "Updated the maximum range value of CFM PPS on a port.";
      reference
          "No specific reference or standard available.";
  }
  revision "2018-11-29"
  {
      description
          "Updated Default value of LACP PPS on a port.";
```


Copyright© 2022 Ciena® Corporation

```
    reference
        "No specific reference or standard available.";
}
revision "2017-09-27"
{
    description
        "Updated Default value of LACP PPS on a port.";
    reference
        "No specific reference or standard available.";
}
revision "2016-08-22"
{
    description
        "Initial version";
    reference
        "No specific reference or standard available.";
}

/*
 * Groupings
 */

grouping cpu-rate-limit-statistics {
  description
    "Common cpu-rate-limit statistics.";

  leaf dropped {
    type yang:counter32;
    description "Number of packets dropped.";
  }

  leaf passed {
    type yang:counter32;
    description
      "Number of packets passed to CPU for processing.";
  }
}

/*
 * Configuration data nodes
 */
container cpu-rate-limiting-global {
    description
      "The global config for cpu-rate-limiting.";

    leaf admin-state {
      type ciena:admin-state;
      default disabled;
      description
        "The global admin state of cpu-rate-limiting.";
    }
}

augment "/if:interfaces/if:interface" {
      when "if:type = 'ianaift:ethernetCsmacd'" {
          description
              "Augment only if the interface type is ethernetCSmacd.";
      }

  description
```


Copyright© 2022 Ciena® Corporation

```
    "Port CPU rate-limiting configuration.";

  container cpu-rate-limiting {
    description
      "The port config for cpu-rate-limiting.";

    leaf admin-state {
      type ciena:admin-state;
      default enabled;
      description
        "The admin state of cpu-rate-limiting on this port.";
    }

    leaf arp {
      type uint32 {
        range "0..2500";
      }
      default 50;
      description
        "The arp PPS limit on this port.";
    }

    leaf bootp {
      type uint32 {
        range "0..2500";
      }
      default 5;
      description
        "The bootp/dhcp PPS limit on this port.";
    }

    leaf cfm {
      type uint32 {
        range "0..2500";
      }
      default 50;
      description
        "The cfm PPS limit on this port.";
    }

    leaf cft {
      type uint32 {
        range "0..2500";
      }
      default 250;
      description
        "The cft PPS limit on this port.";
    }

    leaf dhcpv6 {
      type uint32 {
        range "0..2500";
      }
      default 5;
      description
        "The dhcpv6 PPS limit on this port.";
    }

    leaf dot1x {
      type uint32 {
```


Copyright© 2022 Ciena® Corporation

```
        range "0..2500";
      }
      default 5;
      description
        "The dot1x PPS limit on this port.";
    }

    leaf eoam {
      type uint32 {
        range "0..2500";
      }
      default 5;
      description
        "The eoam PPS limit on this port.";
    }

    leaf esmc {
      type uint32 {
        range "0..2500";
      }
      default 40;
      description
        "The esmc PPS limit on this port.";
    }

    leaf igmp {
      type uint32 {
        range "0..2500";
      }
      default 25;
      description
        "The igmp PPS limit on this port.";
    }

    leaf ip-control {
      type uint32 {
        range "0..5000";
      }
      default 700;
      description
        "The ip-control PPS limit on this port.";
    }

    leaf ip-icmp {
      type uint32 {
        range "0..5000";
      }
      default 100;
      description
        "The ip-icmp PPS limit on this port.";
    }

    leaf ip-mgmt {
      type uint32 {
        range "0..5000";
      }
      default 700;
      description
        "The ip-mgmt PPS limit on this port.";
    }
```


Copyright© 2022 Ciena® Corporation

```
leaf ipv6-mgmt {
  type uint32 {
    range "0..5000";
  }
  default 700;
  description
    "The ipv6-mgmt PPS limit on this port.";
}

leaf ipv6-icmp {
  type uint32 {
    range "0..5000";
  }
  default 100;
  description
    "The ipv6-icmp PPS limit on this port.";
}

leaf lacp {
  type uint32 {
    range "0..2500";
  }
  default 21;
  description
    "The lacp PPS limit on this port.";
}

leaf lldp {
  type uint32 {
    range "0..2500";
  }
  default 5;
  description
    "The lldp PPS limit on this port.";
}

leaf lpbk {
  type uint32 {
    range "0..2500";
  }
  default 5;
  description
    "The lpbk PPS limit on this port.";
}

leaf pvst {
  type uint32 {
    range "0..2500";
  }
  default 5;
  description
    "The pvst PPS limit on this port.";
}

leaf raps {
  type uint32 {
    range "0..2500";
  }
  default 200;
```


Copyright© 2022 Ciena® Corporation

```
        description
          "The raps PPS limit on this port.";
      }

      leaf rmtlpbk {
        type uint32 {
          range "0..2500";
        }
        default 5;
        description
          "The rmtlpbk PPS limit on this port.";
      }

      leaf twamp {
        type uint32 {
          range "0..2500";
        }
        default 100;
        description
          "The twamp PPS limit on this port.";
      }

      leaf twamp-rsp {
        type uint32 {
          range "0..2500";
        }
        default 250;
        description
          "The twamp-rsp PPS limit on this port.";
      }

      leaf xstp {
        type uint32 {
          range "0..2500";
        }
        default 5;
        description
          "The xstp PPS limit on this port.";
      }

      leaf cicp {
        type uint32 {
          range "0..2500";
        }
        default 200;
        description
          "The cicp PPS limit on this port.";
      }

      leaf dflt {
        type uint32 {
          range "0..10000";
        }
        default 10;
        description
          "The default PPS limit on this port.";
      }
    }
  }
```


Copyright© 2022 Ciena® Corporation

```
/*
 * Operational state data nodes
 */

augment "/if:interfaces-state/if:interface" {
      when "if:type = 'ianaift:ethernetCsmacd'" {
          description
              "Augment only if the interface type is ethernetCSmacd.";
      }

  description
    "Port CPU rate-limiting operational data.";

  container cpu-rate-limiting-state {
    description "cpu-rate-limiting statistics.";

    container arp {
      description "arp statistics.";
      uses cpu-rate-limit-statistics;
    }

    container bootp {
      description "bootp/dhcp statistics.";
      uses cpu-rate-limit-statistics;
    }

    container cfm {
      description "cfm statistics.";
      uses cpu-rate-limit-statistics;
    }

    container cft {
      description "cft statistics.";
      uses cpu-rate-limit-statistics;
    }

    container dhcpv6 {
      description "dhcpv6 statistics.";
      uses cpu-rate-limit-statistics;
    }

    container dot1x {
      description "dot1x statistics.";
      uses cpu-rate-limit-statistics;
    }

    container eoam {
      description "eoam statistics.";
      uses cpu-rate-limit-statistics;
    }

    container esmc {
      description "esmc statistics.";
      uses cpu-rate-limit-statistics;
    }

    container igmp {
      description "igmp statistics.";
      uses cpu-rate-limit-statistics;
```


Copyright© 2022 Ciena® Corporation

```
}

container ip-control {
  description "ip-control statistics.";
  uses cpu-rate-limit-statistics;
}

container ip-icmp {
  description "ip-icmp statistics.";
  uses cpu-rate-limit-statistics;
}

container ip-mgmt {
  description "ip-mgmt statistics.";
  uses cpu-rate-limit-statistics;
}

container ipv6-mgmt {
  description "ipv6-mgmt statistics.";
  uses cpu-rate-limit-statistics;
}

container ipv6-icmp {
  description "ipv6-icmp statistics.";
  uses cpu-rate-limit-statistics;
}

container lacp {
  description "lacp statistics.";
  uses cpu-rate-limit-statistics;
}

container lldp {
  description "lldp statistics.";
  uses cpu-rate-limit-statistics;
}

container lpbk {
  description "lpbk statistics.";
  uses cpu-rate-limit-statistics;
}

container pvst {
  description "pvst statistics.";
  uses cpu-rate-limit-statistics;
}

container raps {
  description "raps statistics.";
  uses cpu-rate-limit-statistics;
}

container rmtlpbk {
  description "rmtlpbk statistics.";
  uses cpu-rate-limit-statistics;
}

container twamp {
  description "twamp statistics.";
  uses cpu-rate-limit-statistics;
```


Copyright© 2022 Ciena® Corporation

```
      }

      container twamp-rsp {
        description "twamp-rsp statistics.";
        uses cpu-rate-limit-statistics;
      }

      container xstp {
        description "xstp statistics.";
        uses cpu-rate-limit-statistics;
      }

      container cicp {
        description "cicp statistics.";
        uses cpu-rate-limit-statistics;
      }

      container dflt {
        description "default statistics.";
        uses cpu-rate-limit-statistics;
      }
    }
  }

  /*
   * RPC
   */

  rpc cpu-rate-limit-clear {
    description
      "Clear the cpu-rate-limit statistics.";

    input {
      leaf port {
        type if:interface-ref;
        description
          "The port to clear statistics on.";
      }
    }
    output {
      leaf status {
        type string;
        description
          "Status of the clear statistics operation.";
      }
    }
  }

}
```


Copyright© 2022 Ciena® Corporation

ciena-cpu-sds.yang

```
module ciena-cpu-sds {
   namespace "http://www.ciena.com/ns/yang/ciena-cpu-sds";
   prefix "ciena-cpu-sds";

   organization
      "Ciena Corporation";

   contact
      "Web URL:  http://www.ciena.com/
       E-mail:   yang@ciena.com
       Postal:   7035 Ridge Road
                 Hanover, Maryland 21076
                 U.S.A.
       Phone:    +1 800-921-1144
       Fax:      +1 410-694-5750";

   description
      "This YANG module defines the nodes for cpu  specific
       provisioning for the NETCONF protocol for 6x.

       Copyright (c) 2017 Ciena Corporation.  All rights
       reserved.

       All information contained herein is, and remains
       the property of Ciena Corporation. Dissemination of this
       information or reproduction of this material is strictly
       forbidden unless prior written permission is obtained from
       Ciena Corporation.";

   revision "2017-10-30" {
      description
         "Initial version";
      reference
         "RFC 6020: YANG - A Data Modeling Language for
          the Network Configuration Protocol (NETCONF).
          No specific reference, standard not available.";
   }

   container cpu-cos-remark {
      description
         "This is a container which holds configuration data for cpu cos
remarking.";

      leaf cpu-cos-remark-l2 {
         type enumeration {
            enum on {
               description
                  "cpu-cos-remark-l2 is on, allow remarking of cpu generated
packets.";
            }
            enum off {
               description
                   "cpu-cos-remark-l2 is off, do not allow remarking of cpu
generated packets.";
            }
         }
         default off;
         description
```

```
            "This is a leaf which holds the cpu-cos-remark-l2 configuration.";
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-dhcp.yang

```
module ciena-dhcp {
  namespace "http://www.ciena.com/ns/yang/ciena-dhcp";
  prefix dhcp;

  import ietf-inet-types {
    prefix inet;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This module contains the YANG definition of DHCP client and layer-2
     relay agent parameters per Ciena functional specifications
     106-0008-001_CES_DHCPClient_FS.doc and
     106-0009-001_CES_DHCP_Relay_FS.doc.

     Copyright (c) 2016 Ciena Corporation.  All rights reserved.

     All information contained herein is, and remains the property of
     Ciena Corporation. Dissemination of this information or
     reproduction of this material is strictly forbidden unless prior
     written permission is obtained from Ciena Corporation.";

  revision 2018-03-16 {
    description
      "Added release-on-link-transition leaf in dhcp-client container.";
    reference
      "Ciena Functional Specification.";
  }

  revision 2017-06-26 {
    description
      "Added '-time' to some leaves to make their purpose more clear.
       Changed some leaf-list nodes to leaf nodes because they could
       have only one element.  Renamed the
       dhcp-client-state/option-value/vivsi leaf-list to the
       dhcp-client/option-value/bootfile-uri leaf because the original
       name did not reflect the actual value.";
    reference
      "Ciena Functional Specification.";
  }

  revision 2017-04-25 {
    description
      "Added tpid-rotation to DHCP.  Updated the file to move
       operational state data containers up a level now that operational
       data is supported and better understood.";
    reference
```

```
    "Ciena Functional Specification.";
}

revision 2016-08-04 {
  description
    "Initial revision.";
  reference
    "RFC 2131, Dynamic Host Configuration Protocol
     RFC 2132, DHCP Options and BOOTP Vendor Extensions
     RFC 3046, DHCP Relay Agent Information Option
     RFC 3925, Vendor-Identifying Vendor Options for
               Dynamic Host Configuration Protocol version 4
               (DHCPv4)";
}

typedef state {
  type enumeration {
    enum enabled {
      description
      "The node is enabled.";
    }
    enum disabled {
      description
      "The node is disabled.";
    }
  }
  description
    "The state of the node.";
}

typedef admin-state {
  type state;
  description
    "The admin state of the node.";
}

typedef oper-state {
  type state;
  description
    "The oper state of the node.";
}

typedef on-off-state {
  type enumeration {
    enum off {
      description
        "The off state.";
    }
    enum on {
      description
        "The on state.";
    }
  }
  description
    "The on / off state.";
}

typedef relay-type {
  type enumeration {
    enum vlan {
```


Copyright© 2022 Ciena® Corporation

```
        description
          "Relay agent is attached to a VLAN.";
      }
      enum vs {
        description
          "Relay agent is attached to a virtual switch.";
      }
    }
    description
      "The type of network element to which the relay agent is
       attached.";
  }

  container dhcp-client {
    description
      "This is the top-level container for the DHCP client entity.  It
       contains DHCP client configuration values and the operational
       state container.";
    reference
      "RFC 2131, RFC 2132";

    leaf admin-state {
      type admin-state;
      description
        "The global administrative state of the DHCP client.";
    }

    leaf interface-type {
      type enumeration {
        enum active {
          description
            "Select the active interface.  This value applies only to
             devices that have an active interface.";
        }
        enum local {
          description
            "Select the local interface.  This value applies only to
             devices that have a local interface.";
        }
        enum remote {
          description
            "Select the remote interface.";
        }
      }
      description
        "The name of the interface to which the DHCP client is
         connected.  Two names are in common use for the non-remote
         interface:  active (SAOS 8.x) and local (SAOS 6.x).  For 8.x
         platforms, active is accepted and local is rejected with an
         error message; for 6.x platforms, local is accepted and active
         is rejected with an error message.  In addition, some 6.x
         platforms do not have a local interface.  For those, local will
         also be rejected with an error message.";
    }

    leaf discovery-interval {
      type uint8 {
        range "1..60";
      }
      units seconds;
```


Copyright© 2022 Ciena® Corporation

```
      description
        "The number of seconds between the transmission of successive
         Discover messages.";
    }

    leaf requested-lease-time {
      type uint32 {
        range "45..4294967294";
      }
      units seconds;
      description
        "The lease time requested from the server, in seconds.  Uses
         the IP Address Lease Time option, option 51.";
      reference
        "RFC 2132, Section 9.2";
    }

    leaf release-on-link-transition {
      type on-off-state;
      description
        "The administrative state of Dhcp client re-init on NNI link
transition.";
    }

    container option-enable {
      description
        "Specifies the option codes to be included in the Parameter
         Request List option, option 55, and to be used if received.
         (An option may be included in the Parameter Request List even
         if not enabled, but it will be ignored if received if it is
         not enabled.)";
      reference
        "RFC 2132, Section 9.8";

      leaf subnet-mask {
        type boolean;
        description
          "Subnet Mask option, option 1";
        reference
          "RFC 2132, Section 3.3";
      }

      leaf time-offset {
        type boolean;
        description
          "Time Offset option, option 2";
        reference
          "RFC 2132, Section 3.4";
      }

      leaf router {
        type boolean;
        description
          "Router Option, option 3";
        reference
          "RFC 2132, Section 3.5";
      }

      leaf domain-name-server {
        type boolean;
```


Copyright© 2022 Ciena® Corporation

```
    description
      "Domain Name Server Option, option 6";
    reference
      "RFC 2132, Section 3.8";
  }

  leaf log-server {
    type boolean;
    description
      "Log Server Option, option 7";
    reference
      "RFC 2132, Section 3.9";
  }

  leaf host-name {
    type boolean;
    description
      "Host Name Option, option 12";
    reference
      "RFC 2132, Section 3.14";
  }

  leaf domain-name {
    type boolean;
    description
      "Domain Name option, option 15";
    reference
      "RFC 2132, Section 3.17";
  }

  leaf ntp-servers {
    type boolean;
    description
      "Network Time Protocol (NTP) Servers Option, option 42";
    reference
      "RFC 2132, Section 8.3";
  }

  leaf lease-time {
    type boolean;
    description
      "IP Address Lease Time, option 51";
    reference
      "RFC 2132, Section 9.2";
  }

  leaf tftp-server-name {
    type boolean;
    description
      "TFTP Server Name option, option 66";
    reference
      "RFC 2132, Section 9.4";
  }

  leaf bootfile-name {
    type boolean;
    description
      "Bootfile Name option, option 67";
    reference
      "RFC 2132, Section 9.5";
```


Copyright© 2022 Ciena® Corporation

```
      }

      leaf vivsi {
        type boolean;
        description
          "Vendor-Identifying Vendor-Specific Information Option,
           option 125";
        reference
          "RFC 3925, Section 4";
      }
    }
  }

  container dhcp-client-state {
    config false;
    description
      "DHCP client operational values.";
    reference
      "RFC 2131, RFC 2132";

    leaf oper-state {
      type oper-state;
      description
        "The global operational state of the DHCP client.";
    }

    leaf dhcp-state {
      type string;
      description
        "The state of the DHCP client, as described in RFC 2131.";
    }

    leaf lease-time {
      type uint32;
      units seconds;
      description
        "The duration of the current lease in seconds.  The received
         value of the IP Address Least Time option, option 51.";
      reference
        "RFC 2132, Section 9.2";
    }

    leaf lease-time-remaining {
      type uint32;
      units seconds;
      description
        "The remaining time of the current lease in seconds.";
    }

    leaf renewal-time {
      type uint32;
      units seconds;
      description
        "The initial time before the current lease is renewed (T1), in
         seconds.";
    }

    leaf renewal-time-remaining {
      type uint32;
      units seconds;
```


Copyright© 2022 Ciena® Corporation

```
    description
      "The remaining time before the current lease is renewed (T1),
       in seconds.";
  }

  leaf rebinding-time {
    type uint32;
    units seconds;
    description
      "The initial time before the current lease is rebound (T2), in
       seconds.";
  }

  leaf rebinding-time-remaining {
    type uint32;
    units seconds;
    description
      "The remaining time before the current lease is rebound (T2),
       in seconds.";
  }

  leaf dhcp-server-identifier {
    type inet:ipv4-address;
    description
      "The IP address of the server providing the current lease.
       The received value of the Server Identifier option, option
       54.";
    reference
      "RFC 2132, Section 9.7";
  }

  container option-value {
    description
      "Contains the values of options received.";

    leaf subnet-mask {
      type inet:ipv4-address;
      description
        "Subnet Mask option, option 1";
      reference
        "RFC 2132, Section 3.3";
    }

    leaf time-offset {
      type int32;
      units seconds;
      description
        "Time Offset option, option 2.  Offset from Coordinated
         Universal Time (UTC), in seconds.";
      reference
        "RFC 2132, Section 3.4";
    }

    leaf-list router {
      type inet:ipv4-address;
      max-elements 63;
      description
        "Router Option, option 3";
      reference
        "RFC 2132, Section 3.5";
```


Copyright© 2022 Ciena® Corporation

```
}

leaf-list domain-name-server {
  type inet:ipv4-address;
  max-elements 63;
  description
    "Domain Name Server Option, option 6";
  reference
    "RFC 2132, Section 3.8";
}

leaf-list log-server {
  type inet:ipv4-address;
  max-elements 63;
  description
    "Log Server Option, option 7";
  reference
    "RFC 2132, Section 3.9";
}

leaf host-name {
  type string;
  description
    "Host Name Option, option 12";
  reference
    "RFC 2132, Section 3.14";
}

leaf domain-name {
  type string;
  description
    "Domain Name option, option 15";
  reference
    "RFC 2132, Section 3.17";
}

leaf-list ntp-servers {
  type inet:ipv4-address;
  max-elements 63;
  description
    "Network Time Protocol (NTP) Servers Option, option 42";
  reference
    "RFC 2132, Section 8.3";
}

leaf lease-time {
  type uint32;
  units seconds;
  description
    "IP Address Lease Time, option 51, in seconds";
  reference
    "RFC 2132, Section 9.2";
}

leaf tftp-server-name {
  type string;
  description
    "TFTP Server Name option, option 66";
  reference
    "RFC 2132, Section 9.4";
```


Copyright© 2022 Ciena® Corporation

```
      }

      leaf bootfile-name {
        type string;
        description
          "Bootfile Name option, option 67";
        reference
          "RFC 2132, Section 9.5";
      }

      leaf bootfile-uri {
        type string;
        description
          "Ciena Bootfile URI Sub-Option (16) of the Vendor-Identifying
           Vendor-Specific Information Option, option 125";
        reference
          "RFC 3925, Section 4";
      }
    }
  }

  container dhcp-tpid-rotation {
    description
      "Top level container for the DHCP TPID rotation configuration.";

    leaf state {
      type on-off-state;
      description
        "The state of the TPID rotation controller.  When 'on', the
         egress TPID used by the DHCP and DHCPv6 clients while searching
         for an appropriate server on the remote interface is
         periodically rotated to the next TPID value in the sequence
         0x8100, 0x88a8, 0x9100, and untagged.";
    }
  }

  container dhcp-tpid-rotation-state {
    config false;
    description
      "DHCP TPID rotation operational state.";
    leaf last-tpid {
      type enumeration {
        enum tpid-none {
          description
            "No TPID has been set by the TPID rotation controller.
             Egress frames use the default TPID.";
        }
        enum tpid-untagged {
          description
            "Egress frames are untagged.";
        }
        enum tpid-8100 {
          description
            "Egress frames use TPID 0x8100";
        }
        enum tpid-88a8 {
          description
            "Egress frames use TPID 0x88a8";
        }
        enum tpid-9100 {
```


Copyright© 2022 Ciena® Corporation

```
          description
            "Egress frames use TPID 0x9100";
        }
      }
      description
        "The value of the last used TPID in the rotation.";
    }
  }

  container dhcp-l2-relay-agent {
    description
      "This is the top-level container for the DHCP layer-2 relay agent
       entity.";
    reference
      "RFC 3046";

    leaf admin-state {
      type admin-state;
      description
        "The global administrative state of the DHCP L2 relay agent.";
    }

    leaf circuit-id-type {
      type enumeration {
        enum cid-string {
          description
            "The circuit is identified by a string entered by the
             user.";
        }
        enum slot-port {      // 6.x only
          description
            "The circuit is identified by the slot and port number of
             the device port to which it is connected.";
        }
        enum slot-port-vlan { // 6.x only
          description
            "The circuit is identified by the slot and port number of
             the device port to which it is connected, and the VLAN
             ID.";
        }
        enum li-vs {          // 8.x only
          description
            "The circuit is identified by the logical interface and
             virtual switch ID of the logical interface to which it is
             connected.";
        }
      }
      description
        "Set the type of data contained in the Agent Circuit ID
         Sub-option.";
      reference
        "RFC 3046, Section 3.1";
    }

    leaf remote-id-type {
      type enumeration {
        enum device-hostname {
          description
            "The remote host is identified by the host name of this
             device.";
```


Copyright© 2022 Ciena® Corporation

```
      }
      enum device-mac {
        description
          "The remote host is identified by the MAC address of this
           device.";
      }
      enum rid-string {
        description
          "The remote host is identified by a string entered by the
           user.";
      }
    }
    description
      "Sets the type of data contained in the Agent Remote ID
       Sub-option.";
    reference
      "RFC 3046, Section 3.2";
  }

  leaf replace-option82 {
    type on-off-state;
    description
      "Specifies whether or not the relay agent should replace
       an existing option 82.";
  }

  list relay-agent {
    key "relay-type relay-id";

    leaf relay-type {
      type relay-type;
      description
        "The type of network element to which the relay agent is
         attached.";
    }

    leaf relay-id {
      type string;
      description
        "The name or VID of the VLAN, VS or MPLS-VC to which this
         relay agent is attached.";
    }

    leaf admin-state {
      type admin-state;
      description
        "The administrative state of this DHCP relay agent.";
    }

    description
      "List of relay agents and their configuration data.";
  }

  list relay-interface {
    key "relay-type relay-id interface-id sub-vlan";

    leaf relay-type {
      type relay-type;
      description
        "The type of network element to which the relay interface is
```


Copyright© 2022 Ciena® Corporation

```
      attached.";
  }

  leaf relay-id {
    type string;
    description
      "The name or VID of the VLAN, VS or MPLS-VC to which this
       relay agent is attached.";
  }

  leaf interface-id {
    type string;
    description
      "The name or port number of this relay agent interface.";
  }

  leaf sub-vlan {
    type uint32 {
      range "0..4094";
    }
    description
      "The VID of the sub-VLAN associated with this relay agent
       interface.  Zero means none.";
  }

  leaf trust-mode {
    type enumeration {
      enum client-trusted {
        description
          "Trust clients on this logical interface.";
      }
      enum server-trusted {
        description
          "Trust servers on this logical interface.";
      }
      enum dualrole-trusted {
        description
          "Trust both clients and servers on this logical
           interface.";
      }
      enum untrusted {
        description
          "Trust neither clients nor servers on this logical
           interface.";
      }
    }
    description
      "Grouping of relay interface attributes common to all
       interface types.";
  }

  leaf cid-string {
    type string {
      length "0..64";
    }
    description
      "Set the value of the Agent Circuit ID Sub-option for this
       logical interface as a string.";
    reference
      "RFC 3046, Section 3.1";
```


Copyright© 2022 Ciena® Corporation

```
      }

      leaf rid-string {
        type string {
          length "0..64";
        }
        description
          "Set the value of the Agent Remote ID Sub-option for this
           logical interface as a string.";
        reference
          "RFC 3046, Section 3.2";
      }

      description
        "List of relay agents interfaces and their configuration data.";
    }
  }

  container dhcp-l2-relay-agent-state {
    config false;
    description
      "DHCP L2 relay agent operational values.";

    list relay-agent {
      key "relay-type relay-id";

      leaf relay-type {
        type relay-type;
        description
          "The type of network element to which the relay agent is
           attached.";
      }

      leaf relay-id {
        type string;
        description
          "The name or VID of the VLAN, VS or MPLS-VC to which this
           relay agent is attached.";
      }

      leaf oper-state {
        type oper-state;
        description
          "The operational state of thise DHCP relay agent.";
      }

      description
        "List of relay agents and their configuration data.";
    }

    container statistics {
      config false;
      description
        "DHCP L2 relay agent statistics.";

      // Global statistics

      leaf forwarded {
        type uint32;
        description
```


Copyright© 2022 Ciena® Corporation

```
      "The number of DHCP packets forwarded rather than relayed.
       The are valid DHCP frames that didn't need to be processed by
       the relay.";
  }

  leaf relayed {
    type uint32;
    description
      "The number of DHCP packets processed by the relay and not
       dropped.";
  }

  leaf dropped {
    type uint32;
    description
      "The number of DHCP packets dropped by the relay.";
  }

  leaf not-for-relay {
    type uint32;
    description
      "The number of DHCP packets passed to the relay but determined
       to not belong to the relay.  These packets were dropped.";
  }

  // Relay instance statistics

  list relay-agent {
    key "relay-type relay-id";

    leaf relay-type {
      type relay-type;
      description
        "The type of network element to which the relay agent is
         attached.";
    }

    leaf relay-id {
      type string;
      description
        "The name or VID of the VLAN, VS or MPLS-VC to which this
         relay agent is attached.";
    }

    leaf for-relay {
      type uint32;
      description
        "The number of packets received by the relay agent instance
         that were actually intended for a relay.";
    }

    leaf relayed-client-messages {
      type uint32;
      description
        "The number of messages received from a DHCP client that
         were relayed.";
    }

    leaf relayed-server-messages {
      type uint32;
```


Copyright© 2022 Ciena® Corporation

```
    description
      "The number of messages received from a DHCP server that
       were relayed.";
  }

  leaf option-82-added {
    type uint32;
    description
      "The number of packets that have had option 82 added.";
  }

  leaf option-82-removed {
    type uint32;
    description
      "The number of packets that have had option 82 removed.";
  }

  leaf option-82-replaced {
    type uint32;
    description
      "The number of packets that have had option 82 replaced.";
  }

  leaf option-82-adding-would-have-exceeded-mtu {
    type uint32;
    description
      "The number of packets that should have had option 82 added
       but doing so would have exceeded the MTU.  The packets are
       forwarded without option 82 added.";
  }

  leaf dropped-received-on-untrusted-client-interfaces {
    type uint32;
    description
      "The number of packets dropped because they were received on
       untrusted client interfaces.";
  }

  leaf dropped-received-on-untrusted-server-interfaces {
    type uint32;
    description
      "The number of packets dropped because they were received on
       untrusted server interfaces.";
  }

  leaf dropped-spoofed-dhcp-packets {
    type uint32;
    description
      "The number of packets dropped because they were spoofed.";
  }

  leaf dropped-no-trusted-server-interfaces {
    type uint32;
    description
      "The number of packets dropped because there were no trusted
       server interfaces configured.";
  }

  leaf dropped-no-trusted-client-interfaces {
    type uint32;
```


Copyright© 2022 Ciena® Corporation

```
          description
            "The number of packets dropped because there were no trusted
             client interfaces configured.";
        }

        leaf dropped-relay-configuration {
          type uint32;
          description
            "The number of packets dropped because the configuration of
             that relay interface was invalid.  This usually means that
             the circuit-id-type was set to cid-string but the interface
             had no cid-string configured, or the remote-id-type was set
             to rid-string but the interface had no rid-string
             configured."; }

        leaf general-errors {
          type uint32;
          description
            "The number general errors encountered by the L2 relay
             agent.";
        }

        description
          "List of relay agents and their statistics.";
      }
    }
  }

  rpc dhcp-client-lease-renew {
    description
      "Renew the lease.";
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-dhcpv6.yang

```
module ciena-dhcpv6 {
  namespace "http://www.ciena.com/ns/yang/ciena-dhcpv6";
  prefix dhcpv6;

  import ietf-inet-types {
    prefix inet;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This module contains the YANG definition of DHCPv6 client
     parameters per Ciena functional specification
     106-0054-001_CES_DHCPv6_FS.doc.

     Copyright (c) 2016 Ciena Corporation.  All rights reserved.

     All information contained herein is, and remains the property of
     Ciena Corporation. Dissemination of this information or
     reproduction of this material is strictly forbidden unless prior
     written permission is obtained from Ciena Corporation.";

  revision 2017-06-26 {
    description
      "Moved dhcp-client/operational-state node up one level and renamed
       dhcp-client-state.  Added '-time' to some leaves to make their
       purpose more clear.  Changed some leaf-list nodes to leaf nodes
       because they could have only one element.  Corrected the type of
       the dhcpv6-server-identifier leaf from inet:ipv6-address to
       string.  Changed the type of the ra-flags leaf from string to
       bits because bits is a better type for the purpose.  Added the
       ra-received leaf because this information was missing.";
    reference
      "Ciena Guide for YANG, revised 2017-01-23";
  }

  revision 2017-04-13 {
    description
      "Initial revision.";
    reference
      "RFC 3315, Dynamic Host Configuration Protocol for IPv6 (DHCPv6)
       RFC 6221, Lightweight DHCPv6 Relay Agent
       RFC 4649, Dynamic Host Configuration Protocol for IPv6 (DHCPv6)
       Relay Agent Remote-ID Option";
  }

  typedef state {
    type enumeration {
```

```
      enum enabled {
        description
          "The node is enabled.";
      }
      enum disabled {
        description
          "The node is disabled.";
      }
    }
    description
      "The state of the node.";
  }

  typedef admin-state {
    type state;
    description
      "The admin state of the node.";
  }

  typedef oper-state {
    type state;
    description
      "The oper state of the node.";
  }

  typedef on-off-state {
    type enumeration {
      enum off {
        description
          "The off state.";
      }
      enum on {
        description
          "The on state.";
      }
    }
    description
      "The on / off state.";
  }

  typedef relay-type {
    type enumeration {
      enum vlan {
        description
          "Relay agent is attached to a VLAN.";
      }
      enum vs {
        description
          "Relay agent is attached to a virtual switch.";
      }
    }
    description
      "The type of network element to which the relay agent is
       attached.";
  }

  container dhcpv6-client {
    description
      "This is the top-level container for the DHCPv6 client entity.  It
       contains DHCPv6 client configuration values and the operational
```


Copyright© 2022 Ciena® Corporation

```
     state container.";
  reference
    "RFC 3315";

  leaf admin-state {
    type admin-state;
    description
      "The global administrative state of the DHCPv6 client.";
  }

  leaf interface-type {
    type enumeration {
      enum active {
        description
          "Select the active interface.  This value applies only to
           devices that have an active interface.";
      }
      enum local {
        description
          "Select the local interface.  This value applies only to
           devices that have a local interface.";
      }
      enum remote {
        description
          "Select the remote interface.";
      }
    }
    description
      "The name of the interface to which the DHCPv6 client is
       connected.  Two names are in common use for the non-remote
       interface:  active (SAOS 8.x) and local (SAOS 6.x).  For 8.x
       platforms, active is accepted and local is rejected with an
       error message; for 6.x platforms, local is accepted and active
       is rejected with an error message.  In addition, some 6.x
       platforms do not have a local interface.  For those, local will
       also be rejected with an error message.";
  }

  leaf rapid-commit-state {
    type admin-state;
    description
      "The rapid commit administrative state.  When enabled, the
       client includes the Rapid Commit Option (option code 14) in its
       Solicit message and is prepared to perform the Solicit-Reply
       message exchange.";
    reference
      "RFC 3315 Dynamic Host Configuration Protocol for IPv6
       (DHCPv6), Section 22.14 Rapid Commit Option";
  }

  leaf requested-preferred-lifetime {
    type uint32;
    units seconds;
    must "(. = 0) or
          (../requested-valid-lifetime = 0) or
          (. <= ../requested-valid-lifetime)" {
      error-message
        "Preferred-lifetime cannot exceed valid-lifetime when both" +
        " are defined and non-zero";
      description
```


Copyright© 2022 Ciena® Corporation

```
        "Either or both requested-preferred-lifetime or
         requested-valid-lifetime can be zero (meaning user has no
         requested value) or absent, but if both are defined and
         non-zero then requested-preferred-lifetime must be <=
         requested-valid-lifetime.";
    }
    description
      "The IA (Identity Assignment) preferred lifetime requested by
       the client.";
    reference
      "RFC 3315 Dynamic Host Configuration Protocol for IPv6 (DHCPv6);
       RFC 4862 IPv6 Stateless Address Autoconfiguration";
  }

  leaf requested-valid-lifetime {
    type uint32;
    units seconds;
    description
      "The IA (Identity Assignment) valid lifetime requested by the
       client.";
    reference
      "RFC 3315 Dynamic Host Configuration Protocol for IPv6 (DHCPv6);
       RFC 4862 IPv6 Stateless Address Autoconfiguration";
  }

  container option-enable {
    description
      "Specifies the option codes to be included in the Option Request
       Option (option code 6) and to be used if received.";
    reference
      "RFC 3315, Section 22.7 Option Request Option";

    leaf dns-server-list {
      type boolean;
      description
        "DNS Recursive Name Server option, option code 23";
      reference
        "RFC 3646 DNS Configuration options for Dynamic Host
         Configuration Protocol for IPv6 (DHCPv6), Section 3 DNS
         Recursive Name Server Option";
    }

    leaf domain-search-list {
      type boolean;
      description
        "Domain Search List option, option code 24";
      reference
        "RFC 3646 DNS Configuration options for Dynamic Host
         Configuration Protocol for IPv6 (DHCPv6), Section 4 Domain
         Search List Option";
    }

    leaf posix-timezone {
      type boolean;
      description
        "New POSIX Timezone Option, option code 41";
      reference
        "RFC 4833 Timezone Options for DHCP, Section 3";
    }
```


Copyright© 2022 Ciena® Corporation

```
      leaf tzdb-timezone {
        type boolean;
        description
          "New TZDB Timezone Option, option code 42";
        reference
          "RFC 4833 Timezone Options for DHCP, Section 3";
      }

      leaf ntp-server {
        type boolean;
        description
          "NTP (Network Time Protocol) Server Option, option code 56";
        reference
          "RFC 5908 Network Time Protocol (NTP) Server Option for
           DHCPv6";
      }

      leaf boot-file-url {
        type boolean;
        description
          "Boot File Uniform Resource Locator (URL) Option, option code
           59";
        reference
          "RFC 5970 DHCPv6 Options for Network Boot, Section 3.1";
      }
    }
  }

  container dhcpv6-client-state {
    config false;
    description
      "DHCPv6 client operational values.";
    reference
      "RFC 3315";

    leaf oper-state {
      type oper-state;
      description
        "The global operational state of the DHCPv6 client.";
    }

    leaf dhcpv6-state {
      type string;
      description
        "The state of the DHCPv6 client, as described in RFC 2131.";
    }

    leaf autoconfig-state {
      type string;
      description
        "The current autoconfiguration state according to the latest
         values of the Router Advertisement M and O flags:  Disabled,
         Stateful or Stateless.";
      reference
        "RFC 4862 IPv6 Stateless Address Autoconfiguration";
    }

    leaf ra-received {
      type boolean;
      description
```


Copyright© 2022 Ciena® Corporation

```
      "True if a Router Advertisement (RA) message has been received
       and the value of ra-flags is valid.";
  }

  leaf ra-flags {
    type bits {
      bit M {
        description
          "Managed Address Configuration Flag";
      }
      bit O {
        description
          "Other Configuration Flag";
      }
    }
    description
      "The latest values of the Router Advertisement M and O flags.
       Valid if ra-received is True. Other flags are defined in RFC
       5175 but only the M and O flags affect the behavior of
       DHCPv6.";
    reference
      "RFC 5175 IPv6 Router Advertisement Flags Option";
  }

  leaf renewal-time {
    type uint32;
    units seconds;
    description
      "The initial time before the current lease is renewed (T1), in
       seconds.";
  }

  leaf renewal-time-remaining {
    type uint32;
    units seconds;
    description
      "The remaining time before the current lease is renewed (T1),
       in seconds.";
  }

  leaf rebinding-time {
    type uint32;
    units seconds;
    description
      "The initial time before the current lease is rebound (T2), in
       seconds.";
  }

  leaf rebinding-time-remaining {
    type uint32;
    units seconds;
    description
      "The remaining time before the current lease is rebound (T2),
       in seconds.";
  }

  leaf preferred-lifetime {
    type uint32;
    units seconds;
    description
```


Copyright© 2022 Ciena® Corporation

```
      "The Preferred Lifetime of the current IA.";
    reference
      "RFC 3315 Dynamic Host Configuration Protocol for IPv6
       (DHCPv6);
       RFC 4862 IPv6 Stateless Address Autoconfiguration";
  }

  leaf preferred-lifetime-remaining {
    type uint32;
    units seconds;
    description
      "The remaining Preferred Lifetime of the current IA.";
    reference
      "RFC 3315 Dynamic Host Configuration Protocol for IPv6
       (DHCPv6);
       RFC 4862 IPv6 Stateless Address Autoconfiguration";
  }

  leaf valid-lifetime {
    type uint32;
    units seconds;
    description
      "The Valid Lifetime of the current IA.";
    reference
      "RFC 3315 Dynamic Host Configuration Protocol for IPv6
       (DHCPv6);
       RFC 4862 IPv6 Stateless Address Autoconfiguration";
  }

  leaf valid-lifetime-remaining {
    type uint32;
    units seconds;
    description
      "The remaining Valid Lifetime of the current IA.";
    reference
      "RFC 3315 Dynamic Host Configuration Protocol for IPv6
       (DHCPv6);
       RFC 4862 IPv6 Stateless Address Autoconfiguration";
  }

  leaf dhcpv6-server-identifier {
    type string;
    description
      "The DHPC Unique Identifier (DUID) identifying the server
       providing the current IA, as received in the Server Identifier
       Option, option code 2.";
    reference
      "RFC 3315, Section 9 DHCP Unique Identifier (DUID)
       RFC 3315, Section 22.3 Server Identifier Option";
  }

  container option-value {
    description
      "Contains the values of options received.";

    leaf-list dns-server-list {
      type inet:ipv6-address;
      max-elements 4095;
      description
        "DNS Recursive Name Server option, option code 23";
```


Copyright© 2022 Ciena® Corporation

```
        reference
          "RFC 3646 DNS Configuration options for Dynamic Host
           Configuration Protocol for IPv6 (DHCPv6), Section 3 DNS
           Recursive Name Server Option";
      }

      leaf-list domain-search-list {
        type string;
        description
          "Domain Search List option, option code 24";
        reference
          "RFC 3646 DNS Configuration options for Dynamic Host
           Configuration Protocol for IPv6 (DHCPv6), Section 4 Domain
           Search List Option";
      }

      leaf posix-timezone {
        type string;
        description
          "New POSIX Timezone Option, option code 41";
        reference
          "RFC 4833 Timezone Options for DHCP, Section 3";
      }

      leaf tzdb-timezone {
        type string;
        description
          "New TZDB Timezone Option, option code 42";
        reference
          "RFC 4833 Timezone Options for DHCP, Section 3";
      }

      leaf-list ntp-server {
        // REVISIT gajohnso 2016-07-12:  It is not clear what the type
        // of this option should be or how it should be presented.
        // The option value consists of a list of suboptions, and each
        // suboption may be either an IPv6 unicast address, an IPv6
        // multicast address, or an FQDN.
        type inet:ipv6-address;
        description
          "NTP (Network Time Protocol) Server Option, option code 56";
        reference
          "RFC 5908 Network Time Protocol (NTP) Server Option for
           DHCPv6, Section 4";
      }

      leaf boot-file-url {
        type string;
        description
          "Boot File Uniform Resource Locator (URL) Option, option
           code 59";
        reference
          "RFC 5970 DHCPv6 Options for Network Boot, Section 3.1";
      }
    }
  }

  container dhcpv6-ldra {
    description
      "This is the top-level container for the DHCPv6 LDRA (lightweight
```


Copyright© 2022 Ciena® Corporation

```
     DHCPv6 relay agent) entity.";
  reference
    "RFC 6221, Lightweight DHCPv6 Relay Agent";

  leaf admin-state {
    type admin-state;
    description
      "The global administrative state of the LDRA.";
  }

  leaf interface-id-type {
    type enumeration {
      enum intid-string {
        description
          "The Interface-id option contains a string provided by the
           user to identify the interface on which the client message
           was received.";
      }
      enum slot-port {
        description
          "The Interface-id option contains the slot and port numbers
           of the interface on which the client message was
           received.";
      }
      enum slot-port-vlan {
        description
          "The Interface-id option contains the slot, port and VLAN
           numbers of the interface on which the client message was
           received.";
      }
    }
    description
      "The type of information sent by the relay agent in the
       Interface-id option to identify the interface on which the
       client message was received.";
    reference
      "RFC 6221, Section 5.3.2
       RFC 3315, Section 22.18";

  }

  leaf remote-id-option {
    type on-off-state;
    description
      "Determines whether or not the relay agent sends the DHCPv6
       Relay Agent Remote-ID option.";
    reference
      "RFC 4649";
  }

  leaf remote-id-type {
    type enumeration {
      enum device-hostname {
        description
          "The remote host is identified by the host name of this
           device.";
      }
      enum device-mac {
        description
          "The remote host is identified by the MAC address of this
```


Copyright© 2022 Ciena® Corporation

```
            device.";
        }
        enum rid-string {
          description
            "The remote host is identified by a string entered by the
             user.";
        }
      }
      description
        "The type of information sent by the relay agent in the
         Remote-ID option to identify the remote host";
      reference
        "RFC 4649";
    }

    leaf remote-id-enterprise-number {
      type uint32 {
        range "1..4294967295";
      }
      description
        "The vendor's registered Enterprise Number as registered with
         IANA.";
      reference
        "RFC 4649, Section 3";
    }

    list relay-agent {
      key "relay-type relay-id";

      leaf relay-type {
        type relay-type;
        description
          "The type of network element to which the relay agent is
           attached.";
      }

      leaf relay-id {
        type string;
        description
          "The name or VID of the VLAN, VS or MPLS-VC to which this
           relay agent is attached.";
      }

      leaf admin-state {
        type admin-state;
        description
          "The administrative state of this relay agent.";
      }

      description
        "List of relay agents and their configuration data.";
    }

    list relay-interface {
      key "relay-type relay-id interface-id sub-vlan";

      leaf relay-type {
        type relay-type;
        description
          "The type of network element to which the relay interface is
```


Copyright© 2022 Ciena® Corporation

```
      attached.";
  }

  leaf relay-id {
    type string;
    description
      "The name or VID of the VLAN, VS or MPLS-VC to which this
       relay agent is attached.";
  }

  leaf interface-id {
    type string;
    description
      "The name or port number of this relay agent interface.";
  }

  leaf sub-vlan {
    type uint32 {
      range "0..4094";
    }
    description
      "The VID of the sub-VLAN associated with this relay agent
       interface.  Zero means none.";
  }

  leaf trust-mode {
    type enumeration {
      enum client-trusted {
        description
          "Trust clients on this logical interface.";
      }
      enum server-trusted {
        description
          "Trust servers on this logical interface.";
      }
      enum dualrole-trusted {
        description
          "Trust both clients and servers on this logical
           interface.";
      }
      enum untrusted {
        description
          "Trust neither clients nor servers on this logical
           interface.";
      }
    }
    description
      "Grouping of relay interface attributes common to all
       interface types.";
  }

  leaf interface-id-string {
    type string {
      length "0..64";
    }
    description
      "Set the value of the Interface ID String for this logical
       interface.";
    reference
      "RFC 6221, Section 5.3.2
```


Copyright© 2022 Ciena® Corporation

```
            RFC 3315, Section 22.18";
        }

        leaf rid-string {
          type string {
            length "0..64";
          }
          description
            "Set the value of the Agent Remote ID String for this logical
             interface.";
          reference
            "RFC 4649, Section 3";
        }

        description
          "List of relay agents interfaces and their configuration data.";
      }
    }

    rpc dhcpv6-client-lease-renew {
      description
        "Renew the lease.";
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-dns-client.yang

```
module ciena-dns-client {
  namespace "http://www.ciena.com/ns/yang/ciena-dns-client";
  prefix dnsc;

  import ciena-types {
    prefix ciena;
  }

  import ietf-inet-types {
    prefix inet;
  }

  import ietf-system {
    prefix sys;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module defines Ciena's global configuration for DNS client
settings.

     Copyright (c) 2016 Ciena Corporation.  All rights
     reserved.

     All information contained herein is, and remains
     the property of Ciena Corporation. Dissemination of this
     information or reproduction of this material is strictly
     forbidden unless prior written permission is obtained from
     Ciena Corporation.";

  revision 2017-05-25 {
    description
      "Initial version";
    reference
      "RFC 6020: YANG - A Data Modeling Language for
       the Network Configuration Protocol (NETCONF).
       No specific reference; standard not available.";
  }

  /*
   * Typedefs
   */
  typedef dns-scope {
    type enumeration {
      enum user {
        description
          "The user-configured DNS parameter is currently operational.";
```


Copyright© 2022 Ciena® Corporation

```
      }
      enum dhcpv4 {
        description
          "The DHCPv4-configured DNS parameter is currently operational.";
      }
      enum dhcpv6 {
        description
          "The DHCPv6-configured DNS parameter is currently operational.";
      }
    }
      description
       "Whether the currently operational DNS parameter was configured by the
         user, DHCPv4, or DHCPv6.";
}

/*
 * Groupings
 */
grouping dhcp-server {
  description
    "Operational data pertaining to DNS servers configured through DHCP.";

  leaf domain-name {
    type inet:domain-name;
    description
      "Specifies the domain name configured through DHCP.";
  }

  list server {
    key address;
    description
    "State information pertaining to a DNS server configured through DHCP.";

    leaf address {
      type inet:ip-address;
      description
        "The IP address of the DNS server.";
    }

    leaf admin-status {
      type ciena:admin-state;
      description
        "The administrative state of the DNS server.";
    }

    leaf oper-status {
      type ciena:oper-state;
      description
        "The operational state of the DNS server.";
    }
  }
}

/*
 * Identities
 */

/*
 * Features
 */
```


Copyright© 2022 Ciena® Corporation

```
/*
 * Configuration data nodes
 */
augment /sys:system {

  container dns-client {
    description
      "Configuration parameters of the DNS client resolver.";

    leaf admin-status {
      type ciena:admin-state;
      description
        "The global administrative state of the DNS client resolver.";
    }

    leaf domain-name {
      type string {
        length "0..64";
      }
      description
       "Specifies the domain name configured by the user for this device.";
    }

    list server {
      key address;
      ordered-by user;
      description
        "Defines administrative information pertaining to a DNS server
         configured by the user.";

      leaf address {
        type inet:ip-address;
        description
          "The IP address of the configured DNS server.";
      }

      leaf admin-status {
        type ciena:admin-state;
        description
          "The administrative state of the configured DNS server.";
      }
    }
  } // end container dns-config
} // end /sys:system


/*
 * Operational state data nodes
 */
augment /sys:system-state {

  container dns-client {
    config false;
    description
      "Operational data for the DNS client resolver.";

    leaf domain-name-scope {
      type dns-scope;
      description
```


Copyright© 2022 Ciena® Corporation

```
          "Whether the currently operational domain name is user,
           DHCPv4, or DHCPv6.";
      }

      leaf server-scope {
        type dns-scope;
        description
          "Whether the currently operational DNS server is user,
           DHCPv4, or DHCPv6.";
      }

      container user-servers {
        description
         "Operational data pertaining to DNS servers configured by the user.";

        list server {
          key address;
          description
            "State information pertaining to a DNS server configured by the
user.";

          leaf address {
            type inet:ip-address;
            description
              "The IP address of the configured DNS server.";
          }

          leaf oper-status {
            type ciena:oper-state;
            description
              "The operational state of the configured DNS server.";
          }
        }
      }

      container dhcpv4-servers {
        description
          "Operational data pertaining to DNS servers configured through
DHCPv4.";
        uses dhcp-server;
      }

      container dhcpv6-servers {
        description
          "Operational data pertaining to DNS servers configured through
DHCPv6.";
        uses dhcp-server;
      }
    } // end container dns
  } // end /sys:system-state

  rpc resolve-ip {
    description
      "Resolve an IP address to a host name.";

    input {

      leaf address {
        type inet:ip-address;
        description
```


Copyright© 2022 Ciena® Corporation

```
            "The IP address to be resolved.";
        }
      }
      output {

        leaf host-name {
          type inet:domain-name;
          description
            "The host name resolved from the input IP address.";
        }
      }
    }

    rpc resolve-hostname {
      description
        "Resolve a host name to an IP address.";

      input {

        leaf host-name {
          type inet:domain-name;
          description
            "The host name to be resolved.";
        }
      }
      output {

        leaf address {
          type inet:ip-address;
          description
            "The IP address resolved from the input host name.";
        }
      }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-dot1x.yang

```
module ciena-dot1x {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-dot1x";
    prefix "ciena-dot1x";

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
                 Hanover, Maryland 21076
                 U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
        "This YANG module defines Ciena's global configuration of 802.1x";

    revision "2015-01-20" {
        description
            "Initial version";
        reference
            "Enter a reference here";
    }

    /*
     * typedefs
     */

    /*
     * Identities
     */

    /*
     * Features
     */

    /*
     * Configuration data nodes
     */
    container dot1x-config {
        leaf admin-enabled {
            type boolean;
            default false;
            description
                "Global administrative state of 802.1x functionality on the
                 box";
        }
    }

    /*
     * Operational state data nodes
     */
    container dot1x-state {
        config false;

        description
```


Copyright© 2022 Ciena® Corporation

```
            "Data nodes for the operational state/status of 802.1x.";
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-egress-qos.yang

```
module ciena-egress-qos
{
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-egress-qos";
    prefix "ciena-egress-qos";

    import ciena-mef-egress-qos
    {
        prefix "mef-egress-qos";
    }

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
                 Hanover, Maryland 21076
                 U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
        "This YANG module defines Ciena's management data definition for the
         management of egress-qos";

    revision "2016-03-29"
    {
        description
            "Initial version";
    }

    typedef ciena-drop-probability
    {
        type enumeration
        {
            enum 0pct;
            enum 1pct;
            enum 2pct;
            enum 3pct;
            enum 4pct;
            enum 5pct;
            enum 6pct;
            enum 7pct;
            enum 8pct;
            enum 9pct;
            enum 10pct;
            enum 25pct;
            enum 50pct;
            enum 75pct;
            enum 100pct;
        }
    }

    /*
     * Configuration data nodes
     */
    augment "/mef-egress-qos:egress-qos/mef-egress-qos:queue-group-profile"
```


Copyright© 2022 Ciena® Corporation

```
    {
        container ciena-queue-group-profile
        {
            leaf dynamic-max-queue-depth
            {
                type enumeration
                {
                   enum low
                   {
                       value 1;
                   }
                   enum medium
                   {
                       value 2;
                   }
                   enum medium-high
                   {
                       value 3;
                   }
                   enum high
                   {
                       value 4;
                   }
                }
            }
        }
    }

    augment "/mef-egress-qos:egress-qos/mef-egress-qos:scheduler-profile"
    {
        container ciena-scheduler-profile
        {
            leaf wdrr-scheduler-granularity
            {
                type uint32
                {
                    range "50..1600";
                }
            }
        }
    }

    augment "/mef-egress-qos:egress-qos/mef-egress-qos:congestion-avoidance-
profile"
    {
        container ciena-saos-sds-congestion-avoidance-profile
        {
            leaf green-lower-threshold
            {
                type uint16
                {
                    range "1..100";
                }
                description "Maps to green-lower-threshold on non-3960
                    platforms and to tcp-green-threshold on 3960.";
            }

            leaf green-upper-threshold
            {
                type uint16
```


Copyright© 2022 Ciena® Corporation

```
        {
            range "1..100";
        }
        description "Maps to green-upper-threshold on non-3960
            platforms and to tcp-green-upper-threshold on 3960.";
    }

    leaf green-max-drop-probability
    {
        type ciena-drop-probability;
        description "Maps to green-max-drop-probability on non-3960
          platforms and to tcp-green-max-drop-probability on 3960.";
    }

    leaf yellow-lower-threshold
    {
        type uint16
        {
            range "1..100";
        }
        description "Maps to yellow-lower-threshold on non-3960
            platforms and to tcp-yellow-threshold on 3960.";
    }
    leaf yellow-upper-threshold
    {
        type uint16
        {
            range "1..100";
        }
        description "Maps to yellow-upper-threshold on non-3960
            platforms and to tcp-yellow-upper-threshold on 3960.";
    }

    leaf yellow-max-drop-probability
    {
        type ciena-drop-probability;
        description "Maps to yellow-max-drop-probability on non-3960
          platforms and to tcp-yellow-max-drop-probability on 3960.";
    }

    leaf non-tcp-threshold
    {
        type uint16
        {
            range "1..100";
        }
        description "For 3960 platforms only.";
    }

    leaf non-tcp-upper-threshold
    {
        type uint16
        {
            range "1..100";
        }
        description "For 3960 platforms only.";
    }

    leaf non-tcp-max-drop-probability
    {
```


Copyright© 2022 Ciena® Corporation

```
                type ciena-drop-probability;
                description "For 3960 platforms only.";
            }

            leaf yellow-admit-limit
            {
                type uint16
                {
                    range "1..100";
                }
                description "For 3960 platforms only.";
            }
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-eoam.yang

```
module ciena-eoam {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-eoam";
    prefix "ciena-eoam";

    organization
      "Ciena Corporation";

    contact
      "Web URL: http://www.ciena.com/
       E-mail:  yang@ciena.com
       Postal:  7035 Ridge Road
                Hanover, Maryland 21076
                U.S.A.
       Phone:   +1 800-921-1144
       Fax:     +1 410-694-5750";

    description
      "This YANG module contains the schema definition for Ciena's
       global configuration of Ethernet Operations,
       Administration and Management (EOAM).";

    revision "2015-02-25" {
      description
        "Initial Version.";
      reference
        "Ciena EOAM global configuration";
    }

    container global-config-status {
        leaf admin-state {
            type enumeration {
                enum enabled {
                    description
                        "Enable EOAM globally on this device.";
                }
                enum disabled {
                    description
                       "Disable EOAM globally on this device.";
                }
            }
            default disabled;
            description
                "Enable or disable EOAM globally on this device";
        }
        leaf oper-state {
            type enumeration {
                enum enabled {
                    description
                        "EOAM is globally enabled on this device.";
                }
                enum disabled {
                   description
                        "EOAM is globally enabled on this device.";
                }
            }
            config false;
            description
                "EOAM global operational status.";
```


Copyright© 2022 Ciena® Corporation

```
        }
        description
            "EOAM global operational status on this device";
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-ethernet-port-deviation.yang

```
module ciena-ethernet-port-deviation {
    namespace "http://www.ciena.com/ns/yang/ciena-ethernet-port-dev";
    prefix "ciena-eth-dev";

    import ietf-interfaces {
        prefix "if";
    }
    import ciena-ethernet-port {prefix ciena-eth;}

    revision "2015-01-20" {
        description
            "Initial version";
    }


   deviation /if:interfaces/if:interface/ciena-eth:link-flap-detect {
       deviate not-supported;
   }

   deviation /if:interfaces/if:interface/ciena-eth:link-flap-count {
       deviate not-supported;
   }


   deviation /if:interfaces/if:interface/ciena-eth:link-flap-detect-time {
      deviate not-supported;
   }

   deviation /if:interfaces/if:interface/ciena-eth:link-flap-hold-time {
       deviate not-supported;
   }

}
```


Copyright© 2022 Ciena® Corporation

ciena-ethernet-port.yang

```
module ciena-ethernet-port {
    namespace "http://www.ciena.com/ns/yang/ciena-ethernet-port";
    prefix "ciena-eth";

    import ietf-interfaces {
        prefix "if";
    }

    import iana-if-type {
        prefix "ianaift";
    }

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
                 Hanover, Maryland 21076
                 U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
        "This YANG module defines Ciena's specific model of IEEE 802.3
         Ethernet Port.";

    revision "2015-01-20" {
        description
            "Initial version";
        reference
            "Enter a reference here";
    }

    augment "/if:interfaces/if:interface" {
        when "if:type = 'ianaift:ethernetCsmacd'";

        leaf mode {
            type enumeration {
                enum auto {
                     description
                         "This is the default port mode whereby the mode of
                         the port is determined and assigned by the system.";
                }
                enum rj45 {
                    description
                        "Connector is standardized as the IEC 60603-7
                        8P8C modular connector with eight conductors.";
                }
                enum sfp {
                    description
                         "Small form-factor pluggable (SFP) connector.";
                }

             }
             default auto;
        }
```


Copyright© 2022 Ciena® Corporation

```
        leaf link-flap-detect {
            type boolean;
            default "false";
            description
                "Enable the detection of link flap. Only when it is enabled
               do the following leafs apply. If disabled, the leaf and their
                values stay, but have have no impact on detection of flap.";
        }

        leaf link-flap-count {
            type uint8;
            default 5;
            description
                "The number of times the link can flap before the device
                 declares the link to be down. This is measured over the
                 time period defined by the next leaf link-flap-detect-time.
              Once down, the device will wait for link-flap-hold-time before
                 trying to bring the link back up.";
        }

        leaf link-flap-detect-time {
            type uint32 {
                range "1..600";
            }
            default 10;
            description
                "The amount of time in seconds the device will count the
                 number of times the link has flapped, before declaring the
                 link to be down.";
        }

        leaf link-flap-hold-time {
            type uint32 {
                range "1..600";
            }
            default 300;
            description
                "The amount of time in seconds the device will hold the
                 link in down state before trying to bring it up again.";
        }
    }


    // operational state parameters for Ethernet interfaces
    augment "/if:interfaces-state/if:interface" {
        when "if:type = 'ianaift:ethernetCsmacd'";

        leaf mode {
            type enumeration {
                enum rj45 {
                    description
                        "Connector is standardized as the IEC 60603-7
                        8P8C modular connector with eight conductors.";
                }
                enum sfp {
                    description
                        "Small form-factor pluggable (SFP) connector.";
                }
            }
```


Copyright© 2022 Ciena® Corporation

```
            description
                "The (manual) configured mode on the port.";
        }

        leaf operational-state {
            type enumeration {
                enum linkflap;
            }
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-event.yang

```
module ciena-event {
   namespace "http://www.ciena.com/ns/yang/ciena-event";
   prefix "ciena-event";

   import ciena-types {
      prefix ciena;
   }

   import ietf-syslog-types {
      prefix syslogtypes;
   }

   organization
      "Ciena Corporation";

   contact
      "Web URL: http://www.ciena.com/
      E-mail:  yang@ciena.com
      Postal:  7035 Ridge Road
      Hanover, Maryland 21076
      U.S.A.
      Phone:   +1 800-921-1144
      Fax:     +1 410-694-5750";

   description
      "This YANG module defines Ciena's global configuration for event
settings.

      Copyright (c) 2017 Ciena Corporation.  All rights
      reserved.

      All information contained herein is, and remains
      the property of Ciena Corporation. Dissemination of this
      information or reproduction of this material is strictly
      forbidden unless prior written permission is obtained from
      Ciena Corporation.";

   revision "2017-04-26" {
      description
         "Initial version.";
   }

   feature config-events {
      description
         "If there's support for enabling/disabling events.";
   }

   feature syslog-event {
      description
         "If there's support for displaying syslog events.";
   }

   identity event-severity {
      description
         "Base event severity identity.";
   }

   identity event-cleared {
```

```
      base event-severity;
      description
         "Cleared event";
   }

   identity event-indeterminate {
      base event-severity;
      description
         "Indeterminate event";
   }

   identity event-critical {
      base event-severity;
      description
         "Critical event";
   }

   identity event-major {
      base event-severity;
      description
         "Major event";
   }

   identity event-minor {
      base event-severity;
      description
         "Minor event";
   }

   identity event-warning {
      base event-severity;
      description
         "Warning event";
   }

   identity event-config {
      base event-severity;
      description
         "Config event";
   }

   identity event-info {
      base event-severity;
      description
         "Info event";
   }

   identity event-debug {
      base event-severity;
      description
         "Debug event";
   }

   container events {
      description
         "Configuration of SAOS events.";

      if-feature config-events;

      list events {
```


Copyright© 2022 Ciena® Corporation

```
        description
           "SAOS events.";

        key event-id;

        leaf event-id {
           type uint32;
           description
              "Specifies the event ID to the event.";
        }

        leaf user-severity {
           type identityref {
              base event-severity;
           }
           description
              "The user specified severity for the event.";
        }

        leaf event-state {
           type ciena:admin-state;
           description
              "The administrative state of the event.";
        }
     }
  }

  container event-state {
     config false;
     description
        "The operational state for SAOS events.";

     list event {
        description
           "SAOS events.";

        key event-id;

        leaf event-id {
           type uint32;
           description
              "Specifies the event ID to the event.";
        }

        leaf default-severity {
           type identityref {
              base event-severity;
           }
           description
              "The default severity for the event.";
        }

        leaf manager-name {
           type string;
           description
              "The manager that owns the event.";
        }

        leaf syslog-severity {
           if-feature syslog-event;
```


Copyright© 2022 Ciena® Corporation

```
            type syslogtypes:severity;
            description
               "The syslog severity for this event.";
         }

         leaf event-description {
            type string;
            description
               "The event description.";
         }
      }

      list event-severity-map {
         config false;
         description
            "A mapping between SAOS event severities and syslog severities.";
         key event-severity;

         leaf event-severity {
            type identityref {
               base event-severity;
            }
            description
               "SAOS event severity.";
         }

         leaf syslog-severity {
            type syslogtypes:severity;
            description
               "Syslog event severity.";
         }
      }
   }
}
```


Copyright© 2022 Ciena® Corporation

ciena-ext-lag.yang

```
module ciena-ext-lag {

  namespace "http://www.ciena.com/ns/yang/ciena-ext-lag";
  prefix "ciena-ext-lag";

  import ietf-yang-types {
    prefix "yang";
  }
  import ciena-ieee-lag {
    prefix "ciena-ieee-lag";
  }
  import ietf-interfaces {
    prefix "if";
  }
  import mef-cfm {
    prefix "cfm";
  }

  organization
   "Ciena, Inc";

  contact
   "   Mib Meister
    115 North Sullivan Road
    Spokane Valley, WA 99037
    USA
    Phone:  +1 509 242 9000
    Email:  support@ciena.com";

  description
   "This yang module is an extension to the IEEE8023-LAG yang model.
    It defines the managed objects for the creation and
    deletion of Aggregators.";

  revision 2018-08-10 {
    description
     "Added support to set/unset CFM service on a LAG member port.";
  }

  revision 2016-03-06 {
    description
     "Initial revision";
  }


//
// Type Definitions related to ciena proprietary extansion of LACP
//

  // AGG REDUNDANCY STATE

  identity agg-redundancy-state {
    description
     "Indicates the redundancy-state of the aggregation";
  }

  identity agg-redundancy-state-down        {
        description
```


Copyright© 2022 Ciena® Corporation

```
        "Indicates that the MC-LAG node is oper-down";
}

identity agg-redundancy-state-unavailable {
  base agg-redundancy-state;
  description
   "Indicates that the MC-LAG node is currently
    unavailable";
}

identity agg-redundancy-state-active      {
  base agg-redundancy-state;
  description
   "Indicates that the MC-LAG node is currently in active
    state";
}

identity agg-redundancy-state-standby     {
  base agg-redundancy-state;
  description
   "Indicates that the MC-LAG mode is currently in standby
    state. This means the peer RG node is in active state";
}

identity agg-redundancy-state-standalone  {
  base agg-redundancy-state;
  description
   "Indicates that the MC-LAG node is currently in
    standalone state. Connection with the peer RG node is
    lost";
}

identity agg-redundancy-state-none        {
  base agg-redundancy-state;
  description
   "Indicates that the MC-LAG node is currently in undefined
    state";
}


// AGG MODE

identity agg-mode-type {
  description
   "Indicates the aggregation mode";
}

identity agg-mode-lacp {
  base agg-mode-type;
  description
   "Indicates the aggregation mode for the agg is lacp";
}

identity agg-mode-manual {
  base agg-mode-type;
  description
   "Indicates the aggregation mode for the agg is manual";
}
```


Copyright© 2022 Ciena® Corporation

```
// MIN LINK AGGREGATION STATE

identity min-link-aggregation-state {
  description
   "Indicates the state of min-link for the aggregation";
}

identity min-link-on {
  base min-link-aggregation-state;
  description
   "Indicates on-state for min-link-aggregation";
}
identity min-link-off {
  base min-link-aggregation-state;
  description
   "Indicates off-state for min-link-aggregation";
}


typedef min-link-threshold-value-range {
  type uint8 {
    range "1..8";
  }
  description
   "Indicates the range of legal values for min-link-threshold";
}


// MULTI CHASSIS STATE

identity multi-chassis-state {
  description
   "Specifies if the LAG is an MC-LAG";
}

identity multi-chassis-enable {
  base multi-chassis-state;
  description
   "Indicates user intends multi-chassis operation";
}

identity multi-chassis-disable {
  base multi-chassis-state;
  description
   "Indicates user intends regular LACP operation";
}


// MULTI CHASSIS CONNECT STATE

identity multi-chassis-connect-state {
  description
   "Specifies the Connect State of the MC-LAG node.";
}

identity multi-chassis-init {
  base multi-chassis-connect-state;
  description
   "Indicates that the MC-LAG node is in init state";
}
```


Copyright© 2022 Ciena® Corporation

```
identity multi-chassis-disconnected {
  base multi-chassis-connect-state;
  description
   "Indicates that the MC-LAG node is in disconnected
    state";
}

identity multi-chassis-connected {
  base multi-chassis-connect-state;
  description
   "Indicates that the MC-LAG node is in connected state";
}

identity multi-chassis-mismatch {
  base multi-chassis-connect-state;
  description
   "Indicates that the MC-LAG node is in mismatch state";
}

identity multi-chassis-none {
  base multi-chassis-connect-state;
  description
   "Indicates that the MC-LAG node is in undefined state";
}


// MULTI CHASSIS AGG ROLE

identity multi-chassis-agg-role {
  description
   "Specifies if the MC-LAG node is primary or backup";
}

identity multi-chassis-agg-role-primary {
  description
   "In case of multi-chassis operation, this is the primary
    link to the dual-homing device";
}

identity multi-chassis-agg-role-backup {
  base multi-chassis-agg-role;
  description
   "In case of multi-chassis operation, this is the backup
    link to the dual-homing device";
}

identity multi-chassis-agg-role-none {
  base multi-chassis-agg-role;
  description
   "In case of multi-chassis operation, this is the dual-
    homing device";
}


// LAG HASH MODE

identity lag-hash-mode-type {
  description
   "Specifies the hash mode used by the aggregation
```


Copyright© 2022 Ciena® Corporation

```
    for known unicast frames." ;
}

identity mac-based-hash-mode {
  base lag-hash-mode-type;
  description
   "Indicates that the hash mode used for known unicast
    frames is mac-based";
}

identity ip-based-hash-mode {
  base lag-hash-mode-type;
  description
   "Indicates that the hash mode used for known unicast
    frames is ip-based";
}

identity enhanced-hash-mode {
  base lag-hash-mode-type;
  description
   "Indicates that the hash mode used for known unicast
    frames is enhanced";
}


// LAG PROTECTION MODE

identity lag-protection-mode-type {
  description
   "Specifies the LACP protection-mode in which the
    given aggregation is operating." ;
}

identity lag-protection-mode-proprietary {
  base lag-protection-mode-type;
  description
   "Indicates that the LACP protection-mode is proprietary";
}

identity lag-protection-mode-standard {
  base lag-protection-mode-type;
  description
   "Indicates that the LACP protection-mode is standard";
}


// FLOOD HASH MODE

identity lag-flood-hash-mode-type {
  description
   "Specifies the aggregation hash mode used by
    unknown unicast, multicast and broadcast frames for all
    aggregation ports in the system." ;
}

identity simplified-flood-hash-mode {
  base lag-flood-hash-mode-type;
  description
   "Indicates simplified flood hash mode for unknown unicast
    multicast, and broadcast frames";
```


Copyright© 2022 Ciena® Corporation

```
  }

  identity enhanced-flood-hash-mode {
    base lag-flood-hash-mode-type;
    description
     "Indicates enhanced flood hash mode for unknown unicast,
      multicast, and broadcast frames";
  }


//
// Groupings related to ciena proprietary extension of LACP
//

  grouping mc-lag-parameters-group {

    description
     "User-configurable ciena-proprietary MC-LAG parameters";

    leaf multi-chassis {
      type identityref {
        base multi-chassis-state;
      }
      description
       "This object specifies if the LAG is an MC-LAG";
    }

    leaf icl {
      type string;
      description
       "This object specifies the inter-chassis link name that the
         multi-chassis LAG is using.";
    }

  }

  grouping mc-lag-operational-group {

    description
     "Operational states of various MC-LAG components";

    leaf rg-node-index {
      type uint32;
      description
       "This object specifies the MC-LAG RG node ID.";
    }

    leaf rg-dynamic-priority {
      type uint32;
      description
       "This object specifies the MC-LAG RG node dynamically
        assigned priority.";
    }

    leaf rg-oper-key {
      type uint32;
      description
       "This object specifies the MC-LAG RG node operational key.";
    }
```


Copyright© 2022 Ciena® Corporation

```
leaf connect-state {
  type identityref {
    base multi-chassis-connect-state;
  }
  description
   "This object specifies the Connect State of the MC-LAG node.";
}

leaf redundancy-state {
  type identityref {
    base agg-redundancy-state;
  }
  description
   "This object specifies the oper state of the MC-LAG node";
}

leaf peer-system-mac {
  type yang:mac-address;
  description
   "This object specifies the MC-LAG peer node system MAC
    address.";
}

leaf peer-rg-node-id {
  type uint32;
  description
   "This object specifies the MC-LAG peer RG node ID.";
}

leaf peer-rg-dynamic-priority {
  type uint32;
  description
   "This object specifies the MC-LAG peer RG node dynamically
    assigned priority.";
}

leaf peer-rg-oper-key {
  type uint32;
  description
   "This object specifies the MC-LAG peer RG node operational
    key.";
}

leaf agg-oper-mode {
  type identityref {
    base agg-mode-type;
  }
  description
   "This object specifies the operation mode in which a given
    port is operating." ;
}

leaf agg-role {
  type identityref {
    base multi-chassis-agg-role;
  }
  description
   "This object specifies if the MC-LAG node is primary or backup";
}
```


Copyright© 2022 Ciena® Corporation

```
}


grouping mc-lag-statistics-group {

  description
   "Packet and timer related statistics for MC-LAG";

  leaf disconnect-rx {
    type yang:counter32;
    description
     "This object specifies the number of Disconnect messages
      received by the MC-LAG node.";
  }

  leaf disconnect-tx {
    type yang:counter32;
    description
     "This object specifies the number of Disconnect messages
      transmitted by the MC-LAG node.";
  }

  leaf config-mismatch-rx {
    type yang:counter32;
    description
     "This object specifies the number of mismatch config messages
      received by the MC-LAG node.";
  }

  leaf key-mismatch-count {
    type yang:counter32;
    description
     "This object specifies the number of messages received
      with mismatching key by the MC-LAG node.";
  }

  leaf out-of-sequence-rx {
    type yang:counter32;
    description
     "This object specifies the number of out of sequence messages
      received by the MC-LAG node.";
  }

  leaf peer-unreachable-count {
    type yang:counter32;
    description
     "This object specifies the number of times peer was
      unreachable.";
  }

  leaf unknown-rx {
    type yang:counter32;
    description
     "This object specifies the number unknown messages received
      by the MC-LAG node.";
  }

  leaf total-rx {
    type yang:counter32;
    description
```


Copyright© 2022 Ciena® Corporation

```
      "This object specifies the total number of messages received
       by the MC-LAG node.";
  }

  leaf total-tx {
    type yang:counter32;
    description
     "This object specifies the total number of messages
      transmitted by MC-LAG node.";
  }

  leaf total-down-time {
    type uint32;
    description
     "This object specifies total time for which the MC-LAG node
      has been operationally down.";
  }

  leaf up-time {
    type uint32;
    description
     "This object specifies time for which the MC-LAG node has been
      operationally up.";
  }

  leaf time-in-protect-state {
    type uint32;
    description
     "This object specifies time for which the MC-LAG node has been
      in protection state.";
  }

  leaf last-time-protected {
    type uint32;
    description
     "This object specifies time of day of the last MC-LAG
      protection switching.";
  }

  leaf number-of-switchovers {
    type yang:counter32;
    description
     "This object specifies the number of MC-LAG protection
      switching.";
  }

}

grouping min-link-agg-parameters-group {

  description
   "User configurable parameters related to ciena-proprietary
    min-link-aggregation";

  leaf min-link-aggregation {
    type identityref {
      base min-link-aggregation-state;
    }
    description
     "This object allows the user to enable or disable min-link-
```


Copyright© 2022 Ciena® Corporation

```
        aggregation on an agg";
    }

    leaf min-link-threshold {
      type min-link-threshold-value-range;
      description
        "This object allows the user to specify the threshold for
         min-link-aggregation";
    }
  }

  grouping min-link-agg-operational-group {

    description
     "Operational data related to ciena-proprietary
      min-link-aggregation";

    leaf min-link-aggregation {
      type identityref {
        base min-link-aggregation-state;
      }
      description
       "This object allows the user to enable or disable min-link-
        aggregation on an agg";
    }

    leaf min-link-threshold {
      type min-link-threshold-value-range;
      description
        "This object allows the user to specify the threshold for
         min-link-aggregation";
    }
  }

  augment "/ciena-ieee-lag:agg-global" {

    description
     "Augmenting configurable parameters global to all Aggregators in the
      system to the standard yang";

    leaf flood-hash-mode {
      type identityref {
        base lag-flood-hash-mode-type;
      }
      description
       "This object specifies the aggregation hash mode used by
        unknown unicast, multicast and broadcast frames for all
        aggregation ports in the system." ;
    }

    leaf marker-timeout {
      type uint32 {
        range "0..1000";
      }

      description
       "This object indicates the marker timeout in the system";
    }
  }
```


Copyright© 2022 Ciena® Corporation

```
augment "/ciena-ieee-lag:agg-global-state" {

  description
   "Augmenting ciena-proprietary global info to the standard
    yang";

  leaf max-lags {
    type uint32;
    description
     "The maximum number of aggregators that this device
      supports.";
  }

  leaf num-lags {
    type uint32;
    description
     "The current number of LAGs that are
      configured in this device.";
  }
}

augment "/if:interfaces/if:interface/ciena-ieee-lag:agg" {

  description
   "Adding ciena-proprietary configurable parameters to the
    standard yang.";

  leaf system-mac {
    type yang:mac-address;
    description
     "A 6-octet read-only value specifying the
      MC-LAG redundancy group MAC.";
  }

  leaf lag-protection-revert-state {
    type enumeration {
      enum on  {
        description
         "Indicates that if the primary port goes from
          operationally disabled to enabled, it shall become
          active, and one of the secondary ports shall become
          backup when the protection-revert-timer expires";
      }
      enum off {
        description
         "Indicates that the secondary port will remain active
          even if the primary port goes from operationally
          disabled to enabled";
      }
    }
    description
     "This object specifies that if the primary port goes from
      operationally disabled to operationally enabled it shall
      become active, and one of the secondary ports shall become
      backup if this object is set to 'on' and the
      protection-revert-timer expires." ;
  }

  leaf lag-protection-revert-timer {
    type uint32 {
```


Copyright© 2022 Ciena® Corporation

```
        range "0..720000";
      }
      description
       "This object specifies the revert protection timer value.
        The default for  multi-chassis LAG is 300000 ms and for
        regular LAG is 5000 ms";
    }

    leaf hash-mode {
      type identityref {
        base lag-hash-mode-type;
      }
      description
       "This object specifies the hash mode used by the aggregation
        for known unicast frames." ;
    }

    leaf lag-protection-mode {
      type identityref {
        base lag-protection-mode-type;
      }
      description
       "This object specifies the LACP protection-mode in which the
        given aggregation is operating." ;
    }

    leaf agg-admin-mode {
      type identityref {
        base agg-mode-type;
      }
      description
       "This object specifies the administrative mode in which a
        given port is operating." ;
    }

    uses mc-lag-parameters-group;
    uses min-link-agg-parameters-group;
  }

  augment "/if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:member-ports" {
    leaf protection-port {
      type boolean;
      default "false";
      description
       "This object specifies whether the member port is primary or protection
port.";
    }
  }

  augment "/if:interfaces-state/if:interface/ciena-ieee-lag:agg-state" {

    description
     "Adding ciena-proprietary operational data to the standard
      yang";
    uses mc-lag-operational-group;
    uses mc-lag-statistics-group;
    uses min-link-agg-operational-group;
  }
```


Copyright© 2022 Ciena® Corporation

```
  augment "/if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:member-ports" {
    list cfm-service {
      key "maintenance-domain maintenance-association";
      description
        "The CFM service to be associated with a LAG member port.";

      uses cfm:maintenance-association-reference;
    }
  }

  // Notifications
  notification mclag-state-change {

    description
     "This notification is sent when MC-LAG oper state changes.
      Var binding index represents the MC-LAG PGID.";

    list state {
      key "index";
      description
       "Change in redundancy-state per Aggregation in the system";

      leaf index {
        type uint32 {
          range "1..2147483647";
        }
        description
         "The unique identifier allocated to this Aggregator by
          the local System.  This attribute identifies an
          Aggregator instance among the subordinate managed
          objects of the containing object.
          This value is read-only. This object refer to the
          dot3ad-agg-index in the IEEE-8023-LAG-MIB.";
        reference
         "IEEE 802.3 Subclause 30.7.1.1.1";
      }

      leaf redundancy-state {
        type identityref {
          base agg-redundancy-state;
        }
        description
         "This object specifies the oper state of the MC-LAG node";
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-fd-sds.yang

```
module ciena-fd-sds {
   namespace "http://www.ciena.com/ns/yang/ciena-fd-sds";
   prefix "ciena-fd-sds";

   import ciena-mef-fd {
      prefix "mef-fd";
   }

   import ieee-dot1q-types {
      prefix dot1q-types;
   }

   organization
      "Ciena Corporation";

   contact
      "Web URL: http://www.ciena.com/
      E-mail:  yang@ciena.com
      Postal:  7035 Ridge Road
      Hanover, Maryland 21076
      U.S.A.
      Phone:   +1 800-921-1144
      Fax:     +1 410-694-5750";

   description
      "This YANG module defines Ciena's configuration of
      the Forwarding Domain (Bridge Domain, Virtual Switch)
      specific to SAOS 6.x

      Copyright (c) 2018 Ciena Corporation.  All rights
      reserved.

      All information contained herein is, and remains
      the property of Ciena Corporation. Dissemination of this
      information or reproduction of this material is strictly
      forbidden unless prior written permission is obtained from
      Ciena Corporation.";

   revision "2018-04-12" {
      description
         "Initial version";
      reference
         "RFC 6020: YANG - A Data Modeling Language for
         the Network Configuration Protocol (NETCONF).
         No specific reference; standard not available.";
   }

   augment "/mef-fd:fds/mef-fd:fd" {

      container ciena-fd-sds {

         leaf decap-cos-policy {
            type enumeration {
               enum vc-inherit {
                  description
                     "Inherit decap COS policy from VC";
               }
               enum tunnel-inherit {
```

```
                  description
                     "Inherit decap COS policy from tunnel";
               }
               enum leave {
                  description
                     "Inherit decap COS policy leave";
               }
               enum fixed {
                  description
                     "Use fixed decap COS value";
               }
            }
            description
               "Decapsulation COS policy";
         }

         leaf decap-fixed-dot1dpri {
            type dot1q-types:priority-type;
            description
               "Fixed decapsulation COS";
         }

         leaf encap-fixed-dot1dpri {
            type dot1q-types:priority-type;
            description
               "Fixed encapsulation COS";
         }
         description
            "Contains SAOS 6.x specific FD leafs";
      }
      description
         "This module augmented in ciena-mef-fd module";
   }
}
```


Copyright© 2022 Ciena® Corporation

ciena-fd.yang

```
module ciena-fd {
    namespace "http://www.ciena.com/ns/yang/ciena-fd";
    prefix "ciena-fd";

    import ciena-mef-fd {
        prefix "mef-fd";
    }

    import ciena-reserved-vlan-sds {
        prefix "ciena-reserved-vlan-sds";
    }

    import ciena-types {
        prefix "ciena-types";
    }

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
        Hanover, Maryland 21076
        U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
        "This YANG module defines Ciena's configuration of
        the Forwarding Domain (Bridge Domain, Virtual Switch,
                VPLS representation for the NETCONF protocol).";

    revision "2019-06-26" {
        description
            "Added 'when statement' to conditionally create a leaf 'egress-
tpid' when mode is 'vlan'.";
    }

    revision "2017-12-05" {
        description
            "Added leaf egress-tpid to support egress-tpid
             configuration on a VLAN";
        reference
            "RFC 6020: YANG - A Data Modeling Language for
             the Network Configuration Protocol (NETCONF).
             No specific reference; standard not available.";
    }

    revision "2017-06-20" {
        description
            "Added PFG administrative state and operational state
             for forwarding domain.";
        reference
            "RFC 6020: YANG - A Data Modeling Language for
             the Network Configuration Protocol (NETCONF).
             No specific reference; standard not available.";
    }
```


Copyright© 2022 Ciena® Corporation

```
    revision "2017-02-01" {
        description
            "Added reserved-vlan for forwarding domain of type vpls.";
    }

    revision "2016-09-01" {
        description
            "Initial version";
    }

    augment "/mef-fd:fds/mef-fd:fd" {

        container ciena-fd {

            // 6.x support for Enabling L2PT MAC Translation of Tagged PVST
frames on Forwarding Domains
            leaf tagged-pvst-l2pt {
                type enumeration {
                    enum enabled {
                        description "Enable L2PT MAC Translation on Tagged
PVST frames entering this forwarding domain from UNI";
                    }
                    enum disabled {
                        description "Disable L2PT MAC Translation on Tagged
PVST frames entering this forwarding domain from UNI";
                    }
                }
                default "disabled";

                description
                    "It determines that tagged PVST frames on UNI's of this
forwarding domain shall undergo L2PT MAC
                    Translation in case operation is enabled.";
            }

           // 6.x support to display reserved-vlan on Forwarding Domains of
type vpls
            leaf reserved-vlan {
                type ciena-reserved-vlan-sds:reserved-vlan-ref;

                description
                    "This is reference of VID, which is reserved for
                    forwarding domain of type vpls.";
            }

            // 6.x support for private forwarding groups admin state
            leaf pfg-admin-state {
                type ciena-types:admin-state;
                description
                   "This represents private forwarding group administrative
                    state of corresponding forwarding domain.";
                default disabled;
            }

            // 6.x support for vlan egress-tpid
            leaf egress-tpid {
                when "../../mef-fd:mode = 'vlan'";
                type enumeration {
                    enum 0x8100 {
```


Copyright© 2022 Ciena® Corporation

```
                        description
                            "VLAN tagged.";
                    }
                    enum 0x88a8 {
                        description
                            "Provider bridging.";
                    }
                    enum 0x9100 {
                        description
                            "Proprietary.";
                    }
                }
                default 0x8100;
                description
                    "TPID to use for VLAN based Egress TPID stamping.";
            }
        }
    }

    augment "/mef-fd:fds-state/mef-fd:fd" {

        container ciena-fd-state {
            // This container augments Ciena-Fd operational data
            // which may be used in future

            config false;

            leaf reserved-vlan-auto {
                type ciena-reserved-vlan-sds:reserved-vlan-ref;

                description
                    "This is reference of VID, which is auto allocated
                    to forwarding domain of type vpls.";
            }

            leaf pfg-oper-state {
                type ciena-types:oper-state;
                description
                    "This represents private forwarding group operational
                     state of corresponding forwarding domain.";
                default disabled;
            }

            description
                "This represents operational data for Ciena-Fd";
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-file-transfer-types.yang

```
module ciena-file-transfer-types {
  namespace
    "http://www.ciena.com/ns/yang/ciena-file-transfer-types";

  prefix "ciena-ftt";

  import ietf-inet-types {
    prefix inet;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module defines Ciena's file transfer types.

     Copyright (c) 2017 Ciena Corporation.  All rights
     reserved.

     All information contained herein is, and remains
     the property of Ciena Corporation. Dissemination of this
     information or reproduction of this material is strictly
     forbidden unless prior written permission is obtained from
     Ciena Corporation.";

  revision 2017-10-31 {
    description
      "Added server-port to the file-transfer-parameters grouping.";
    reference
      "";
  }

  revision 2017-03-08 {
    description
      "Initial version";
    reference
      "009-3273-007 (39XX 51XX SAOS 6.16 Administration & Security)
       Standard Revision A

       8700 Packetwave Platform Administration and Security, SAOS
       8.4, 380-1874-301, 8700, Rev A";
  }

  /*
   * typedefs
   */
  typedef file-transfer-status {
    type enumeration {
      enum idle {
```


Copyright© 2022 Ciena® Corporation

```
        description
          "No files are being downloaded at this time.";
      }
      enum busy {
        description
          "A file is currently being downloaded.";
      }
    }
    description
      "File transfer status values.";
  }

  /*
   * identities
   */

  identity file-transfer-server {
    description
      "Base identity for file transfer server types.";
  }

  identity tftp-server {
    base file-transfer-server;
    description
      "TFTP file transfer server type.";
  }

  identity ftp-server {
    base file-transfer-server;
    description
      "FTP file transfer server type.";
  }

  identity sftp-server {
    base file-transfer-server;
    description
      "SFTP file transfer server type.";
  }

  identity http-server {
    base file-transfer-server;
    description
      "HTTP file transfer server type.";
  }

  /*
   * Groupings
   */
  grouping file-transfer-parameters {
    choice file-transfer-syntax {
      case url {
        leaf url {
          type inet:uri;
          description
            "Use a URL to specify the transfer protocol,
             address/hostname, path, username, and password";
        }
      }

      case component {
```


Copyright© 2022 Ciena® Corporation

```
        leaf filename {
          type string;
          description
            "Path and filename.";
        }

        leaf server-type {
          type identityref {
            base file-transfer-server;
          }
          description
            "The server type for file transfer.";
        }

        leaf address {
          type inet:host;
          description
            "The server IP address or hostname. The
             server-type leaf must be specified
             when using this leaf.";
        }

        leaf server-port {
          type inet:port-number;
          description
            "The server port number.";
        }

        leaf login-id {
          type string;
          description
            "Login ID for the specified server.";
        }

        leaf password {
          type string;
          description
            "Password associated with the login-id.";
        }
      }
      description
        "Choice between using a URL to specify the file transfer
         information or individual components.";
    }
    description
      "File transfer options.";
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-flood-containment-profile.yang

```
module ciena-flood-containment-profile {
   namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-flood-containment-
profile";
    prefix "ciena-flood-containment-profile";

    import ciena-mef-flood-containment-profile {
        prefix "mef-fc";
    }

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
                 Hanover, Maryland 21076
                 U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
       "This YANG module defines Ciena's global settings for configuring flood
         containment profiles.";

    revision "2019-06-28" {
        description
            "Added leaf ignore-multicast-cfm-msgs to ignore multicast CFM
packets
             filtering by flood-containment rules.";
    }

    revision "2016-05-26" {
        description
            "Initial revision.";
    }

    /*
     * Configuration data nodes
     */
    augment "/mef-fc:flood-containment-profiles" {

        container ciena-flood-containment-profiles {

            leaf resource-mode {
                type boolean;
            }

            leaf ignore-raps-msgs {
                type boolean;
            }

            leaf ignore-multicast-cfm-msgs {
                type boolean;
                    description
                        "Ignore multicast cfm messages when this leaf is set
to true.
                         By default it is set to false.";
```

```
                }
            }
        }
}
```


Copyright© 2022 Ciena® Corporation

ciena-fp-sds.yang

```
module ciena-fp-sds {
  namespace "http://www.ciena.com/ns/yang/ciena-fp";
  prefix "ciena-fp-sds";

  import ciena-mef-fp {
    prefix "mef-fp";
  }

  import ciena-mef-classifier {
    prefix "classifier";
  }

  import ieee-dot1q-types
  {
    prefix dot1q-types;
  }

  organization
   "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
    E-mail:  yang@ciena.com
    Postal:  7035 Ridge Road
    Hanover, Maryland 21076
    U.S.A.
    Phone:   +1 800-921-1144
    Fax:     +1 410-694-5750";

  description
    "This YANG module defines Ciena's augmentations for the
    Flow Point model.

    Copyright (c) 2016 Ciena Corporation.  All rights
    reserved.

    All information contained herein is, and remains
    the property of Ciena Corporation. Dissemination of this
    information or reproduction of this material is strictly
    forbidden unless prior written permission is obtained from
    Ciena Corporation.";

  revision "2017-07-14" {
    description
      "Initial Version.";
    reference
      "RFC 6020: YANG - A Data Modeling Language for
       the Network Configuration Protocol (NETCONF).
       No specific reference; standard not available.";
  }

  /*
   * Config data
   */

  augment "/mef-fp:fps/mef-fp:fp" {
    description
```


Copyright© 2022 Ciena® Corporation

```
    "Ciena specific augmentations for FP";

  container ciena-fp-sds {
    description
      "Ciena SDS specific configuration data.";

    list per-classifier-settings {
      key "classifier";
      description
        "ciena-mef-fp contains only a single transform per tag per FP,
         though a single FP may encompass many classifiers.  The SDS
         platforms allow one transform per classifier (where there is one
         VS member per classifier). SDS platforms also allow for several
         transforms for just one VLAN member (one translation per inner
         tag value) ";

      leaf classifier {
        type classifier:classifier-ref;
        description
          "Reference to classifier";
      }

      leaf outer-to-vid {
        type uint16 {
           range "1..4095";
        }
        description
          "translate-tag for the outer vid of the packet";
      }

      leaf inner-to-vid {
        type uint16 {
           range "1..4095";
        }
        description
          "translate-tag for the inner vid of the packet";
      }

      leaf encap-cos-policy {
        type enumeration {
          enum fd-inherit {
            description
              "Inherit encap COS policy from parent forwarding domain";
          }
          enum port-inherit {
            description
              "Inherit encap COS policy from logical port";
          }
          enum fixed {
            description
              "Use fixed encap COS value";
          }
        }
        description
          "Encapsulation COS policy";
      }

      leaf encap-fixed-dot1dpri {
        type dot1q-types:priority-type;
        description
```


Copyright© 2022 Ciena® Corporation

```
            "fixed encapsulation COS";
        }

        leaf translate-tag {
          type uint16 {
             range "0..4094";
          }
          description
            "translate-tag";
        }
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-g8032.yang

```
module ciena-g8032 {
  namespace "http://www.ciena.com/ns/yang/ciena-g8032";
  prefix "ciena-g8032";

  import ciena-itu-g8032 {
    prefix "g8032";
  }

  import mef-cfm {
    prefix "cfm";
    revision-date 2012-04-16;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module is an extension to the ITU.T G.8032 Ethernet
     Ring Protocol (ERP) YANG model and defines Ciena's specific
     configuration parameters and deviations.";

  revision 2016-07-07 {
    description
      "This revision adds the following new leafs :
        - termination
        - wtr-timer
        - guard-timer
        - hold-off-time";
  }

  revision 2016-04-25 {
    description
      "Initial version";
  }

  typedef flush-propagation-state {
    type enumeration {
      enum on {
        description
          "Indicates on-state for flush-propagation.";
      }
      enum off {
        description
          "Indicates off-state for flush-propagation.";
      }
    }

    description
      "Indicates the state of flush-propagation for the ring.";
```


Copyright© 2022 Ciena® Corporation

```
  }

   augment "/g8032:g8032-rings/g8032:g8032-ring/g8032:" +
           "ring-ports/g8032:ring-port" {

    description
      "Adding hold off time as a configurable parameter to per port section
of  logical ring ";

       leaf hold-off-time {
            type g8032:hold-off-timer-type;
            description
              "The hold-off timer is used to coordinate the
              timing of protection switches.";
            reference
              "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.8";
       }
   }


  augment "/g8032:g8032-rings/g8032:g8032-ring/g8032:ring-ports" +
          "/g8032:ring-port" {

    description
      "Adding ciena-proprietary configurable parameters to the
       standard yang.";

      container cfm-service {
        description
          "Represents a cfm service that can be used to detect
           signal failure on a ring port. It is a metric used
           for ring protection switching.";

        uses cfm:maintenance-association-reference;
      }
  }

  augment "/g8032:g8032-rings/g8032:g8032-ring/g8032:" +
          "erp-instances/g8032:erp-instance" {

    description
      "Adding configurable parameters for instance.";

      leaf flush-propagate {
        type flush-propagation-state;
           default off;
         description
           "Specifies the state of flush-propagation for a ring.";
      }
  }

  augment "/g8032:g8032-rings/g8032:g8032-ring/g8032:" +
          "erp-instances/g8032:erp-instance/g8032:ports/g8032:port" {

    description
      "Adding sub-ring-termination state for subring instance.";

     leaf termination {
         type boolean;
           description
```


Copyright© 2022 Ciena® Corporation

```
            "specifies whether the port termination for the sub ring is true
or false.";
      }
  }

  augment "/g8032:g8032-rings/g8032:g8032-ring" {

    description
      "Adding wrt configurable parameter to logical ring section";

     leaf wtr-timer {
         type g8032:wtr-timer-type;
           description
              "The wait-to-restore timer associated with the ERP
               instance.";
           reference
              "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.4";
     }
  }

  augment "/g8032:g8032-rings/g8032:g8032-ring" {

    description
      "Adding guard-timer configurable parameter to logical ring section";

     leaf guard-timer {
         type g8032:guard-timer-type;
           description
              "The guard timer is used to prevent Ethernet ring nodes
              from acting upon outdated R-APS messages and prevents
              the possibility of forming a closed loop.";
           reference
              "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.5";
     }
  }

}
```


Copyright© 2022 Ciena® Corporation

ciena-icl.yang

```
module ciena-icl {

  namespace "http://www.ciena.com/ns/yang/ciena-icl";
  prefix ciena-icl;

  import ietf-yang-types {
    prefix yang;
  }

  import ciena-mef-fd {
    prefix mef-fd;
  }

  revision 2016-06-24 {
    description
     "Initial revision";
  }

//
// Typedefs related to ciena-icl
//

  typedef icl-state {
    type enumeration {
      enum enabled {
        description
         "Indicates icl-link is enabled";
      }
      enum disabled {
        description
         "Indicates icl-link is disabled";
      }
    }
  }

//
// Groupings related to ciena-icl
//

  grouping icl-key-group {

    leaf name {
      type string;
      description
       "This object specifies the name of the Inter-chassis link (ICL)." ;
    }

    leaf domain {
      type leafref {
        path "/mef-fd:fds/mef-fd:fd/mef-fd:name";
      }
      description
       "This object specifies the flooding domain for the ICL." ;
    }

    leaf domain-type {
      type leafref {
        path "/mef-fd:fds/mef-fd:fd/mef-fd:mode";
```


Copyright© 2022 Ciena® Corporation

```
      }
    }
  }

//
// Config data related to ciena-icl
//

  list icl-parameters {

    key "name domain-type domain";
    config true;
    description
     "List of configurable parameters for icl";

    uses icl-key-group;

    leaf remote-mac-address {
      type yang:mac-address;
      mandatory true;
      description
       "This object specifies the Inter-chassis link remote MAC address." ;
    }


    leaf cfm-service-primary {
      type string;
      description
       "This object specifies primary CFM service on the ICL." ;
    }

    leaf cfm-service-secondary {
      type string;
      description
       "This object specifies the name of the Inter-chassis link (ICL)." ;
    }

    leaf admin-state {
      type icl-state;
      description
       "This object specifies the admin state of the ICL." ;
    }

    leaf heartbeat-interval {
      type uint32;
      description
       "This object specifies the ICL heartbeat interval. The range is 1 to
30s and the default is 1s." ;
    }
  }

//
// Operational Data related to ciena-icl
//

  list icl-operational {

    key "name domain-type domain";
    config false;
    description
```


Copyright© 2022 Ciena® Corporation

```
     "List of operational data for icl";

    uses icl-key-group;

    leaf oper-state {
      type icl-state;
      config false;
      description
       "This object specifies the ICL operational state." ;
    }
    leaf status {
      type enumeration {
        enum init   {
          description
           "Indicates that the ICL is in init state. The node has
            just been enabled and is in the process of
            evaluating its state";
        }
        enum active {
          description
           "Indicates that the ICL is operationally up.";
        }
        enum failed {
          description
           "Indicates that the ICL is operationally down.";
        }
        enum down   {
          description
           "Indicates that the ICL is administratively down.";
        }
        enum none   {
          description
           "Indicates that the ICL is in an undefined state.";
        }
      }
      config false;
      description
       "This object specifies the status of the ICL." ;
    }

    leaf cfm-fault {
      type enumeration {
        enum no  {
          description
           "Indicates, if applicable, that there is no cfm fault.";
        }
        enum yes {
          description
           "Indicates, if applicable, that there is a cfm fault.";
        }
      }
      config false;
      description
       "This object specifies if CFM fault is set on the ICL." ;
    }
    leaf vpls-fault {
      type enumeration {
        enum no  {
          description
           "Indicates, if applicable, that there is no vpls fault.";
```


Copyright© 2022 Ciena® Corporation

```
        }
        enum yes {
          description
           "Indicates, if applicable, that there is a vpls fault.";
        }
      }
      config false;
      description
       "This object specifies if VPLS fault is set on the ICL." ;
    }
    leaf rx-timeout {
      type enumeration {
        enum no  {
          description
           "Indicates that there is no Rx timeout fault on the ICL.";
        }
        enum yes {
          description
           "Indicates that there is a Rx timeout fault on the ICL.";
        }
      }
      config false;
      description
       "This object specifies if Rx timeout fault is set on the
        ICL" ;
    }
    leaf interval-mismatch {
      type enumeration {
        enum no  {
          description
           "Indicates that there is no interval mismatch fault on
            the ICL.";
        }
        enum yes {
          description
           "Indicates that there is no interval mismatch fault on
            the ICL.";
        }
      }
      config false;
      description
       "This object specifies if interval mismatch fault is set on
        the ICL" ;
    }
  }

  list icl-statistics {

    key "name domain-type domain";
    config false;
    description
     "List of statistics for icl";

    uses icl-key-group;

    leaf up-time {
      type uint32;
      config false;
      description
```


Copyright© 2022 Ciena® Corporation

```
       "This object specifies total time for which the ICL has been
operationally down." ;
    }
    leaf total-down-time {
      type uint32;
      config false;
      description
       "This object specifies total time for which the ICL has been
operationally down." ;
    }
    leaf rx-frames {
      type yang:counter32;
      config false;
      description
      "This object specifies the number of CICP frames received by the ICL." ;
    }
    leaf tx-frames {
      type yang:counter32;
      config false;
      description
       "This object specifies the number of CICP frames transmitted by the
ICL." ;
    }
    leaf rx-unknown-frames {
      type yang:counter32;
      config false;
      description
       "This object specifies the number of unknown frames received by the
ICL." ;
    }
    leaf rx-htbt-frames {
      type yang:counter32;
      config false;
      description
       "This object specifies the number of Hearbeat frames received by the
ICL." ;
    }
    leaf tx-htbt-frames {
      type yang:counter32;
      config false;
      description
      "This object specifies the number of Hearbeat frames transmitted by the
ICL." ;
    }
    leaf tx-failed-frames {
      type yang:counter32;
      config false;
      description
       "This object specifies the number of failed to transmit frames." ;
    }
    leaf number-failures {
      type yang:counter32;
      config false;
      description
       "This object specifies the number of ICL failures." ;
    }
    leaf rx-config-mismatch {
      type yang:counter32;
      config false;
      description
```


Copyright© 2022 Ciena® Corporation

```
       "This object specifies the number of ICL config mismatch frames
received." ;
    }
  }

//
// RPCs related to ciena-icl
//

  rpc clear-link-statistics {
    description
     "Clear statistics for a specific ICL or for all configured
      ICLs";

    input {
      leaf name {
        type string;
        description
         "The name of the ICL whose statistics will be cleared";
      }
    }
  }

//
// Notifications related to ciena-icl
//

  notification icl-state-change {

    description
     "This notification is sent when there is a change in icl with
      respect to admin state and oper state";

    list state {

      key "name domain-type domain";
      uses icl-key-group;

      leaf admin-state {
        type icl-state;
        description
         "This object captures per-icl change in admin-state";
      }
      leaf oper-state {
        type icl-state;
        description
         "This object captures per-icl change in oper-state";
      }
    }
  }
}
// rpc
// clear statistics
// clear link statistics
```


Copyright© 2022 Ciena® Corporation

ciena-ieee-ethernet-port.yang

```
module ciena-ieee-ethernet-port {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-ieee-ethernet-
port";
    prefix "ciena-ieee-eth";

    import ietf-interfaces {
        prefix "if";
    }

    import ietf-yang-types {
        prefix "yt";
    }

    import iana-if-type {
        prefix "ianaift";
    }


    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
                 Hanover, Maryland 21076
                 U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
        "This YANG module contains the schema definition for the CSMA/CD
         defined Ethernet port.";

    revision "2014-09-15" {
        description
            "Initial Version.";
        reference
            "Add the reference here.";
    }

    augment "/if:interfaces/if:interface" {
        when "if:type = 'ianaift:ethernetCsmacd'";

       leaf advertised-duplex {
           type bits {
               bit full;
               bit half;
           }
           description
               "The duplex mode this port supports and advertised
                to the other end.";
       }

       leaf advertised-port-speed {
           type bits {
               bit ten-mega-bits;
               bit hundred-mega-bits;
```


Copyright© 2022 Ciena® Corporation

```
                bit one-giga-bits;
                bit ten-giga-bits;
            }
            description
               "The port-speed this port supports and advertised to the other
end.";
        }

        leaf advertised-flow-control-on {
                    type bits {
                        bit asym-tx;
                        bit asym-rx;
                        bit sym;
                    }
           description
               "The advertised flow control models this port
                    supports. If nothing or a blank string is specified then
it means
                     advertised-flow-control is off";
        }

        leaf duplex {
            type enumeration {
                enum half;
                enum full;
            }
                default full;
        }

        leaf port-speed {
             type enumeration {
                 enum auto;
                 enum 10Mb;
                 enum 100Mb;
                 enum 1Gb;
                 enum 10Gb;
                 enum 40Gb;
                 enum 100Gb;
             }
         }

         leaf flow-control {
             type enumeration {
                 enum off;
                 enum asym-tx;
                 enum asym-rx;
                 enum sym;
             }
             description
                 "Configures the ports flow control operation.";
         }

         leaf inter-packet-gap {
             type uint32 {
                 range "1..12";
             }
             units bytes;
             description
                 "Ethernet devices must allow a minimum idle period between
                  transmission of Ethernet packets known as the interpacket
```


Copyright© 2022 Ciena® Corporation

```
                gap (IPG).";
        }

        leaf auto-negotiation {
            type boolean;
            description
                 "A boolean value indicating if mode is auto or manual";
        }
    }

    // operational state parameters for Ethernet interfaces
    augment "/if:interfaces-state/if:interface" {
        when "if:type = 'ianaift:ethernetCsmacd'";

        leaf duplex {
            type enumeration {
                enum half;
                enum full;
            }
        }

        leaf flow-control {
            type enumeration {
                enum unknown;
                enum off;
                enum asym-tx;
                enum asym-rx;
                enum sym;
            }
            description
                "Shows ports flow control configuration.";
        }

        leaf port-type {
            type enumeration {
                enum ethernet;
                enum fast-ethernet;
                enum gig-ethernet;
                enum lag-port;
                enum unknown;
                enum gig-hundred-fx;
                enum triple-speed;
                enum ten-gig-ethernet;
                enum hundred-gig-ethernet;
                enum gig-ten-gig-ethernet;
            }
            description
                "The port type for the port";
        }

        leaf operational-status {
            type enumeration {
                enum up;
                enum down;
            }
            description
                "The current operational state of the port.";
        }

        leaf mac-address {
```


Copyright© 2022 Ciena® Corporation

```
            type yt:mac-address;
            description
                "The MAC address of this port.";
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-ieee-lag-deviation.yang

```
module ciena-ieee-lag-deviation {

  namespace "http://www.ciena.com/ns/yang/ciena-ieee-lag-dev";
  prefix ciena-ieee-lag-dev;

  import ciena-ieee-lag {
    prefix ciena-ieee-lag;
  }
  import ietf-interfaces {
    prefix "if";
  }

  revision 2017-07-24 {
    description
    "Removed previous deviation that restricted characters in interface name";
  }

  revision 2016-09-13 {
    description
     "Deviation added to restrict invalid characters in interface name";
  }

  revision 2016-02-29 {
    description
     "Initial Revision";
  }

  //
  // Config data that is supported only as read-only in 6.16
  //

  deviation /if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:agg-collector-max-delay {
    deviate not-supported;
  }

  deviation /if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:member-ports/ciena-ieee-lag:agg-port-actor-system-priority {
    deviate not-supported;
  }

  deviation /if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:member-ports/ciena-ieee-lag:agg-port-actor-admin-key {
    deviate not-supported;
  }

  deviation /if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:member-ports/ciena-ieee-lag:agg-port-partner-admin-system-priority {
    deviate not-supported;
  }

  deviation /if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:member-ports/ciena-ieee-lag:agg-port-partner-admin-system-ID {
    deviate not-supported;
  }

  deviation /if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:member-ports/ciena-ieee-lag:agg-port-partner-admin-key {
```

```
    deviate not-supported;
  }

  deviation /if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:member-ports/ciena-ieee-lag:agg-port-actor-port-priority {
    deviate not-supported;
  }

  deviation /if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:member-ports/ciena-ieee-lag:agg-port-partner-admin-port {
    deviate not-supported;
  }

  deviation /if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:member-ports/ciena-ieee-lag:agg-port-partner-admin-port-priority {
    deviate not-supported;
  }

  deviation /if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:member-ports/ciena-ieee-lag:agg-port-actor-admin-state/ciena-ieee-
lag:aggregation {
    deviate not-supported;
  }

  deviation /if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:member-ports/ciena-ieee-lag:agg-port-actor-admin-state/ciena-ieee-
lag:synchronization {
    deviate not-supported;
  }

  deviation /if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:member-ports/ciena-ieee-lag:agg-port-actor-admin-state/ciena-ieee-
lag:collecting {
    deviate not-supported;
  }

  deviation /if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:member-ports/ciena-ieee-lag:agg-port-actor-admin-state/ciena-ieee-
lag:distributing {
    deviate not-supported;
  }

  deviation /if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:member-ports/ciena-ieee-lag:agg-port-actor-admin-state/ciena-ieee-
lag:defaulted {
    deviate not-supported;
  }

  deviation /if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:member-ports/ciena-ieee-lag:agg-port-actor-admin-state/ciena-ieee-
lag:expired {
    deviate not-supported;
  }

  deviation /if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:member-ports/ciena-ieee-lag:agg-port-partner-admin-state/ciena-ieee-
lag:lacp-activity {
    deviate not-supported;
  }
```


Copyright© 2022 Ciena® Corporation

```
  deviation /if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:member-ports/ciena-ieee-lag:agg-port-partner-admin-state/ciena-ieee-
lag:lacp-timeout {
    deviate not-supported;
  }

  deviation /if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:member-ports/ciena-ieee-lag:agg-port-partner-admin-state/ciena-ieee-
lag:aggregation {
    deviate not-supported;
  }

  deviation /if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:member-ports/ciena-ieee-lag:agg-port-partner-admin-state/ciena-ieee-
lag:synchronization {
    deviate not-supported;
  }

  deviation /if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:member-ports/ciena-ieee-lag:agg-port-partner-admin-state/ciena-ieee-
lag:collecting {
    deviate not-supported;
  }

  deviation /if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:member-ports/ciena-ieee-lag:agg-port-partner-admin-state/ciena-ieee-
lag:distributing {
    deviate not-supported;
  }

  deviation /if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:member-ports/ciena-ieee-lag:agg-port-partner-admin-state/ciena-ieee-
lag:defaulted {
    deviate not-supported;
  }

  deviation /if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:member-ports/ciena-ieee-lag:agg-port-partner-admin-state/ciena-ieee-
lag:expired {
    deviate not-supported;
  }

  //
  // Config data that is not supported in 6.16
  //

  deviation /if:interfaces/if:interface/ciena-ieee-lag:agg/ciena-ieee-
lag:member-ports/ciena-ieee-lag:agg-port-protocol-DA {
    deviate not-supported;
  }

  //
  // Read-only data not supported in 6.16
  //

  deviation /if:interfaces-state/if:interface/ciena-ieee-lag:agg-state/
ciena-ieee-lag:distribution-ports/ciena-ieee-lag:agg-port-debug-actor-churn-
state {
    deviate not-supported;
  }
```


Copyright© 2022 Ciena® Corporation

```
  deviation /if:interfaces-state/if:interface/ciena-ieee-lag:agg-state/
ciena-ieee-lag:distribution-ports/ciena-ieee-lag:agg-port-debug-partner-
churn-state {
    deviate not-supported;
  }

  deviation /if:interfaces-state/if:interface/ciena-ieee-lag:agg-state/
ciena-ieee-lag:distribution-ports/ciena-ieee-lag:agg-port-debug-actor-churn-
count {
    deviate not-supported;
  }

  deviation /if:interfaces-state/if:interface/ciena-ieee-lag:agg-state/
ciena-ieee-lag:distribution-ports/ciena-ieee-lag:agg-port-debug-partner-
churn-count {
    deviate not-supported;
  }

  deviation /if:interfaces-state/if:interface/ciena-ieee-lag:agg-state/
ciena-ieee-lag:distribution-ports/ciena-ieee-lag:agg-port-debug-actor-sync-
transition-count {
    deviate not-supported;
  }

  deviation /if:interfaces-state/if:interface/ciena-ieee-lag:agg-state/
ciena-ieee-lag:distribution-ports/ciena-ieee-lag:agg-port-debug-partner-
sync-transition-count {
    deviate not-supported;
  }

  deviation /if:interfaces-state/if:interface/ciena-ieee-lag:agg-state/
ciena-ieee-lag:distribution-ports/ciena-ieee-lag:agg-port-debug-actor-
change-count {
    deviate not-supported;
  }

  deviation /if:interfaces-state/if:interface/ciena-ieee-lag:agg-state/
ciena-ieee-lag:distribution-ports/ciena-ieee-lag:agg-port-debug-partner-
change-count {
    deviate not-supported;
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-ieee-lag.yang

```
module ciena-ieee-lag {

  namespace "http://www.ciena.com/ns/yang/ciena-ieee-lag";
  prefix ciena-ieee-lag;

  import ietf-interfaces {
    prefix "if";
  }
  import ietf-yang-types   {
    prefix "yang";
  }
  import iana-if-type {
    prefix "ianaift";
  }

  organization
   "Ciena Corporation";

  contact
   "Web URL: http://www.ciena.com/
    E-mail:  yang@ciena.com
    Postal:  7035 Ridge Road
             Hanover, Maryland 21076
             U.S.A.
    Phone:   +1 800-921-1144
    Fax:     +1 410-694-5750";

  description
   "This YANG module defines link aggregation for the NETCONF
    protocol.";

  revision 2016-02-19 {
    description
     "Initial revision";
    reference
     "IEEE Standard for Local and metropolitan area networks --
      Link Aggregation - 802.1AX";
  }

  //
  // Type definitions related to LACP
  //

  typedef lacp-key {
    type uint16;
    description
     "The Actor or Partner Key value.";
  }

  // CHURN STATE

  identity churn-state {
    description
     "The state of the Churn Detection machine.";
  }

  identity no-churn {
    base churn-state;
```


Copyright© 2022 Ciena® Corporation

```
    description
     "Indicates that there is no churn detected by the
      Churn Detection State Machine";
  }

  identity churn {
    base churn-state;
    description
     "Indicates that there hasn't been a convergence detected
      by the Churn Detection State Machine within a specified
      time.";
  }
  identity churn-monitor {
    base churn-state;
    description
     "Indicates that the Churn Detection State Machine is within
      the specified time for detecting convergence.";
  }


  // LACP-ACTIVITY

  identity lacp-activity-type {
    description
     "Active or Passive - Indicates whether LACP PDUs are
      initiated always or only in response to another LACP PDU.
      The IEEE 802.1ax Link Aggregation will always operate in
      Active mode.";
  }

  identity lacp-active {
    base lacp-activity-type;
    description
     "Indicates lacp is in active mode";
  }

  identity lacp-passive {
    base lacp-activity-type;
    description
     "Indicates lacp is in passive mode";
  }


  // LACP-TIMEOUT

  identity lacp-timeout-type {
    description
     "Long or Short - Indicates time to wait before sending
      LACP PDUs. The IEEE 802.1ax Link Aggregation will always
      use the long timeout.";
  }

  identity lacp-timeout-long {
    base lacp-timeout-type;
    description
     "Indicates Long wait time before sending LACP PDUs";

  }

  identity lacp-timeout-short {
```


Copyright© 2022 Ciena® Corporation

```
 base lacp-timeout-type;
 description
  "Indicates Short wait time before sending LACP PDUs";

}


// AGGREGATION

identity aggregation-type {
  description
   "Individual or Aggregatable - Indicates wherher this port
    can form an aggregation with other ports, or if it can
    only form an aggregation with itself. The port in
    one-adapter IEEE 802.1ax Link Aggregation will be marked
    as Individual, or Aggregatable if there is more than one
    port.";
}

identity aggregatable {
  base aggregation-type;
  description
   "Indicates that this port can only form aggregation
    with itself";
}

identity individual {
  base aggregation-type;
  description
   "Indicates that this port can form an aggregation
    with other ports";
}


// SYNCHRONIZATION

identity synchronization-type {
  description
   "IN-SYNC or OUT_OF_SYNC - Indicates whether the aggregation
    has determined it has reached synchronization with the
    partner.";
}

identity in-sync {
  base synchronization-type;
  description
   "Indicates aggregation has not reached synchronization
    with the partner";
}

identity out-of-sync {
  base synchronization-type;
  description
   "Indicates aggregation has not reached synchronization
    with the partner";
}


// COLLECTING
```

```
  identity collecting-type {
    description
    "Is_Collecting or Is_Not_Collecting - Indicates whether the aggregation is
      receiving packets.";
  }

  identity is-collecting {
    base collecting-type;
    description
     "Indicates aggregation is receiving packets";
  }

  identity is-not-collecting {
    base collecting-type;
    description
     "Indicates aggregation is not receiving packets";
  }


  // DISTRIBUTING

  identity distributing-type {
    description
     "Is_Distributing or Is_Not_Distributing - Indicates whether the
aggregation is
      sending packets.";
  }

  identity is-distributing {
    base distributing-type;
    description
     "Indicates aggregation is sending packets";
  }

  identity is-not-distributing {
    base distributing-type;
    description
     "Indicates aggregation is not sending packets";
  }


  // DEFAULTED

  identity defaulted-type {
    description
    "Default_In_Use or Default_Not_In_Use - Indicates whether the IEEE 802.1ax
      Link Aggregation is using default values for partner
      information.";
  }

  identity default-in-use {
    base defaulted-type;
    description
     "Indicates default values for partner are being used";
  }

  identity default-not-in-use {
    base defaulted-type;
    description
     "Indicates default values for partner are not being
```


Copyright© 2022 Ciena® Corporation

```
      used";
  }


  // EXPIRED

  identity expired-type {
    description
     "Is_Expired or Is_Not_Expired - Indicates whether the IEEE 802.1ax
      Link Aggregation is operated in expired mode.";
  }

  identity is-expired {
    base expired-type;
    description
     "Indicates operation is in expired mode";
  }

  identity is-not-expired {
    base expired-type;
    description
     "Indicates operation is not in expired mode";
  }


  // AGG PORT RX STATE

  identity agg-port-rx-state {
    description
     "This attribute holds the value 'currentRx' if the Receive
      state machine for the Aggregation Port is in the
      CURRENT state, 'expired' if the Receive state machine
      is in the EXPIRED state, 'defaulted' if the Receive state
      machine is in the DEFAULTED state, 'initialize' if the
      Receive state machine is in the INITIALIZE state,
      'lacpDisabled' if the Receive state machine is in the
      LACP_DISABLED state, or 'portDisabled' if the Receive
      state machine is in the PORT_DISABLED state.
      This value is read-only.";
  }

  identity current-rx {
    base agg-port-rx-state;
    description
     "Indicates that the Receive State Machine is in
      the CURRENT state.";
  }

  identity expired-rx {
    base agg-port-rx-state;
    description
     "Indicates that the Receive State Machine is in
      the EXPIRED state.";
  }

  identity defaulted-rx {
    base agg-port-rx-state;
    description
     "Indicates that the Receive State Machine is in
      the DEFAULTED state.";
```


Copyright© 2022 Ciena® Corporation

```
}

identity initialize-rx {
  base agg-port-rx-state;
  description
   "Indicates that the Receive State Machine is in
    the INITIALIZE state.";
}

identity lacp-disabled-rx {
  base agg-port-rx-state;
  description
   "Indicates that the Receive State Machine is in
    the LACP_DISABLED state.";
}

identity port-disabled-rx {
  base agg-port-rx-state;
  description
   "Indicates that the Receive State Machine is in
    the PORT_DISABLED state.";
}


// AGG PORT MUX STATE

identity agg-port-mux-state {
  description
   "This attribute holds the value 'detached' if the Mux
    state machine (5.4.14) for the Aggregation Port is
    in the DETACHED state, 'waiting' if the Mux state machine
    is in the WAITING state, 'attached' if the Mux state
    machine for the Aggregation Port is in the ATTACHED
    state, 'collecting' if the Mux state machine for the
    Aggregation Port is in the COLLECTING state, 'distributing'
    if the Mux state machine for the Aggregation Port is
    in the DISTRIBUTING state, and 'collectingDistributing'
    if the Mux state machine for the Aggregation Port is in
    the COLLECTING_DISTRIBUTING state.
    This value is read-only.";
}

identity detached-mux {
  base agg-port-mux-state;
  description
   "Indicates that the Mux State Machine for the Aggregation
    Port is in DETACHED state.";
}

identity waiting-mux {
  base agg-port-mux-state;
  description
   "Indicates that the Mux State Machine for the Aggregation
    Port is in WAITING state.";

}

identity attached-mux {
  base agg-port-mux-state;
  description
```


Copyright© 2022 Ciena® Corporation

```
    "Indicates that the Mux State Machine for the Aggregation
     Port is in ATTACHED state.";
}

identity collecting-mux {
  base agg-port-mux-state;
  description
   "Indicates that the Mux State Machine for the Aggregation
    Port is in ATTACHED state.";

}

identity distributing-mux {
  base agg-port-mux-state;
  description
   "Indicates that the Mux State Machine for the Aggregation
    Port is in DISTRIBUTING state.";
}

identity collecting-distributing-mux {
  base agg-port-mux-state;
  description
   "Indicates that the Mux State Machine for the Aggregation
    Port is in COLLECTING_DISTRIBUTING state.";
}


//
// Groupings related to LACP
//
grouping lacp-state {
  leaf lacp-activity    {
    type identityref {
      base lacp-activity-type;
    }
    description
     "Active or Passive - Indicates whether LACP PDUs are
      initiated always or only in response to another LACP PDU.
      The IEEE 802.1ax Link Aggregation will always operate in
      Active mode.";
  }
  leaf lacp-timeout     {
    type identityref {
      base lacp-timeout-type;
    }
    description
     "Long or Short - Indicates time to wait before sending
      LACP PDUs. The IEEE 802.1ax Link Aggregation will always
      use the long timeout.";
  }
  leaf aggregation     {
    type identityref {
      base aggregation-type;
    }
    description
     "Individual or Aggregatable - Indicates wherher this port
      can form an aggregation with other ports, or if it can
      only form an aggregation with itself. The port in
      one-adapter IEEE 802.1ax Link Aggregation will be marked
      as Individual, or Aggregatable if there is more than one
```


Copyright© 2022 Ciena® Corporation

```
        port.";
    }
    leaf synchronization {
      type identityref {
        base synchronization-type;
      }
      description
       "IN-SYNC or OUT_OF_SYNC - Indicates whether the aggregation
        has determined it has reached synchronization with the
        partner.";
    }
    leaf collecting      {
      type identityref {
        base collecting-type;
      }
      description
       "Enabled or Disabled - Indicates whether the aggregation is
        receiving packets.";
    }
    leaf distributing    {
      type identityref {
        base distributing-type;
      }
      description
       "Enabled or Disabled - Indicates whether the aggregation is
        sending packets.";
    }
    leaf defaulted       {
      type identityref {
        base defaulted-type;
      }
      description
       "True or False - Indicates whether the IEEE 802.1ax
        Link Aggregation is using default values for partner
        information.";
    }
    leaf expired         {
      type identityref {
        base expired-type;
      }
      description
       "True or False - Indicates whether the IEEE 802.1ax
        Link Aggregation is operated in expired mode.";
    }
    description
     "The Actor and Partner State values from the LACPDU.";
  }

  grouping agg-port-parameters-group {
    description
     "per-port aggregation parameter";

    leaf agg-port-actor-system-priority {
      type uint8;
      description
       "A 2-octet read-write value used to define the priority
        value associated with the Actor's System ID.";
      reference
       "[802.1ax] 6.3.2.1.2";
    }
```


Copyright© 2022 Ciena® Corporation

```
leaf agg-port-actor-admin-key {
  type lacp-key;
  description
   "The current administrative value of the Key for the
    Aggregation Port. This is a 16-bit read-write value.
    The meaning of particular Key values is of local
    significance.";
  reference
   "[802.1ax] 6.3.2.1.4";
}

leaf agg-port-partner-admin-system-priority {
  type uint8;
  description
   "A 2-octet read-write value used to define the administrative
    value of priority associated with the Partner's System ID.
    The assigned value is used, along with the value of
    partner-admin-system-ID, partner-admin-key,
    partner-admin-port, and partner-admin-port-priority,
    in order to achieve manually configured aggregation.";
  reference
   "[802.1ax] 6.3.2.1.6";
}

leaf agg-port-partner-admin-system-ID {
  type yang:mac-address;
  description
   "A 6-octet read-write MACAddress value representing
    the administrative value of the Aggregation Port's protocol
    Partner's System ID. The assigned value is used, along with
    the value of agg-port-partnerAdminSystemPriority,
    partner-admin-key, partner-admin-port,
    and partner-admin-port-priority, in order to
    achieve manually configured aggregation.";
  reference
   "[802.1ax] 6.3.2.1.8";
}

leaf agg-port-partner-admin-key {
  type lacp-key;
  description
   "The current administrative value of the Key for the
    protocol Partner. This is a 16-bit read-write value.
    The assigned value is used, along with the value of
    partner-admin-system-priority, partner-admin-system-ID,
    artner-admin-port, and partner-admin-port-priority,
    in order to achieve manually configured aggregation.";
  reference
   "[802.1ax] 6.3.2.1.10";
}

leaf agg-port-actor-port-priority {
  type uint8;
  description
   "The priority value assigned to this Aggregation Port.
    This 16-bit value is read-write.";
  reference
   "[802.1ax] 6.3.2.1.15";
}
```


Copyright© 2022 Ciena® Corporation

```
    leaf agg-port-partner-admin-port {
      type uint16;
      description
       "The current administrative value of the port number
        for the protocol Partner. This is a 16-bit read-write value.
        The assigned value is used, along with the value of
        partner-admin-system-priority,partner-admin-system-ID,
        partner-admin-key, and partner-admin-port-priority,
        in order to achieve manually configured aggregation.";
      reference
       "[802.1ax] 6.3.2.1.16";
    }

    leaf agg-port-partner-admin-port-priority {
      type uint8;
      description
       "The current administrative value of the port priority
        for the protocol Partner. This is a 16-bit read-write value.
        The assigned value is used, along with the value of
        partner-admin-system-priority, partner-admin-system-ID,
        partner-admin-key, and partner-admin-port,
        in order to achieve manually configured aggregation.";
      reference
       "[802.1ax] 6.3.2.1.18";
    }

    container agg-port-actor-admin-state {
      uses lacp-state;
      description
       "A string of 8 bits, corresponding to the administrative
        values of Actor_State (5.4.2) as transmitted by the Actor in
        LACPDUs. The first bit corresponds to bit 0 of Actor_State
        (LACP_Activity), the second bit corresponds to bit 1
        (LACP_Timeout), the third bit corresponds to bit 2
        (Aggregation), the fourth bit corresponds to bit 3
        (Synchronization), the fifth bit corresponds to bit 4
        (Collecting), the sixth bit corresponds to bit 5
        (Distributing), the seventh bit corresponds to bit 6
        (Defaulted), and the eighth bit corresponds to bit 7
        (Expired). These values allow administrative control over the
        values of LACP_Activity, LACP_Timeout and Aggregation.
        This attribute value is read-write.";
      reference
       "[802.1ax] 6.3.2.1.20";
    }

    container agg-port-partner-admin-state {
      uses lacp-state;
      description
       "A string of 8 bits, corresponding to the current
        administrative value of Actor_State for the protocol Partner.
        The bit allocations are as defined in 6.3.2.1.20. This
        attribute value is read-write. The assigned value is used in
        order to achieve manually configured aggregation.";
      reference
       "[802.1ax] 6.3.2.1.22";
    }
  }
```


Copyright© 2022 Ciena® Corporation

```
  grouping agg-port-operational-group {
    description
     "Per-port LACP operational data";

    leaf agg-port-actor-system-ID {
      type yang:mac-address;
      description
       "A 6-octet read-only MAC address value that defines
        the value of the System ID for the System that contains this
        Aggregation Port.";
      reference
       "[802.1ax] 6.3.2.1.3";
    }

    leaf agg-port-actor-oper-key {
      type lacp-key;
      description
       "The current operational value of the Key for the
        Aggregation Port. This is a 16-bit read-only value.
        The meaning of particular Key values is of local
        significance.";
      reference
       "[802.1ax] 6.3.2.1.5";
    }

    leaf agg-port-partner-oper-system-priority {
      type uint8;
      description
       "A 2-octet read-only value indicating the operational value
        of priority associated with the Partner's System ID. The
        value of this attribute may contain the manually configured
        value carried in agg-port-partner-admin-system-priority
        if there is no protocol Partner.";
      reference
       "[802.1ax] 6.3.2.1.7";
    }

    leaf agg-port-partner-oper-system-ID {
      type yang:mac-address;
      description
       "A 6-octet read-only MACAddress value representing
        the current value of the Aggregation Port's protocol
        Partner's System ID. A value of zero indicates that there is
        no known protocol Partner. The value of this attribute may
        contain the manually configured value carried in
        partner-admin-system-ID if there is no protocol Partner.";
      reference
       "[802.1ax] 6.3.2.1.9";
    }

    leaf agg-port-partner-oper-key {
      type lacp-key;
      description
       "The current operational value of the Key for the
        protocol Partner. The value of this attribute may contain
        the manually configured value carried in
        partner-admin-key if there is no protocol Partner.
        This is a 16-bit read-only value.";
      reference
       "[802.1ax] 6.3.2.1.11";
```

```
}

leaf agg-port-selected-agg-ID {
  type if:interface-ref;
  description
   "The identifier value of the Aggregator that this Aggregation
    Port has currently selected. Zero indicates that the
    Aggregation Port has not selected an Aggregator, either
    because it is in the process of detaching from an Aggregator
    or because there is no suitable Aggregator available for it
    to select. This value is read-only.";
  reference
   "[802.1ax] 6.3.2.1.12";
}

leaf agg-port-attached-agg-ID {
  type if:interface-ref;
  description
   "The identifier value of the Aggregator that this Aggregation
    Port is currently attached to. Zero indicates that the
    Aggregation Port is not currently attached to an Aggregator.
    This value is read-only.";
  reference
   "[802.1ax] 6.3.2.1.13";
}

leaf agg-port-actor-port {
  type uint16;
  description
   "The port number locally assigned to the Aggregation Port.
    The port number is communicated in LACPDUs as the
    Actor_Port. This value is read-only.";
  reference
   "[802.1ax] 6.3.2.1.14";
}

leaf agg-port-partner-oper-port {
  type uint16;
  description
   "The operational port number assigned to this Aggregation
    Port by the Aggregation Port's protocol Partner. The value
    of this attribute may contain the manually configured value
    carried in partner-admin-port if there is no protocol
    Partner. This 16-bit value is read-only.";
  reference
   "[802.1ax] 6.3.2.1.17";
}

leaf agg-port-partner-oper-port-priority {
  type uint8;
  description
   "The priority value assigned to this Aggregation Port
    by the Partner. The value of this attribute may contain the
    manually configured value carried in
    partner-admin-port-priority if there is no
    protocol Partner. This 16-bit value is read-only.";
  reference
   "[802.1ax] 6.3.2.1.19";
}
```


Copyright© 2022 Ciena® Corporation

```
    container agg-port-actor-oper-state {
      uses lacp-state;
      description
       "A string of 8 bits, corresponding to the current
        operational values of Actor_State as transmitted by the
        Actor in LACPDUs. The bit allocations are as defined in
        6.3.2.1.20. This attribute value is read-only.";
      reference
       "[802.1ax] 6.3.2.1.21";
    }

    container agg-port-partner-oper-state {
      uses lacp-state;
      description
       "A string of 8 bits, corresponding to the current values of
        Actor_State in the most recently received LACPDU transmitted
        by the protocol Partner. The bit allocations are as defined
        in 6.3.2.1.20. In the absence of an active protocol Partner,
        this value may reflect the manually configured value
        partner-admin-state. This attribute value is read-only.";
      reference
       "[802.1ax] 6.3.2.1.23";
    }

    leaf agg-port-aggregate-or-individual {
      type boolean;
      description
       "A read-only Boolean value indicating whether the
        Aggregation Port is able to Aggregate ('TRUE') or is
        only able to operate as an Individual link ('FALSE').";
      reference
       "6.3.2.1.24";
    }
  }

  grouping agg-port-statistics-group {
    description
     "per-port statistics for ports belonging to an Aggregator";

    leaf agg-port-stats-LACPDUs-rx {
      type yang:counter32;
      description
       "The number of valid LACPDUs received on this
        Aggregation Port. This value is read-only.";
      reference
       "[802.1ax] 6.3.3.1.2";
    }

    leaf agg-port-stats-marker-PDUs-Rx {
      type yang:counter32;
      description
       "The number of valid Marker PDUs received on this
        Aggregation Port. This value is read-only.";
      reference
       "[802.1ax] 6.3.3.1.3";

    }

    leaf agg-port-stats-marker-response-PDUs-rx {
      type yang:counter32;
```


Copyright© 2022 Ciena® Corporation

```
    description
     "The number of valid Marker Response PDUs received on this
      Aggregation Port. This value is read-only.";
    reference
     "[802.1ax] 6.3.3.1.4";
  }

  leaf agg-port-stats-unknown-rx {
    type yang:counter32;
    description
     "The number of frames received that either:
      - carry the Slow Protocols Ethernet Type value
        (IEEE Std 802.3 Annex 57A.4), but contain an
        unknown PDU, or:
      - are addressed to the Slow Protocols group MAC
        Address (IEEE Std 802.3 Annex 57A.4), but do
        not carry the Slow Protocols Ethernet Type.
      This value is read-only.";
    reference
     "[802.1ax] 6.3.3.1.5";
  }

  leaf agg-port-stats-illegal-rx {
    type yang:counter32;
    description
     "The number of frames received that carry the Slow
      Protocols Ethernet Type value (IEEE Std 802.3 Annex
      57A.4), but contain a badly formed PDU or an illegal
      value of Protocol Subtype (IEEE Std 802.3 Annex 57A.4).
      This value is read-only.";
    reference
     "[802.1ax] 6.3.3.1.6";
  }

  leaf agg-port-stats-LACPDUs-tx {
    type yang:counter32;
    description
     "The number of LACPDUs transmitted on this
      Aggregation Port. This value is read-only.";
    reference
     "[802.1ax] 6.3.3.1.7";
  }

  leaf agg-port-stats-marker-PDUs-tx {
    type yang:counter32;
    description
     "The number of Marker PDUs transmitted on this
      Aggregation Port. This value is read-only.";
    reference
     "[802.1ax] 6.3.3.1.8";
  }

  leaf agg-port-stats-marker-response-PDUs-tx {
    type yang:counter32;
    description
     "The number of Marker Response PDUs transmitted
      on this Aggregation Port. This value is read-only.";
    reference
     "[802.1ax] 6.3.3.1.9";
  }
```


Copyright© 2022 Ciena® Corporation

```
}

grouping agg-port-debug-info-group {

  description
   "A list of the debug parameters for a port.";

  leaf agg-port-debug-rx-state {
    type identityref {
      base agg-port-rx-state;
    }
    description
     "This attribute holds the value 'currentRx' if the Receive
      state machine for the Aggregation Port is in the
      CURRENT state, 'expired' if the Receive state machine
      is in the EXPIRED state, 'defaulted' if the Receive state
      machine is in the DEFAULTED state, 'initialize' if the
      Receive state machine is in the INITIALIZE state,
      'lacpDisabled' if the Receive state machine is in the
      LACP_DISABLED state, or 'portDisabled' if the Receive
      state machine is in the PORT_DISABLED state.
      This value is read-only.";
    reference
     "[802.1ax] 6.3.4.1.2";
  }

  leaf agg-port-debug-last-rx-time {
    type yang:timeticks;
    description
     "The value of aTimeSinceSystemReset (See IEEE Std 802.3 Annex
      F.2.1) when the last LACPDU was received by this Aggregation
      Port. This value is read-only.";
    reference
     "[802.1ax] 6.3.4.1.3";
  }

  leaf agg-port-debug-mux-state {
    type identityref {
      base agg-port-mux-state;
    }
    description
     "This attribute holds the value 'detached' if the Mux
      state machine (5.4.14) for the Aggregation Port is
      in the DETACHED state, 'waiting' if the Mux state machine
      is in the WAITING state, 'attached' if the Mux state
      machine for the Aggregation Port is in the ATTACHED
      state, 'collecting' if the Mux state machine for the
      Aggregation Port is in the COLLECTING state, 'distributing'
      if the Mux state machine for the Aggregation Port is
      in the DISTRIBUTING state, and 'collectingDistributing'
      if the Mux state machine for the Aggregation Port is in
      the COLLECTING_DISTRIBUTING state.
      This value is read-only.";
    reference
     "[802.1ax] 6.3.4.1.4";
  }

  leaf agg-port-debug-mux-reason {
    type boolean;
    description
```


Copyright© 2022 Ciena® Corporation

```
      "A human-readable text string indicating the reason
       for the most recent change of Mux machine state.
       This value is read-only.";
    reference
      "[802.1ax] 6.3.4.1.5";
  }

  leaf agg-port-debug-actor-churn-state {
    type identityref {
      base churn-state;
    }
    description
      "The state of the Actor Churn Detection machine
       (5.4.17) for the Aggregation Port. A value of 'noChurn'
       indicates that the state machine is in either the
       NO_ACTOR_CHURN or the ACTOR_CHURN_MONITOR
       state, and 'churn' indicates that the state machine is in the
       ACTOR_CHURN state. This value is read-only.";
    reference
      "[802.1ax] 6.3.4.1.6";
  }

  leaf agg-port-debug-partner-churn-state {
    type identityref {
      base churn-state;
    }
    description
      "The state of the Partner Churn Detection machine
       (5.4.17) for the Aggregation Port. A value of 'noChurn'
       indicates that the state machine is in either the
       NO_PARTNER_CHURN or the PARTNER_CHURN_MONITOR
       state, and 'churn' indicates that the state machine is
       in the PARTNER_CHURN state.
       This value is read-only.";
    reference
      "[802.1ax] 6.3.4.1.7";
  }

  leaf agg-port-debug-actor-churn-count {
    type yang:counter32;
    description
      "Count of the number of times the Actor Churn state
       machine has entered the ACTOR_CHURN state.
       This value is read-only.";
    reference
      "[802.1ax] 6.3.4.1.8";
  }

  leaf agg-port-debug-partner-churn-count {
    type yang:counter32;
    description
      "Count of the number of times the Partner Churn
       state machine has entered the PARTNER_CHURN state.
       This value is read-only.";
    reference
      "[802.1ax] 6.3.4.1.9";
  }

  leaf agg-port-debug-actor-sync-transition-count {
    type yang:counter32;
```


Copyright© 2022 Ciena® Corporation

```
      description
       "Count of the number of times the Actor's Mux state
        machine (5.4.15) has entered the IN_SYNC state.
        This value is read-only.";
      reference
       "[802.1ax] 6.3.4.1.10";
    }

    leaf agg-port-debug-partner-sync-transition-count {
      type yang:counter32;
      description
       "Count of the number of times the Partner's Mux
        state machine (5.4.15) has entered the IN_SYNC state.
        This value is read-only.";
      reference
       "[802.1ax] 6.3.4.1.11";
    }

    leaf agg-port-debug-actor-change-count {
      type yang:counter32;
      description
       "Count of the number of times the Actor's perception of
        the LAG ID for this Aggregation Port has changed.
        This value is read-only.";
      reference
       "[802.1ax] 6.3.4.1.12";
    }

    leaf agg-port-debug-partner-change-count {
      type yang:counter32;
      description
       "Count of the number of times the Partner's perception of
        the LAG ID (see 5.3.6.1) for this Aggregation Port has
        changed. This value is read-only.";
      reference
       "[802.1ax] 6.3.4.1.13";
    }
  }

  //
  // Data definitions related to LACP
  //

  container agg-global {
    description
     "Contains configurable parameters global to all Aggregators in the
      system";

    leaf global-admin-state {
      type enumeration {
        enum disabled {
          description
            "Disables aggregation on a system wide
            basis.";
        }
        enum enabled {
          description
            "Enables aggregation on a system wide
            basis.";
        }
```


Copyright© 2022 Ciena® Corporation

```
    }
    description
     "Enables or Disables aggregation on a system wide
      basis";
  }
}

container agg-global-state {
  config false;
  description
   "Contains Information global to all Aggregators in the system.";
  leaf tables-last-changed {
    type yang:timeticks;
    config false;
    description
     "This object indicates the time of the
      most recent change to the agg-parameters or
      agg-port-parameters";
  }
}

augment "/if:interfaces/if:interface" {
  when "if:type = 'ianaift:ieee8023adLag'" {
    description
     "Augment only if the interface type is ieee8023adLag.";
  }
  description
   "Augmenting LAG to interfaces.";

  container agg {
    description
     "Configurable parameters for each Aggregator.";
    leaf agg-actor-system-priority {
      type uint16;
      description
       "A 2-octet read-write value indicating the priority value
        associated with the Actor's System ID.";
      reference
       "[802.1ax] 6.3.1.1.5";
    }

    leaf agg-actor-admin-key {
      type lacp-key;
      description
       "The current administrative value of the Key for the
        Aggregator. The administrative Key value may differ from
        the operational Key value for the reasons discussed in
        5.6.2. This is a 16-bit, read-write value. The meaning of
        particular Key values is of local significance.";
      reference
       "[802.1ax] 6.3.1.1.7";
    }

    leaf agg-collector-max-delay {
      type uint16;
      description
       "The value of this 16-bit read-write attribute defines
        the maximum delay, in tens of microseconds, that
        may be imposed by the Frame Collector between
        receiving a frame from an Aggregator Parser, and
```


Copyright© 2022 Ciena® Corporation

```
          either delivering the frame to its MAC Client or discarding
          the frame (see 5.2.3.1.1).";
        reference
         "[802.1ax] 6.3.1.1.32";
      }
      list member-ports {
        key "member-port";
        description
         "List of user configured ports on this Aggregator";

        leaf member-port {
          type if:interface-ref;
          description
           "Configured member ports of the Aggregator identified
            by their interface ID";
        }
        uses agg-port-parameters-group;
        leaf agg-port-protocol-DA {
          type yang:mac-address;
          description
           "A 6-octet read-write MACAddress value specifying
            the destination address to be used when sending Link
            Aggregation Control and Marker PDUs on this port,
            corresponding to the value of Protocol_DA
            in 5.2.7.1.2, 5.2.9.1.2 and 5.5.4.2.1.
            The default DA is the IEEE 802.3 Slow_Protocols_Multicast
            address.";
        reference
         "[802.1ax] 6.3.2.2.1";
        }
      }
    }
  }

  augment "/if:interfaces-state/if:interface" {
    when "if:type = 'ianaift:ieee8023adLag'" {
      description
       "Augment only if the interface type is ieee8023adLag.";
    }
    description
     "Augmenting LAG state to interfaces state.";

    container agg-state {
      description
       "Operational data for each Aggregator.";
      leaf agg-MAC-address {
        type yang:mac-address;
        description
         "A 6-octet read-only value carrying the individual
          MAC address assigned to the Aggregator.";
        reference
         "[802.1ax] 6.3.1.1.9";
      }

      leaf agg-actor-system-ID {
        type yang:mac-address;
        description
         "A 6-octet read-write MAC address value used as a unique
          identifier for the System that contains this Aggregator.
          NOTE-From the perspective of the Link Aggregation
```


Copyright© 2022 Ciena® Corporation

```
      mechanisms described in Clause 5, only a single
      combination of Actor's System ID and System Priority are
      considered, and no distinction is made between the
      values of these parameters for an Aggregator and the
      port(s) that are associated with it; i.e., the protocol
      is described in terms of the operation of aggregation
      within a single System. However, the managed objects
      provided for the Aggregator and the port both allow
      management of these parameters. The result of this is to
      permit a single piece of equipment to be configured by
      management to contain more than one System from the
      point of view of the operation of Link Aggregation. This
      may be of particular use in the configuration of
      equipment that has limited aggregation capability
      (see 5.6).";
    reference
     "[802.1ax] 6.3.1.1.4";
  }

  leaf agg-aggregate-or-individual {
    type boolean;
    description
     "A read-only Boolean value indicating whether the
      Aggregator represents an Aggregate ('TRUE') or
      an Individual link ('FALSE').";
    reference
     "[802.1ax] 6.3.1.1.6";
  }

  leaf agg-actor-oper-key {
    type lacp-key;
    description
     "The current operational value of the Key for the
      Aggregator. The administrative Key value may differ from
      the operational Key value for the reasons discussed in
      5.6.2. This is a 16-bit read-only value. The meaning of
      particular Key values is of local significance.";
    reference
     "[802.1ax] 6.3.1.1.8";
  }

  leaf agg-partner-system-ID {
    type yang:mac-address;
    description
     "A 6-octet read-only MAC address value consisting
      of the unique identifier for the current protocol Partner
      of this Aggregator. A value of zero indicates that there
      is no known Partner. If the aggregation is manually
      configured, this System ID value will be a value assigned
      by the local System.";
    reference
     "[802.1ax] 6.3.1.1.10";
  }

  leaf agg-partner-system-priority {
    type uint16;
    description
     "A 2-octet read-only value that indicates the priority
      value associated with the Partner's System ID. If the
      aggregation is manually configured, this System Priority
```


Copyright© 2022 Ciena® Corporation

```
          value will be a value assigned by the local System.";
        reference
         "[802.1ax] 6.3.1.1.11";
      }

      leaf agg-partner-oper-key {
        type lacp-key;
        description
         "The current operational value of the Key for the
          Aggregator's current protocol Partner. This is
          a 16-bit read-only value. If the aggregation is manually
          configured, this Key value will be a value assigned by the
          local System.";
        reference
         "[802.1ax] 6.3.1.1.12";
      }
      list distribution-ports {

        key "distribution-port";
        description
         "List of operationally active ports on the Aggregator";
        leaf distribution-port {
          type if:interface-ref;
          description
           "Member ports of the Aggregator that are in distribution
            identified by their interface ID";
        }
        uses agg-port-operational-group;
        uses agg-port-statistics-group;
        uses agg-port-debug-info-group;
      }
    }
  }
} /* end of module ciena-ieee-lag */
```


Copyright© 2022 Ciena® Corporation

ciena-ieee-lldp-deviation.yang

```
module ciena-ieee-lldp-deviation {

    namespace "urn:ciena:params:xml:ns:yang:ciena-pn::ciena-ieee-lldp-dev";
    prefix "ciena-ieee-lldp-dev";

    import ciena-ieee-lldp { prefix ieee-lldp; }

    import ietf-interfaces { prefix "if"; }

    revision "2015-03-14" {
        description "Initial revision.";
    }

    //
    //  Changes to supported config data.
    //

    //
    //  Config data that is not supported in SAOS 6.14
    //
    deviation /if:interfaces/if:interface/ciena-ieee-lldp:lldp-port-config/
ciena-ieee-lldp:management-address {
        deviate not-supported;
    }

    //
    //  Read-only data is not supported in SAOS 6.14
    //
}
```


Copyright© 2022 Ciena® Corporation

ciena-ieee-lldp.yang

```
module ciena-ieee-lldp {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn::ciena-ieee-lldp";
    prefix "lldp";

    import ietf-yang-types { prefix "yang"; }

    import iana-if-type { prefix "ianaift"; }

    import ietf-interfaces { prefix "if"; }

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
                 Hanover, Maryland 21076
                 U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
        "This YANG module contains the schema definition for Link Layer
        Discovery Protocol.";

    revision "2015-03-14" {
        description
            "Initial Version.";
        reference "ieee Std 802.1AB";
    }

    feature dot1-extension {
        description
            "This feature means that device supports dot1-extended tlvs";
    }

    feature dot3-extension {
        description
            "This feature means that device supports dot3-extended tlvs";
    }

    //This is IANA-Address-Family-Numbers definition will be
    // moved out of LLDP once we have the standard data type.
    typedef address-family {
        type enumeration {
            enum ipv4 {
                value 1;
                description "IP version 4";
            }
            enum ipv6 {
                value 2;
                description "IP version 6";
            }
            enum nsap {
                value 3;
                description "NSAP";
            }
```


Copyright© 2022 Ciena® Corporation

```
            enum hdlc {
                value 4;
                description "HDLC (8-bit multidrop)";
            }
            enum bbn-1822 {
                value 5;
                description "BBN 1822";
            }
            enum all-802 {
                value 6;
            description "802 (includes all 802 media plus Ethernet 'canonical
format')";
            }
            enum e-163 {
                value 7;
                description "E.163";
            }
            enum e-164 {
                value 8;
                description "E.164 (SMDS, FrameRelay, ATM)";
            }
            enum f-69 {
                value 9;
                description "F.69 (Telex)";
            }
            enum x-121 {
                value 10;
                description "X.121 (X.25, Frame Relay)";
            }
            enum ipx {
                value 11;
                description "IPX (Internetwork Packet Exchange)";
            }
            enum appletalk {
                value 12;
                description "Appletalk";
            }
            enum decnet-iv {
                value 13;
                description "DECnet IV";
            }
            enum banyan-vines {
                value 14;
                description "Banyan Vines";
            }
            enum e-164-with-nsap {
                value 15;
                description "E.164 with NSAP format subaddress";
                reference "ATM Forum UNI 3.1";
            }
            enum dns {
                value 16;
                description "DNS (Domain Name System)";
            }
            enum distinguished-name {
                value 17;
                description "Distinguished Name (per X.500)";
            }
            enum as-number {
                value 18;
```


Copyright© 2022 Ciena® Corporation

```
                description "Autonomous System Number";
            }
            enum xtp-over-ipv4 {
                value 19;
                description "XTP over IP version 4";
            }
            enum xtp-over-ipv6 {
                value 20;
                description "XTP over IP version 6";
            }
            enum xtp-native-mode-xtp {
                value 21;
                description "XTP native mode XTP";
            }
            enum fibre-channel-wwpn {
                value 22;
                description "Fibre Channel World-Wide Port Name";
            }
            enum fibre-channel-wwnn {
                value 23;
                description "Fibre Channel World-Wide Node Name";
            }
            enum gwid {
                value 24;
                description "Gateway Identifier";
            }
            enum l2vpn {
                value 25;
                description "AFI for L2VPN information";
                reference
                    "RFC 4761: Virtual Private LAN Service (VPLS): Using BGP
for Auto-Discovery
                   and Signaling RFC 6074: Provisioning, Auto-Discovery, and
Signaling
                     in Layer 2 Virtual Private Networks (L2VPNs)";
            }
            enum mpls-tp-section-endpoint-identifier {
                value 26;
                description "MPLS-TP Section Endpoint Identifier";
                reference "draft-ietf-mpls-gach-adv";
            }
            enum mpls-tp-lsp-endpoint-identifier {
                value 27;
                description "MPLS-TP LSP Endpoint Identifier";
                reference "draft-ietf-mpls-gach-adv";
            }
            enum mpls-tp-pseudowire-endpoint-identifier {
                value 28;
                description "MPLS-TP Pseudowire Endpoint Identifier";
                reference "draft-ietf-mpls-gach-adv";
            }
            enum eigrp-common-service-family {
                value 16384;
                description "EIGRP Common Service Family";
            }
            enum eigrp-ipv4-service-family {
                value 16385;
                description "EIGRP IPv4 Service Family";
            }
            enum eigrp-ipv6-service-family {
```


Copyright© 2022 Ciena® Corporation

```
                value 16386;
                description "EIGRP IPv6 Service Family";
            }
            enum lisp-canonical-address-format {
                value 16387;
                description "LISP Canonical Address Format (LCAF)";
            }
            enum bgp-ls {
                value 16388;
                description "BGP-LS";
                reference "draft-ietf-idr-ls-distribution";
            }
            enum 48-bit-mac {
                value 16389;
                description "48-bit MAC";
                reference "draft-eastlake-rfc5342bis";
            }
            enum 64-bit-mac {
                value 16390;
                description "64-bit MAC";
                reference "draft-eastlake-rfc5342bis";
            }
        }
        description
            "This typedef is a YANG enumeration of IANA-registered address
             family numbers (AFN).";
        reference
            "IANA Address Family Numbers registry. <http://www.iana.org/
assignments/address-family-numbers>";
    }

    typedef system-capability-map {
        type bits {
            bit other {
                position 0;
                description
                    "indicates that the system has capabilities other than
those listed
                     below.";
            }
            bit repeater {
                position 1;
                description
                    "indicates that the system has repeater capability.";
            }
            bit bridge {
                position 2;
                description
                    "indicates that the system has bridge capability.";
            }
            bit wlan-accesspoint {
                position 3;
                description
                "indicates that the system has WLAN access point capability.";
            }
            bit router {
                position 4;
                description
                    "indicates that the system has router capability.";
            }
```


Copyright© 2022 Ciena® Corporation

```
            bit telephone {
                position 5;
                description
                    "indicates that the system has telephone capability.";
            }
            bit docsis-cable-dev {
                position 6;
                description
                 "indicates that the system has DOCSIS Cable Device capability
                      (IETF RFC 2669 & 2670).";
            }
            bit station-only {
                position 7;
                description
                  "indicates that the system has only station capability and
nothing
                      else.";
            }
        }
        description
            "This describes the system capabilities.";
    }

    grouping man-address-index-group {
        description
            "Index definition for managemet address.";
        leaf man-address-subtype {
            type address-family;
            description
                "The type of managment address identifier encoded for the
managment
                 address.";
        }
        leaf man-address {
            type string {
            }
            description
                "The string value used to identify the management address
component
                 associated with the system.";
        }
    }

    grouping man-address-group {
        description
            "Describes the management address of a system.";
        leaf if-subtype {
            type enumeration {
                enum unknown {
                    value 1;
                    description
                     "Represents the case where the interface is not known.";
                }
                enum if-index {
                    value 2;
                    description
                       "Represents interface identifier based on the ifIndex
MIB object.";
                }
                enum sys-port-num {
```


Copyright© 2022 Ciena® Corporation

```
                    value 3;
                    description
                       "Represents interface identifier based on the system
port numbering
                        convention.";
                }
            }
            description
               "The enumeration value that identifies the interface numbering
               method used for defining the interface number, associated with
                 the system.";
        }
        leaf if-id {
            type int32;
            description
            "The integer value used to identify the interface number regarding
               the management address component associated with the system.";
        }
        leaf address-oid {
            type yang:object-identifier-128;
            description
               "The OID value used to identify the type of hardware component
                 or protocol entity associated with the management address
advertised
                 by the system agent.";
        }
    }

    grouping chassis-group {
        description
            "Describes the chassis information of a system.";
        leaf chassis-id-subtype {
            type enumeration {
                enum chassis-component {
                    description
                        "Represents a chassis identifier based on the value
of entPhysicalAlias
                         object (defined in IETF RFC 4133) for a chassis
component.";
                }
                enum interface-alias {
                    description
                        "Represents a chassis identifier based on the value
of ifAlias
                         object (defined in IETF RFC 2863) for an interface
on the containing
                         chassis.";
                }
                enum port-component {
                    description
                        "Represents a chassis identifier based on the value
of entPhysicalAlias
                         object (defined in IETF RFC 4133) for a port or
backplane component.";
                }
                enum mac-address {
                    description
                        "Represents a chassis identifier based on the value
of a unicast
                         source address.";
```


Copyright© 2022 Ciena® Corporation

```
                }
                enum network-address {
                    description
                        "Represents a chassis identifier based on a network
address, associated
                         with a particular chassis.";
                }
                enum interface-name {
                    description
                        "Represents a chassis identifier based on the value
of ifName
                         object (defined in IETF RFC 2863) for an interface
on the containing
                         chassis.";
                }
                enum local {
                    description
                        "represents a chassis identifier based on a locally
defined value.";
                }
            }
            description
                "Type of chassis id in a system.";
        }
        leaf chassis-id {
            type string {
            }
            description
                "The string value used to identify the chassis component
associated
                 with the system.";
        }
    }

    grouping system-group {
        description
            "Describes the system and its capabilities.";
        leaf system-name {
            type string {
            }
            description
                "The string value used to identify the system name of the
system.";
        }
        leaf system-description {
            type string {
            }
            description
              "The string value used to identify the system description of the
                 system.";
        }
        leaf system-capability-supported {
            type system-capability-map;
            description
              "The bitmap value used to identify which system capabilities are
                 supported on the system.";
        }
        leaf system-capability-enabled {
            type system-capability-map;
            description
```


Copyright© 2022 Ciena® Corporation

```
                "The bitmap value used to identify which system capabilities are
                   enabled on the system.";
            }
        }

        grouping port-group {
            description
                "Describes the port inforamtion of a system.";
            leaf port-id-subtype {
                type enumeration {
                    enum interface-alias {
                        description
                          "represents a port identifier based on the ifAlias MIB
object,
                             defined in IETF RFC 2863.";
                    }
                    enum port-component {
                        description
                            "represents a port identifier based on the value of
entPhysicalAlias
                             (defined in IETF RFC 4133) for a port component.";
                    }
                    enum mac-address {
                        description
                          "represents a port identifier based on a unicast source
address
                           (encoded in network byte order and IEEE 802.3 canonical
bit order).";
                    }
                    enum network-address {
                        description
                            "represents a port identifier based on a network
address, detected
                             by the agent and associated with a particular port.";
                    }
                    enum interface-name {
                        description
                           "represents a port identifier based on the ifName MIB
object,
                             defined in IETF RFC 2863.";
                    }
                    enum agent-circuit-id {
                        description
                          "represents a port identifier based on the agent-local
identifier
                             of the circuit (defined in RFC 3046).";
                    }
                    enum local {
                        description
                          "represents a port identifier based on a value locally
assigned.";
                    }
                }
                description
                    "Type of port id.";
            }
            leaf port-id {
                type string {
                }
                description
```


Copyright© 2022 Ciena® Corporation

```
                "The string value used to identify the port component associated
                   with the system.";
            }
            leaf port-desc {
                type string {
                }
                description
                  "The string value used to identify the description of the given
                     port associated with the system.";
            }
        }

        grouping dot3-mac-group {
            description
                "Describes the mac capabilities.";
            leaf auto-neg-supported {
                type boolean;
                description
                    "The truth value used to indicate whether the given port
(associated
                     with the system) supports Auto-negotiation.";
            }
            leaf auto-neg-enabled {
                type boolean;
                description
                  "The truth value used to indicate whether port Auto-negotiation
                     is enabled on the given port associated with the system.";
            }
            leaf auto-neg-advertised-cap {
                type binary {
                    length "2";
                }
                description
                    "This object contains the value (bitmap) (defined in IETF RFC
                    3636) which is associated with the given port on the system.";
            }
            leaf oper-mau-type {
                type int32 {
                    range "0..2147483647";
                }
                description
                  "An integer value that indicates the operational MAU type of the
                     given port on the system.";
            }
        }

        grouping dot3-power-group {
            description
                    "Describes the power capabilites.";
            leaf port-class {
                type enumeration {
                    enum p-class-pse {
                        description
                            "PSE class.";
                    }
                    enum p-class-pd {
                        description
                            "PD class.";
                    }
                }
```


Copyright© 2022 Ciena® Corporation

```
            description
                "The value that identifies the port Class of the given port
associated
                 with the system.";
        }
        leaf mdi-supported {
            type boolean;
            description
                "The truth value used to indicate whether the MDI power is
supported
                 on the given port associated with the system.";
        }
        leaf mdi-enabled {
            type boolean;
            description
               "The truth value used to identify whether MDI power is enabled
                 on the given port associated with the system.";
        }
        leaf pair-controlable {
            type boolean;
            description
                "The truth value is derived from the value of
pethPsePortPowerPairsControlAbility
              object (defined in IETF RFC 3621) and is used to indicate whether
              the pair selection can be controlled on the given port associated
                 with the system.";
        }
        leaf pairs {
            type int8 {
                range "1..2";
            }
            description
                "This object contains the value of the pethPsePortPowerPairs
object
                 (defined in IETF RFC 3621) which is associated with the given
                  port on the system.";
        }
        leaf class {
            type int8 {
                range "1..5";
            }
            description
                "This object contains the value of the
pethPsePortPowerClassifications
               object (defined in IETF RFC 3621) which is associated with the
                 given port on the system.";
        }
    }

    grouping dot3-link-agg-group {
        description
           "Link Aggregation information about a particular port component.";
        leaf agg-status {
            type bits {
                bit capable {
                    position 0;
                    description
                        "Specifies whether it is capable.";
                }
                bit enabled {
```


Copyright© 2022 Ciena® Corporation

```
                    position 1;
                    description
                        "Specifies if it is enabled.";
                }
            }
            description
              "The bitmap value contains the link aggregation capabilities and
                  the current aggregation status of the link.";
        }
        leaf agg-port-id {
            type union {
                type int32 {
                    range "1..2147483647";
                }
                type enumeration {
                    enum not-supported {
                        description
                            "Agg not supported.";
                    }
                }
            }
            description
              "This object contains the IEEE 802.3 aggregated port identifier,
                  aAggPortID (IEEE 802.3-2002, 30.7.2.1.1), derived from the
ifNumber
                  of the ifIndex for the port component in link aggregation.";
        }
    }

    grouping dot3-frame-size-group {
        description
            "Frame size information about a particular port component.";
        leaf max-frame-size {
            type int32 {
                range "0..65535";
            }
            description
                "An integer value indicating the maximum supported frame size
                  in octets on the given port of the system.";
        }
    }

    grouping dot1-port-vlan-group {
        description
            "Port vlan information about a particular port component.";
        leaf port-vlan-id {
            type union {
                type int32 {
                    range "1..4094";
                }
                type enumeration {
                    enum not-supported {
                        description
                            "Port vlan not supported.";
                    }
                }
            }
            description
                "The integer value used to identify the port's VLAN identifier
                  associated with the system.";
```


Copyright© 2022 Ciena® Corporation

```
        }
    }

    grouping dot1-proto-vlan-key-group {
        description
            "Describes the indices for protocol vlan.";
        leaf proto-vlan-id {
            type union {
                type int32 {
                    range "1..4094";
                }
                type enumeration {
                    enum not-supported {
                        description
                            "Protocol vlan not supported.";
                    }
                }
            }
            description
              "The integer value used to identify the port and protocol VLANs
               associated with the given port associated with the system.";
        }
    }

    grouping dot1-proto-vlan-group {
        description
            "Describes the protocol vlan.";
        uses dot1-proto-vlan-key-group;
        leaf proto-vlan-supported {
            type boolean;
            description
                "The truth value used to indicate whether the given port
(associated
                 with the system) supports port and protocol VLANs.";
        }
        leaf proto-vlan-enabled {
            type boolean;
            description
              "The truth value used to indicate whether the port and protocol
               VLANs are enabled on the given port associated with the system.";
        }
    }

    grouping dot1-vlan-name-key-group {
        description
            "Describes the indices for vlan names.";
        leaf vlan-id {
            type int16 {
                range "0..4094";
            }
            description
             "The integer value used to identify the IEEE 802.1Q VLAN IDs with
                 which the given port is compatible.";
        }
    }

    grouping dot1-vlan-name-group {
        description
            "Describes the Vlan name.";
        uses dot1-vlan-name-key-group;
```


Copyright© 2022 Ciena® Corporation

```
        leaf vlan-name {
            type string {
            }
            description
               "The string value used to identify VLAN name identified by the
                 Vlan Id associated with the given port on the system.";
        }
    }

    grouping dot1-protocol-key-group {
        description
            "Describes the protocol index.";
        leaf protocol-index {
            type int32 {
                range "1..2147483647";
            }
            description
                "This object represents an arbitrary local integer value used
                 by this agent to identify a particular protocol identity.";
        }
    }

    grouping dot1-protocol-group {
        description
            "Describes the protocols supported over the port.";
        uses dot1-protocol-key-group;
        leaf protocol-id {
            type binary {
            }
            description
             "The octet string value used to identify the protocols associated
                 with the given port of the system.";
        }
    }

    container lldp-global-config {
        description
            "Global LLDP variables that could be configured.";
        leaf admin-enabled {
            type boolean;
            description
                "Globally is LLDP administratively enabled or disabled.";
        }
        leaf msg-tx-interval {
            type uint16 {
                range "5..32768";
            }
            units seconds;
            default "30";
            description
              "The interval at which LLDP frames are transmitted on behalf of
                 this LLDP agent.";
        }
        leaf msg-tx-hold-multiplier {
            type uint8 {
                range "2..10";
            }
            default "4";
            description
```


Copyright© 2022 Ciena® Corporation

```
                "The time-to-live value expressed as a multiple of the msg-
tx-interval
                 object.";
        }
        leaf reinit-delay {
            type uint8 {
                range "1..10";
            }
            units seconds;
            default "2";
            description
               "Specifies the delay time in seconds for LLDP to re-initialize
                on any port. The range is 1 to 10 seconds; the default is 2
seconds.";
        }
        leaf tx-delay {
            type uint16 {
                range "1..8192";
            }
            units seconds;
            default "2";
            description
              "The tx-delay indicates the delay (in units of seconds) between
               successive LLDP frame transmissions initiated by value/status
                changes in the LLDP local systems.";
        }
        leaf notification-interval {
            type uint16 {
                range "5..3600";
            }
            units seconds;
            default "5";
            description
             "The interval in seconds within which only one notification could
                 be sent.";
        }
    }

    container global-stats {
        config false;
        description
            "global LLDP statistics that could be read.";
        leaf remote-last-change-time {
            type yang:date-and-time;
            description
              "systime when an entry is last created, modified or deleted for
                 any learned remote system data.";
        }
        leaf remote-inserts {
            type yang:zero-based-counter32;
            description
              "Counter to track the number of complete inserts or new learned
                 entry in the remote system data.";
        }
        leaf remote-deletes {
            type yang:zero-based-counter32;
            description
               "Counter to track the number of complete deletes or removal of
                 learned entry in the remote system data.";
        }
```


Copyright© 2022 Ciena® Corporation

```
        leaf remote-drops {
            type yang:zero-based-counter32;
            description
                "Counter to track the number of incomplete inserts or deletes
               because of insufficient resources in the remote system data.";
        }
        leaf remote-ageouts {
            type yang:zero-based-counter32;
            description
             "Counter to track the number of deletes in the remote system data
                 beacuse of information timeliness interval expiry.";
        }
    }

    container global-operational {
        config false;
        description
          "Global operational variables and list that is local to the system.";
        uses chassis-group;
        uses system-group;

        list local-managment-address {
            key "man-address-subtype man-address";
            description
               "the list contains management address information on the local
                  system known to this agent.";
            uses man-address-index-group;
            leaf man-address-len {
                type int32;
                description
                    "The total length of the management address subtype and
the management
                       address fields in LLDP PDUs transmitted by the local LLDP
agent.";
            }
            uses man-address-group;
        }
    }

    augment "/if:interfaces/if:interface" {
        when "if:type = 'ianaift:ethernetCsmacd'" {
            description
                "Augment only if the interface type is ethernetCSmacd.";
        }
        description
           "Augmenting ietf-interface module to add configuration parameters
for LLDP.";

        container lldp-port-config {
            description
                "per port configurable variables.";
            leaf admin-status {
                type enumeration {
                    enum tx-only {
                        description
                           "The LLDP agent transmits LLDPframes on this port
and it does
                            not store any information about the remote systems
connected.";
                    }
```


Copyright© 2022 Ciena® Corporation

```
                    enum rx-only {
                        description
                          "The LLDP agent receives, but it does not transmit
LLDP frames
                             on this port.";
                    }
                    enum tx-and-rx {
                        description
                          "The LLDP agent transmits and receives LLDP frames
on this port.";
                    }
                    enum disabled {
                        description
                           "The LLDP agent does not transmit or receive LLDP
frames on this
                             port.";
                    }
                }
                default tx-and-rx;
                description
                   "defines the LLDP protocol configurational status on each
port.";
            }
            leaf notify {
                type boolean;
                default "false";
                description
                    "defines the state of LLDP notification for each port.";
            }
            leaf send-port-desc-tlv {
                type boolean;
                default "false";
                description
                    "controls to transmission of port description tlv.";
            }
            leaf send-sys-name-tlv {
                type boolean;
                default "false";
                description
                    "controls to transmission of system name tlv.";
            }
            leaf send-sys-desc-tlv {
                type boolean;
                default "false";
                description
                    "controls to transmission of system description tlv.";
            }
            leaf send-sys-cap-tlv {
                type boolean;
                default "false";
                description
                    "controls to transmission of system capabilities tlv.";
            }

            list management-address {
                key "man-address-subtype man-address";
                description
                  "list of managment address tvls configured for each port.";
                uses man-address-index-group;
                leaf send-mgnt-address-tlv {
```


Copyright© 2022 Ciena® Corporation

```
                    type boolean;
                    default "false";
                    description
                        "controls to transmission of management address tlv
for each management
                         address.";
                }
            }
            leaf man-add-local-ipv4 {
                type boolean;
                default "false";
                description
                    "controls transmission of local interface ipv4 manament
address in tlv.";
            }
            leaf man-add-local-ipv6 {
                type boolean;
                default "false";
                description
                    "controls transmission of local interface ipv6 manament
address in tlv.";
            }
            leaf man-add-remote-ipv4 {
                type boolean;
                default "false";
                description
                   "controls transmission of remote interface ipv4 manament
address in tlv.";
            }
            leaf man-add-remote-ipv6 {
                type boolean;
                default "false";
                description
                   "controls transmission of remote interface ipv6 manament
address in tlv.";
            }
        }

        container dot1-port-config {
            if-feature dot1-extension;
            description
                "dot1 LLDP extension per port config.";
            leaf send-port-vlan-tlv {
                type boolean;
                default "false";
                description
                    "controls to transmission of dot1 port vlan tlv.";
            }
            list dot1-proto-vlan {
                key "proto-vlan-id";
                description
                    "list of protocol vlans.";
                uses dot1-proto-vlan-key-group;
                leaf send-proto-vlan-tlv {
                    type boolean;
                    default "false";
                    description
                      "controls to transmission of dot1 protocol vlan tlv.";
                }
            }
```


Copyright© 2022 Ciena® Corporation

```
            list dot1-vlan-name {
                key "vlan-id";
                description
                    "list of vlan names.";
                uses dot1-vlan-name-key-group;
                leaf send-vlan-name-tlv {
                    type boolean;
                    default "false";
                    description
                        "controls to transmission of dot1 vlan name tlv.";
                }
            }
            list dot1-protocol {
                key "protocol-index";
                description
                    "list of protocols.";
                uses dot1-protocol-key-group;
                leaf send-protocol-tlv {
                    type boolean;
                    default "false";
                    description
                        "controls to transmission of dot1 protocol tlv.";
                }
            }
        }

        container dot3-port-config {
            if-feature dot3-extension;
            description
                "dot3 LLDP extension per port config.";
            leaf send-mac-phy-status-tlv {
                type boolean;
                default "false";
                description
                    "controls to transmission of dot3 mac phy status tlv.";
            }
            leaf send-power-via-mdi-tlv {
                type boolean;
                default "false";
                description
                    "controls to transmission of dot3 power via mdi tlv.";
            }
            leaf send-link-aggregation-tlv {
                type boolean;
                default "false";
                description
                   "controls to transmission of dot3 link aggregation tlv.";
            }
            leaf send-max-frame-size-tlv {
                type boolean;
                default "false";
                description
                    "controls to transmission of dot3 max frame size tlv.";
            }
        }
    }

    augment "/if:interfaces-state/if:interface" {
        when "if:type = 'ianaift:ethernetCsmacd'" {
            description
```


Copyright© 2022 Ciena® Corporation

```
                "Augment only if the interface type is ethernetCSmacd.";
        }
        description
            "Augmenting ietf-interfaces-state module to add operational
parameters for LLDP.";

        container tx-stats {
            description
                "frame transmit statistics.";
            leaf total-frames {
                type yang:counter32;
                description
                   "the number of LLDP frames transmitted by this LLDP agent
on the
                     indicated port.";
            }
        }

        container rx-stats {
            description
                "frame receive statistics.";
            leaf total-discards {
                type yang:counter32;
                description
                 "the number of LLDP frames received by this LLDP agent on the
                     indicated port, and then discarded for any reason.";
            }
            leaf total-errors {
                type yang:counter32;
                description
                "the number of invalid LLDP frames received by this LLDP agent
                   on the indicated port, while this LLDP agent is enabled.";
            }
            leaf total-frames {
                type yang:counter32;
                description
               "the number of valid LLDP frames received by this LLDP agent on
                     the indicated port, while this LLDP agent is enabled.";
            }
            leaf total-tlv-discards {
                type yang:counter32;
                description
                "the number of LLDP TLVs discarded for any reason by this LLDP
                     agent on the indicated port.";
            }
            leaf total-tlv-unrecognized {
                type yang:counter32;
                description
              "the number of LLDP TLVs received on the given port that are not
                     recognized by this LLDP agent on the indicated port.";
            }
            leaf total-tlv-ageouts {
                type yang:zero-based-counter32;
                description
                    "the counter that represents the number of age-outs that
occurred
                     on a given port.";
            }
        }
```


Copyright© 2022 Ciena® Corporation

```
        container port-operational {
            description
                "this table contains per port information associated with the
                 local system.";
            uses port-group;
        }

        container dot1-port-operational {
            if-feature dot1-extension;
            description
                "dot1 LLDP extension per port operational.";
            uses dot1-port-vlan-group;
            list dot1-proto-vlan {
                key "proto-vlan-id";
                description
                    "Protocoal Vlan's supported on this port.";
                uses dot1-proto-vlan-group;
            }
            list dot1-vlan-name {
                key "vlan-id";
                description
                    "Vlan name's supported on this port.";
                uses dot1-vlan-name-group;
            }
            list dot1-protocol {
                key "protocol-index";
                description
                    "Protocols supported on this port.";
                uses dot1-protocol-group;
            }
        }

        container dot3-port-operational {
            if-feature dot3-extension;
            description
                "dot3 LLDP extension per port operational.";
            uses dot3-mac-group;
            uses dot3-power-group;
            uses dot3-link-agg-group;
            uses dot3-frame-size-group;
        }
    }

    container remote-ports {
        config false;
        description
            "per port varibales of the remote system known to this agent.";
        list remote-entry {
            key "time-mark port-number remote-index";
            description
                "list of remote systems discored.";
            leaf time-mark {
                type yang:timeticks;
                description
                "it is the time ticks from sysuptime at which the remote entry
                    was updated. based on TimeFilter to retrive only updated
entries.";
            }
            leaf port-number {
                type int32;
```


Copyright© 2022 Ciena® Corporation

```
                description
                "Each port contained in the chassis (that is known to the LLDP
                     agent) is uniquely identified by a port number.";
            }
            leaf remote-index {
                type int32 {
                    range "1..2147483647";
                }
                description
                 "This object represents an arbitrary local integer value used
                  by this agent to identify a particular connection instance,
unique
                      only for the indicated remote system.";
            }

            uses chassis-group;
            uses port-group;
            uses system-group;

            list managment-address {
                key "man-address-subtype man-address";
                description
                   "list of management address tlvs received from the remote
system.";
                uses man-address-index-group;
                uses man-address-group;
            }

            list unknown-tlv {
                key "tlv-type";
                description
                    "list of unknown tlvs received from the remote system.";
                leaf tlv-type {
                    type uint16 {
                        range "9..126";
                    }
                    description
                       "This object represents the value extracted from the
type field
                            of the TLV.";
                }
                leaf tlv-info {
                    type binary {
                    }
                    description
                       "This object represents the value extracted from the
type field
                            of the TLV.";
                }
            }

            list org-def-info {
                key "org-id sub-type org-index";
                description
                 "organizationally defined information that is not recognized
by
                      the local agent.";
                leaf org-id {
                    type binary {
                        length "3";
```

```
                    }
                    description
                        "The Organizationally Unique Identifier (OUI), as
defined in IEEE
                         std 802-2001, is a 24 bit (three octets) globally
unique assigned
                         number referenced by various standards.";
                }
                leaf sub-type {
                    type uint8 {
                        range "1..255";
                    }
                    description
                      "The integer value used to identify the subtype of the
organizationally
                       defined information received from the remote system.";
                }
                leaf org-index {
                    type int32 {
                        range "1..2147483647";
                    }
                    description
                        "This object represents an arbitrary local integer
value used
                         by this agent to identify a particular unrecognized
organizationally
                         defined information instance.";
                }
                leaf org-info {
                    type binary {
                    }
                    description
                     "The string value used to identify the organizationally
defined
                         information of the remote system.";
                }
            }

            container dot1-remote-entry {
                if-feature dot1-extension;
                config false;
                description
                    "dot1 LLDP extension per remote entry operational.";
                uses dot1-port-vlan-group;
                list dot1-proto-vlan {
                    key "proto-vlan-id";
                    description
                        "Protocol Vlan's of specific remote entry.";
                    uses dot1-proto-vlan-group;
                }
                list dot1-vlan-name {
                    key "vlan-id";
                    description
                        "Vlan name's of specific remote entry.";
                    uses dot1-vlan-name-group;
                }
                list dot1-protocol {
                    key "protocol-index";
                    description
                        "Protocol list of specific remote entry.";
```


Copyright© 2022 Ciena® Corporation

```
                    uses dot1-protocol-group;
                }
            }

            container dot3-remote-entry {
                if-feature dot3-extension;
                config false;
                description
                    "dot3 LLDP extension per remote entry operational.";
                uses dot3-mac-group;
                uses dot3-power-group;
                uses dot3-link-agg-group;
                uses dot3-frame-size-group;
            }
        }
    }

    notification remote-entry-change {
        description
           "A notification that is sent when the global-stats or remote-last-
change-time
            changes.";
        container alarm {
            description
                "Data definitions related to a last remote change.";

            leaf remote-inserts {
                type yang:zero-based-counter32;
                description
               "Counter to track the number of complete inserts or new learned
                    entry in the remote system data.";
            }
            leaf remote-deletes {
                type yang:zero-based-counter32;
                description
                "Counter to track the number of complete deletes or removal of
                    learned entry in the remote system data.";
            }
            leaf remote-drops {
                type yang:zero-based-counter32;
                description
                 "Counter to track the number of incomplete inserts or deletes
                because of insufficient resources in the remote system data.";
            }
            leaf remote-ageouts {
                type yang:zero-based-counter32;
                description
                    "Counter to track the number of deletes in the remote
system data
                     beacuse of information timeliness interval expiry.";
            }
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-ieee-mstp.yang

```
module ciena-ieee-mstp {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-ieee-mstp";
    prefix "mstp";

    import iana-if-type {
        prefix "ianaift";
    }

    import ietf-interfaces {
        prefix "if";
    }

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
                 Hanover, Maryland 21076
                 U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
        "This YANG module contains the schema
         definition for MSTP.";

    revision "2016-10-25" {
        description
            "Added mstp-port-config for lag interfaces.";
    }

    revision "2015-03-14" {
        description
            "Initial Version.";
        reference "IEEE Std 802.1W";
    }

    container mstp-global-config {
        description
            "Global MSTP configuration settings.";
        leaf admin-enabled {
            type boolean;
            description
                "Global enable/disable for MSTP.";
        }
    }

    augment "/if:interfaces/if:interface" {
        when "if:type = 'ianaift:ethernetCsmacd' or
              if:type = 'ianaift:ieee8023adLag'" {
            description
                "Augment only if the interface type
                 is ethernetCSmacd or ieee8023adLag.";
        }
        description
            "Augmenting ietf-interface module to add
```


Copyright© 2022 Ciena® Corporation

```
                configuration parameters for MSTP.";

            container mstp-port-config {
                description
                    "per port configurable variables.";
                leaf admin-enabled {
                    type boolean;
                    description
                      "Per-port enable/disable for MSTP.";
                }
            }
        }
    }
```


Copyright© 2022 Ciena® Corporation

ciena-ieee-port-dot1x-deviation.yang

```
module ciena-ieee-port-dot1x-deviation {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ieee-port-dot1x-dev";
    prefix "ciena-port-dot1x-dev";

    import ciena-ieee-port-dot1x {
        prefix "ciena-port-dot1x";
    }

    import ietf-interfaces {
        prefix "if";
    }

    revision 2015-07-21 {
        description "Initial revision";
    }

    deviation /if:interfaces/if:interface/ciena-port-dot1x:authenticator/
ciena-port-dot1x:retry-max {
        deviate replace {
            type uint8 {
                range "1..10";
            }
        }
    }
    deviation /if:interfaces/if:interface/ciena-port-dot1x:authenticator/
ciena-port-dot1x:quiet-period {
        deviate replace {
            type uint16 {
                range "1..65535";
            }
        }
    }
    deviation /if:interfaces/if:interface/ciena-port-dot1x:authenticator/
ciena-port-dot1x:reauth-period {
        deviate replace {
            type uint16 {
                range "1..65535";
            }
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-ieee-port-dot1x.yang

```
module ciena-ieee-port-dot1x {
   namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-ieee:ns:yang:ieee-
port-dot1x";
    prefix "port-dot1x";

    import ietf-interfaces {
        prefix "if";
    }

    import ietf-yang-types {
        prefix "yang";
    }

    import iana-if-type {
        prefix "ianaift";
    }

    import ciena-pkix-types {
        prefix pkix-types;
    }

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
                 Hanover, Maryland 21076
                 U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
      "This YANG module contains the schema definition for 802.1x, port
        level security.";

    revision "2019-06-20" {
       description
          "Updated the default value of reauth-enabled leaf.";
       reference
          "IEEE Std 802.1X-2004";
    }

    revision "2017-10-27" {
       description
          "Updated with new Common Criteria features.";
       reference
          "";
    }

    revision "2015-01-20" {
      description
        "Initial Version.";
      reference
        "Add the reference here.";
    }
```


Copyright© 2022 Ciena® Corporation

```
    augment "/if:interfaces/if:interface" {
        when "if:type = 'ianaift:ethernetCsmacd'";

        container authenticator {
        leaf admin-enabled {
            type boolean;
                default "false";
            description
                "Is 802.1x administratively enabled or disabled.";
       }

        leaf eapol-version {
            type uint8 {
                range "1..2";
            }
            default "2";
            description
              "Specifies the eapol-version - 1 or 2 (2 is the default).
                    This value is set to 1 to work around issues with client
                  implementations that drop EAPOL frames that use version 2";
        }

            leaf mode {
                type enumeration {
                    enum auto {
                        description
                            "Port is authorized or unauthorized based on the
                              outcome of the authentication process on the port";
                    }
                    enum force-auth {
                        description
                            "Port is forcibly marked as authorized and will
                            allow supplicant access";
                    }
                    enum force-unauth {
                        description
                           "Port is forcibly marked as unauthorized and will
                              unconditional block access to the supplicant.";
                    }
                }
                default auto;
                description
                    "Sets the port authorization mode";
            }

        leaf retry-max {
            type uint8;
                default "2";
            description
                "Specifies the maximum number of re-authentication attempts
on an
                  authenticator port before port is unauthorized.";
        }

        leaf reauth-enabled {
            type boolean;
                default "false";
            description
                "This object is set true if PACP should initiate
                reauthentication periodically, false otherwise";
```


Copyright© 2022 Ciena® Corporation

```
        }

        leaf quiet-period {
            type uint16;
            default "60";
            description
                "Number of seconds that the switch remains in the
                 quiet state following a failed authentication exchange with
                 the client";
            reference
                "IEEE 802.1X Clause 8.6, Figure 12-3";
        }

        leaf reauth-period {
            type uint16;
            default "3600";
            description
                "This object indicates the time period of the
                 reauthentication to the supplicant.";
            reference
                "IEEE 802.1X Clause 8.6, Figure 12-3";
        }

        leaf server-timeout {
        type uint8 {
          range "1..180";
        }
        default "30";
        description
          "Specifies the timeout value for an authenticator on a port
connecting
           to a backend authentication server..";
        }
    }

    container supplicant {
      leaf admin-state {
          type boolean;
                default "false";
          description
             "Enables or disables the port in supplicant role.

                 Enabling sets a port in a Supplicant role.

                 Disabling removes a port from the Supplicant role.  The
running total
                 supplicant statistics for this port are not cleared by this
operation,
                 and will continue to accumulate when the supplicant role on
this port
                 is re-enabled.";
      }
      leaf identity {
          type string;
          description
             "Specifies the username that the supplicant will use as the
identity
              in the authentication process.";
      }
      leaf password {
```


Copyright© 2022 Ciena® Corporation

```
          type string;
          description
             "Specifies the password that the supplicant will use for EAP-MD5
in
              the authentication process. Due to security reasons password
                    will be displayed as ****";
      }
      leaf eapol-version {
            type uint8 {
               range "1..2";
           }
            default "2";
            description
              "Specifies the eapol-version - 1 or 2 (2 is the default).
                   This value is set to 1 to work around issues with client
                  implementations that drop EAPOL frames that use version 2";
        }
        leaf auth-period {
        type uint16 {
          range "1..65535";
        }
                default "30";
        description
          "Specifies the amount of time a supplicant will wait for a (EAP)
           request from an authenticator before timing out.";
        }
        leaf held-period {
        type uint16 {
          range "1..65535";
        }
                default "60";
        description
          "Specifies the held period for a port in the supplicant role.
           The held period is the time period for which the supplicant stops
           trying to authenticate itself to the authenticator after an
           authentication failure.";
        }
        leaf start-period {
        type uint16 {
          range "1..65535";
        }
                default "30";
        description
          "Specifies the amount of time a supplicant will wait for a response
           from an authenticator on a connection request. If the authenticator
           has not responded after this time interval, the supplicant assumes
           that there is no valid authenticator on the other side.";
        }
        leaf max-start {
        type uint16 {
          range "1..65535";
        }
                default "3";
        description
          "Specifies the maximum number of times a supplicant will try
connecting
           to an authenticator before concluding that there is no
authenticator
           present.";
        }
```


Copyright© 2022 Ciena® Corporation

```
          leaf check-cert-time {
            type boolean;
            description
                "Turning this off forces supplicants doing mutual
authentication
                 to ignore certificate validity dates while authenticating the
server's
                 certificate.";
        }
          leaf mutual-auth {
              type boolean;
              description
                 "Causes the supplicant to authenticate or to not authenticate
the server using eap-tls.
                  Turning mutual authentication on requires CA certificates.
                  These certificates must already be installed.";
          }
          leaf eap-method {
              type enumeration {
               enum eap-md5;
               enum eap-tls;
               enum eap-md5-and-tls;
             }
             description
                 "Selects the EAP-Method(s). eap-md5 or eap-tls, or eap-md5
and eap-tls.
                  Using eap-tls requires a supplicant device certificate.
                  This certificate must already be installed.";
          }

      leaf cert-name {
          type string;
          description
             "Specifies the name of the certificate to use for dot1x auth
                     on this port.";
      }

      leaf ocsp {
          type boolean;
          description
             "Enable or disable using OCSP with certificate checking for this
                     port";
      }

      leaf peer-cert-reauth {
          type boolean;
          description
             "Enable or disable performing reauthentication of certificates
                     used for dot1x connections.";
      }

      leaf minimum-tls-version {
          type pkix-types:min-tls-version;
          description
             "The minimum TLS version supported for dot1x auth.";
      }
    }
    }

    augment "/if:interfaces-state/if:interface" {
```


Copyright© 2022 Ciena® Corporation

```
        when "if:type = 'ianaift:ethernetCsmacd'";

        leaf operational-status {
            type enumeration {
                enum notauth {
                    description
                        "Not authorized.";
                }
                enum lbtx;
                enum lbrx;
            }
        }
        container authentication-per-port {
        leaf operational-status {
        type boolean;
        description
          "Indicates the current operational state of the port as an
authenticator.

                This considers all config that might impact this including per
port
                supplicant configuration, over all dot1x admin state,
presence of
                required licenses, etc.";
        }

        leaf authenticate {
            type boolean;
            description
                "This object will be set true by the PAE authenticator to
                request authentication, and if this object is true,
                reauthentication is allowed.

                This object will be false while the PAE authenticator revokes
                authentication.";
        }

        leaf authenticated {
            type boolean;
            description
                "This object will be set true by PACP if the PAE authenticator
                 currently authenticated, and false if the authentication
                 fails or is revoked.";
        }

        leaf retry-count {
            type uint64;
            description
                "The count of the number of authentication attempts.";
            reference
                "IEEE 802.1X Clause 8.9";
        }

        leaf eapol-tx {
          type yang:counter64;
          description
             "The number of EAPOL frames of any type that have been
transmitted by this
              Authenticator";
      }
```


Copyright© 2022 Ciena® Corporation

```
      leaf eapol-rx {
          type yang:counter64;
          description
             "The number of EAPOL frames of any type that have been received
by this
              Authenticator";
      }
      leaf eap-req-id-rx {
          type yang:counter32;
          description
             "The number of EAP Req/Id frames that have been received by this
              Authenticator";
      }
      leaf eap-req-id-tx {
          type yang:counter32;
          description
             "The number of EAP Req/Id frames that have been transmitted by
this
              Authenticator";
      }
      leaf eap-req-rx {
          type yang:counter32;
          description
             "The number of EAP Request (non-Req/Id) frames that have been
received
              by this Supplicant";
      }
      leaf eap-res-tx {
          type yang:counter32;
          description
             "The number of EAP Responses (non-Req/Id) frames that have been
              transmitted by this Supplicant";
      }
    }
    container supplicant {
      leaf operational-status {
          type boolean;
      }

      leaf eapol-tx {
          type yang:counter64;
          description
             "The number of EAPOL frames of any type that have been
transmitted by this
              Supplicant";
      }
      leaf eapol-rx {
          type yang:counter64;
          description
             "The number of EAPOL frames of any type that have been received
by this
              Supplicant";
      }
      leaf eap-req-id-rx {
          type yang:counter32;
          description
             "The number of EAP Req/Id frames that have been received by this
              Supplicant";
      }
      leaf eap-req-id-tx {
```


Copyright© 2022 Ciena® Corporation

```
          type yang:counter32;
          description
             "The number of EAP Req/Id frames that have been transmitted by
this
              Supplicant";
      }
      leaf eap-req-rx {
          type yang:counter32;
          description
             "The number of EAP Request (non-Req/Id) frames that have been
received
              by this Supplicant";
      }
      leaf eap-res-tx {
          type yang:counter32;
          description
             "The number of EAP Responses (non-Req/Id) frames that have been
              transmitted by this Supplicant";
      }
    }

    container authenticator-session {
      leaf frame-tx {
          type yang:counter32;
          description
             "The number of user data frames transmitted on this Port during
the
              session.";
      }
      leaf frame-rx {
          type yang:counter32;
          description
             "The number of user data frames received on this Port during the
              session.";
      }
      leaf bytes-tx {
          type yang:counter64;
          description
             "The number of octets transmitted in user data frames on this
Port
              during the session.";
      }
      leaf bytes-rx {
          type yang:counter64;
          description
             "The number of octets received in user data frames on this Port
              during the session.";
      }
      leaf session-id {
          type string;
          description
             "A unique identifier for the session, in the form of a printable
              ASCII string of at least three characters.";
      }
      leaf session-time {
          type uint32;
          description
             "The duration of the session in seconds";
      }
      leaf terminate-cause {
```


Copyright© 2022 Ciena® Corporation

```
          type enumeration {
             enum supplicant-logo-off;
             enum port-failure;
             enum supplicant-restart;
             enum reauth-failed;
             enum auth-control-force-unauth;
             enum port-reliant;
             enum port-admin-disabled;
             enum not-terminated-yet;
          }
       }
       leaf identity {
          type string;
          description
             "The User-Name representing the identity of the Supplicant PAE";
       }
     }
     }
}
```


Copyright© 2022 Ciena® Corporation

ciena-ieee-port-eoam-deviation.yang

```
module ciena-ieee-port-eoam-deviation {

    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ieee-port-eoam-dev";
    prefix "ciena-ieee-port-eoam-dev";

    import ciena-ieee-port-eoam { prefix ciena-ieee-port-eoam; }

     import ietf-interfaces {
        prefix "if";
    }

    import ietf-yang-types {
        prefix "yang";
    }

    import iana-if-type {
          prefix "ianaift";
    }


    revision 2015-04-02 {
        description "Initial revision.";
    }

    deviation /if:interfaces/if:interface/ciena-ieee-port-eoam:eoam-port-
config/ciena-ieee-port-eoam:pdu-timer {
        deviate replace {
            type uint16 {
                range "100..1000";
            }
        }
        deviate add {
            default "1000";
        }
    }

    deviation /if:interfaces/if:interface/ciena-ieee-port-eoam:eoam-port-
config/ciena-ieee-port-eoam:link-lost-timer {
        deviate replace {
            type uint16 {
                range "500..5000";
            }
        }
        deviate add {
            default "5000";
        }
    }

    deviation /if:interfaces/if:interface/ciena-ieee-port-eoam:port-event-
config/ciena-ieee-port-eoam:error-frame-window {
        deviate replace {
            type uint32 {
                range "10..600";
            }
        }
    }
```


Copyright© 2022 Ciena® Corporation

```
    deviation /if:interfaces/if:interface/ciena-ieee-port-eoam:port-event-
config/ciena-ieee-port-eoam:error-frame-period-threshold {
        deviate replace {
            type uint32 {
                range "0..4294967293";
            }
        }
    }

    deviation /if:interfaces/if:interface/ciena-ieee-port-eoam:port-event-
config/ciena-ieee-port-eoam:error-frame-threshold {
        deviate replace {
            type uint32 {
                range "0..4294967293";
            }
        }
    }

    deviation /if:interfaces/if:interface/ciena-ieee-port-eoam:port-event-
config/ciena-ieee-port-eoam:error-frame-seconds-summary-threshold {
        deviate replace {
            type uint32 {
                range "0..65535";
            }
        }
    }

}
```


Copyright© 2022 Ciena® Corporation

ciena-ieee-port-eoam.yang

```
module ciena-ieee-port-eoam  {
   namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-ieee:ns:yang:ieee-
port-eoam";
       prefix "eoam";


    import ietf-interfaces {
        prefix "if";
    }

    import ietf-yang-types {
        prefix "yang";
    }

    import iana-if-type {
          prefix "ianaift";
    }

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
                 Hanover, Maryland 21076
                 U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
        "This YANG module contains the schema definition for Ethernet
        Operations, Administration and Management (EOAM) at a port level.";

    revision "2015-02-25" {
        description
                "Initial Version.";
        reference
                "[802.3ah] refers to:
                 IEEE Std 802.3ah-2004: 'Draft amendment to -
                 Information technology - Telecommunications and
                 information exchange between systems - Local and
                 metropolitan are networks - Specific requirements - Part
                 3: Carrier sense multiple access with collision detection
                 (CSMA/CD) access method and physical layer specifications
                 - Media Access Control Parameters, Physical Layers and
                 Management Parameters for subscriber access networks',
                 October 2004.

               [802-2001] refers to:
                 'IEEE Standard for LAN/MAN (Local Area
                 Network/Metropolitan Area Network): Overview and
                 Architecture', IEEE 802, June 2001.";
    }

    /*
     * typedefs.
     */
```


Copyright© 2022 Ciena® Corporation

```
    typedef oui {
        type string;
        description
            "An Organizationally Unique Identifier (OUI) is a 24-bit number
            that uniquely identifies a vendor, manufacturer, or other
organization
            globally or worldwide.";
    }


    typedef functions-supported {
        type bits {
            bit unidirectional-support {
                position 0;
                description
                  "Setting 'unidirectionalSupport(0)' indicates that the OAM
                 entity supports the transmission of OAMPDUs on links that are
                     operating in unidirectional mode (traffic flowing in one
                      direction only).  ";
            }
            bit loopback-support {
                position 1;
                description
                    "Setting 'loopbackSupport(1)' indicates that
                     the OAM entity can initiate and respond to loopback
commands.";
            }
            bit event-support {
                position 2;
                description
                 "Setting 'eventSupport(2)' indicates that the OAM entity can
                     send and receive Event Notification OAMPDUs. ";
            }
            bit variable-support {
                position 3;
                description
                 "Setting 'variableSupport(3)' indicates that the OAM entity
can send
                     and receive Variable Request and Response OAMPDUs.";
            }
        }
        description
            "The OAM functions supported on this Ethernet-like interface.
             OAM consists of separate functional sets beyond the basic
             discovery process that is always required.  These functional
             groups can be supported independently by any implementation.
             These values are communicated to the peer via the local
             configuration field of Information OAMPDUs.";
    }

    typedef event {

        type uint32;

        description
               "The type of event that generated this entry in the event log.
               When the OUI is the IEEE 802.3 OUI of 0x0180C2, the following
                event types are defined:
                    erroredSymbolEvent(1),
                    erroredFramePeriodEvent(2),
```


Copyright© 2022 Ciena® Corporation

```
                    erroredFrameEvent(3),
                    erroredFrameSecondsEvent(4),
                    linkFault(256),
                    dyingGaspEvent(257),
                    criticalLinkEvent(258)
                The first four are considered threshold crossing events as they
                  are generated when a metric exceeds a given value within a
specified
                  window. The other three are not threshold crossing events.
When the OUI is
                not 0x0180C2, then some other organization has defined the event
                space. If event subtyping is known to the implementation, it may
                  be reflected here. Otherwise, this value should return all Fs
                   (0xFFFFFFFF).";

        reference
                "[802.3ah], 30.3.6.1.10 and 57.5.3.";
    }

    typedef event-location {

        type enumeration {
            enum local {
                description
                    "This event occured locally.";
            }
            enum remote {
                 description
                "This event was received from the OAM peer via Ethernet OAM.";
            }
        }
        description
            "Indicates whether the event occured locally or was received
             from the OAM peer.";
    }

    typedef oper-status {

        type enumeration {
            enum disabled {
                description
                 "This value indicates that OAM is disabled on this interface
via
                    the admin-state";
            }
            enum link-fault {
                description
                 "If the link has detected a fault and is transmitting OAMPDUs
                    with a link fault indication, the value is link-fault";
            }
            enum passive-wait {
                description
                   "The passive-wait state is returned only by OAM entities
in passive
                 mode (mode) and reflects the state in which the OAM entity is
                    waiting to see if the peer device is OAM capable.";
            }
            enum active-send-local {
                description
```


Copyright© 2022 Ciena® Corporation

```
                    "The active-send-local is used by active mode devices
(mode) and
                reflects the OAM entity actively trying to discover whether the
                    peer has OAM capability but has not yet made that
determination.";
            }
            enum send-local-remote {
                description
                   "The state send-local-remote indicates that the local OAM
entity
                 has discovered the peer but has not yet accepted or rejected
the
                    configuration of the peer.";
            }
            enum send-local-remote-ok {
                description
                 "If the OAM peering is allowed by the local device, the state
                    moves to send-local-remote-ok.";
            }
            enum locally-rejected {
                description
                    "The local device can, for whatever reason, decide that
the peer
                    device is unacceptable and decline OAM peering. If the
local OAM
                    entity rejects the peer OAM entity, the state becomes
locally-rejected.
                   Note that both the send-local-remote and locally-rejected
states
                fall within the state SEND_LOCAL_REMOTE of the Discovery state
                   diagram [802.3ah, Figure 57-5], with the difference being
whether
                 the local OAM client has actively rejected the peering or has
                just not indicated any decision yet. Whether a peering decision
                    has been made is indicated via the local flags field in
the OAMPDU
                   (reflected in the aOAMLocalFlagsField of 30.3.6.1.10). ";
            }
            enum remotely-rejected {
                description
                    "If the remote OAM entity rejects the peering, the state
becomes
                    oamPeeringRemotelyRejected(8). Note that both the
sendLocalAndRemoteOk(6)
                and oamPeeringRemotelyRejected(8) states fall within the state
                 SEND_LOCAL_REMOTE_OK of the Discovery state diagram [802.3ah,
                 Figure 57-5], with the difference being whether the remote OAM
                  client has rejected the peering or has just not yet decided.
This
                    is indicated via the remote flags field in the OAM PDU
(reflected
                    in the aOAMRemoteFlagsField of 30.3.6.1.11).";
            }
            enum operational {
                description
                 "When the local OAM entity learns that both it and the remote
                   OAM entity have accepted the peering, the state moves to
operational(9)
                  corresponding to the SEND_ANY state of the Discovery state
diagram
```


Copyright© 2022 Ciena® Corporation

```
                    [802.3ah, Figure 57-5].";
            }
        }
        description
             "At initialization and failure conditions, two OAM entities on
            the same full-duplex Ethernet link begin a discovery phase to
            determine what OAM capabilities may be used on that link. The
          progress of this initialization is controlled by the OAM sub layer.";
    }

    typedef peer-status {

        type enumeration {
            enum active {
                value 1;
                description
                    "When active(1), the information is valid and represents
the current
                     peer of the OAM entity.";
            }
            enum inactive {
                value 2;
                description
                  "When inactive(2), the information in this row is invalid.
A value
                  of inactive(2) is returned if the oper-status is disabled,
passive-wait,
                  or active-send-local. For all other values of oper-status,
a value
                    of active(1) is returned.";
            }
        }
        description
            "This object indicates whether the information in this row should
            be considered valid.";
    }

    grouping port-config-objects {

        leaf pdu-timer {
            type uint16;
            description
                "The time in milliseconds between information OAMPDUs being
transmitted
                via the Ethernet port";
        }

        leaf link-lost-timer {
            type uint16;
            description
               "The time in milliseconds to wait for an information OAMPDU to
              be received from a peer via the Ethernet port, before declaring
                the link as lost";
        }
        description
            "OAM port configuration objects.";
    }

    grouping loopback-statistics {
```


Copyright© 2022 Ciena® Corporation

```
        leaf loopback-control-transmitted {
            type yang:counter32;
            units frames;
            description
               "A count of the number of Loopback Control OAMPDUs transmitted
               on this interface. A Loopback Control OAMPDU is indicated by a
                valid frame with (1) destination MAC address equal to that of
                 the reserved MAC address for Slow Protocols (See 43B of
[802.3ah]),
                 (2) a lengthOrType field equal to the reserved type for Slow
Protocols,
             (3) a Slow Protocols subtype equal to that of the subtype reserved
                for OAM, and (4) an OAMPDU code that equals the OAM Loopback
Control
                 code. Discontinuities of this counter can occur at re-
initialization
              of the management system, and at other times as indicated by the
                 value of the ifCounterDiscontinuityTime.";

            reference
                "[802.3ah], 30.3.6.1.26.";
        }

        leaf loopback-control-received {
            type yang:counter32;
            units frames;
            description
               "A count of the number of Loopback Control OAMPDUs received on
             this interface. A Loopback Control OAMPDU is indicated by a valid
                 frame with (1) destination MAC address equal to that of the
reserved
                 MAC address for Slow Protocols (See 43B of [802.3ah]), (2) a
lengthOrType
              field equal to the reserved type for Slow Protocols, (3) a Slow
             Protocols subtype equal to that of the subtype reserved for OAM,
             and (4) an OAMPDU code that equals the OAM Loopback Control code.
              Discontinuities of this counter can occur at re-initialization
             of the management system, and at other times as indicated by the
              value of the ifCounterDiscontinuityTime.";

            reference
                 "[802.3ah], 30.3.6.1.27.";
        }
        description
            "Loopback transmit and receive statistics.";
    }

    augment "/if:interfaces/if:interface" {
        when "if:type = 'ianaift:ethernetCsmacd'";

        container eoam-port-config {

            leaf admin-status {
                type enumeration {
                    enum enable {
                        description
                      "Enable OAM administratively on this port or interface";
                    }
                    enum disable {
                        description
```


Copyright© 2022 Ciena® Corporation

```
                        "Disable OAM administratively on this port or interface.
                             The OAM admin-status always starts in the disabled
state until
                              an explicit management action or configuration
information is
                             retained by the system and causes a transition to
the enabled
                              state";
                    }
                }
                default disable;
                description
                    "EOAM port admin state.";
            }

            leaf mode {
                type enumeration {
                    enum active {
                        description
                            "Active mode provides additional capabilities to
initiate
                          monitoring activities with the remote OAM peer entity.
                              As an example, an active OAM entity can put the
                              remote OAM entity in a loopback state, where a
passive OAM entity
                             cannot. The default value of mode is dependent on
the type of
                              system on which this Ethernet like interface
resides. The default
                             value should be 'active' unless it is known that
this system should
                              take on a subservient role to the other device
connected over
                              this interface. Changing this value results in
incrementing the
                            configuration revision field of locally generated
OAMPDUs (30.3.6.1.12)
                            and potentially causes re-doing of the OAM discovery
process if
                              the oper-status was already operational.";
                    }
                    enum passive {
                        description
                            "Passive mode generally waits for the peer to
initiate OAM
                             actions with it.";
                    }
                }
                default active;
                description
                "The mode of OAM opertion for this Ethernet like interface. OAM
                     on Ethernet interfaces maybe in active mode or passive
mode.";
            }

            uses port-config-objects;
            description
                "EOAM port state configuration.";

        } // End container eoam-port-config
```


Copyright© 2022 Ciena® Corporation

```
        container loopback-port-config {

            leaf loopback-ignore-rx {

                type enumeration {
                    enum ignore {
                        description
                            "When the value is ignore, received loopback
commands are ignored.
                             The default value is to ignore loopback commands
(ignore).";
                    }
                    enum process {
                        description
                           "When the value is process, OAM loopback commands
are processed.";
                    }
                }
                default ignore;
                description
                  "Since OAM loopback is a disruptive operation (user traffic
does
                   not pass), this attribute provides a mechanism to provide
controls
                   over whether received OAM loopback commands are processed
or ignored.
                    The attribute has no meaning if the local OAM entity does
not support
                   the loopback function (as defined in functions-supported).";
            }
            description
                "EOAM port loopback confguration.";
        } // End container loopback-port-config

        container port-event-config {

            leaf error-frame-period-window {
                type uint32;
                units frames;
                default 1488096;
                description
                "The number of frames over which the threshold is defined. The
                   default value of the window is the number of minimum size
Ethernet
                  frames that can be received over the physical layer in one
second.
                   If error-frame-period-threshold frame errors occur within
a window
                  of error-frame-period-window frames, an Event Notification
OAMPDU
                  should be generated with an Errored Frame Period Event TLV
indicating
                    the threshold has been crossed in this window.
                    +------------ Range of errd-frame-period-window values--
----------+
                    | 10 mbps link : 1488 - 892800 |
                    | 100 mbps link : 14880 - 8928000 |
                    | 1000 mbps link : 148810 - 89285760 |
                    | 10000 mbps link : 1488096 - 892857120 |
```


Copyright© 2022 Ciena® Corporation

```
                    +-------------------------------------------------------
----------+";
                reference
                    "[802.3ah], 30.3.6.1.38";
            }

            leaf error-frame-period-threshold {
                type uint32;
                units frames;
                default 1;
                description
                 "The number of frame errors that must occur for this event to
                    be triggered. The default value is one frame error. If
error-frame-period-threshold
                   frame errors occur within a window of error-frame-period-
window
                 frames, an Event Notification OAMPDU should be generated with
                an Errored Frame Period Event TLV indicating the threshold has
                    been crossed in this window.";
                reference
                    "[802.3ah], 30.3.6.1.38";
            }

            leaf error-frame-period-event-notification {
                type boolean;
                default false;
                description
                    "Indicates whether the occurrence of an Errored Frame
Period Event
                should result in an Event Notification OAMPDU generated by the
               OAM layer. By default, this attribute is 'on' for Ethernet like
                interfaces that support OAM. If the OAM layer does not support
                event notifications (as indicated via the functions-supported
                    attribute), this value is ignored.";
            }

            leaf error-frame-window {
                type uint32 {
                    range 10..600;
                }
                units tenths-of-a-second;
                default 10;
                description
                    "The amount of time (in 100ms increments) over which the
threshold
                    is defined. The default value is 10 (1 second). If error-
frame-threshold
                    frame errors occur within a window of error-frame-window
seconds
                 (measured in tenths of seconds), an Event Notification OAMPDU
                should be generated with an Errored Frame Event TLV indicating
           the threshold has been crossed in this window.";
                reference
                    "[802.3ah], 30.3.6.1.38";
            }

            leaf error-frame-threshold {
                type uint32;
                units frames;
                default 1;
```


Copyright© 2022 Ciena® Corporation

```
                description
                 "The number of frame errors that must occur for this event to
                    be triggered. The default value is one frame error. If
error-frame-threshold
                     frame errors occur within a window of error-frame-window
(in tenths
                 of seconds), an Event Notification OAMPDU should be generated
                    with an Errored Frame Event TLV indicating the threshold
has been
                     crossed in this window.";
                reference
                     "[802.3ah], 30.3.6.1.36";
            }

            leaf error-frame-event-notification {
                type boolean;
                default false;
                description
                  "Indicates whether the occurrence of an Errored Frame Event
should
                     result in an Event Notification OAMPDU generated by the
OAM layer.
                     By default, this attribute should be 'on' for Ethernet
like interfaces
                   that support OAM. If the OAM layer does not support event
notifications
                  (as indicated via the functions-supported attribute), this
value
                     is ignored.";
            }

            leaf error-frame-seconds-summary-window {
                type uint16 {
                    range 100..9000;
                }
                default 600;
                units tenths-of-a-second;
                description
                    "The amount of time (in 100ms intervals) over which the
threshold
                     is defined. The default value is 100 (10 seconds). If
error-frame-seconds-summary-threshold
                    frame errors occur within a window of error-frame-seconds-
summary-window
                (in tenths of seconds), an Event Notification OAMPDU should be
                    generated with an Errored Frame Seconds Summary Event TLV
indicating
                     the threshold has been crossed in this window.";
                reference
                     "[802.3ah], 30.3.6.1.40";
            }

            leaf error-frame-seconds-summary-threshold {
                type uint32 {
                   range 1..900;
                }
                units errored-frame-seconds;
                default 1;
                description
                 "The number of errored frame seconds that must occur for this
```


Copyright© 2022 Ciena® Corporation

```
                event to be triggered. The default value is one errored frame
               second. If error-frame-seconds-summary-threshold frame errors
                occur within a window of error-frame-seconds-summary-window
(in
                    tenths of seconds), an Event Notification OAMPDU should
be generated
               with an Errored Frame Seconds Summary Event TLV indicating the
                    threshold has been crossed in this window.";
               reference
                    "[802.3ah], 30.3.6.1.40";
            }

            leaf error-frame-seconds-event-notification {
                type boolean;
                default false;
                description
                 "Indicates whether the occurrence of an Errored Frame Seconds
                  Summary Event should result in an Event Notification OAMPDU
generated
              by the OAM layer. By default, this object should have the value
               'on' for Ethernet like interfaces that support OAM. If the OAM
                  layer does not support event notifications (as indicated
via the
                   functions-supported attribute), this value is ignored.";
            }

            leaf dying-gasp {
                type boolean;
                default false;
                description
                "Indicates whether the local device should attempt to indicate
                 a dying gasp fault via the OAMPDU flags field to its peer OAM
                 entity. If the system does not support dying gasp capability,
                   setting this object has no effect, and reading the object
should
              always result in 'disable'. By default, this object should have
              the value enable for Ethernet like interfaces that support OAM.
                  If the OAM layer does not support event notifications (as
indicated
                    via the functions-supported attribute), this value is
ignored.";
            }

            leaf critical-event {
                type boolean;
                default false;
                description
                "Indicates whether the local device should attempt to indicate
               a critical event has occurred via the OAMPDU flags field to its
                peer OAM entity. By default, this object should have the value
                'enable' for Ethernet like interfaces that support OAM. If the
                 OAM layer does not support event notifications (as indicated
via
                  the functions-supported attribute), this value is ignored.";
            }
            description
                "EOAM port event generation configuration.";
        } // End container port-event-config
        description
            "EOAM port configuration.";
```


Copyright© 2022 Ciena® Corporation

```
    }// End augment

    augment "/if:interfaces-state/if:interface" {
       when "if:type = 'ianaift:ethernetCsmacd'";

        container port-status {

            leaf port-oper-status {
                type oper-status;
                description
                 "The OAM operational status for the Ethernet like interface";
            }

            leaf max-pdu-size {
                type uint32 {
                    range 64..1518;
                }
                description
                "The largest OAMPDU that the OAM entity supports. OAM entities
                exchange maximum OAMPDU sizes and negotiate to use the smaller
                 of the two maximum OAMPDU sizes between the peers. This value
                    is determined by the local implementation.";
            }

            leaf configuration-revision {
                type uint32 {
                    range 0..65535;
                }
                description
                 "The configuration revision of the OAM entity as reflected in
                  the latest OAMPDU sent by the OAM entity. The configuration
revision
                is used by OAM entities to indicate configuration changes have
                occurred which might require the peer OAM entity to re-evaluate
                    whether the peering is allowed. See local_satisfied in
[802.3ah,
                    57.3.1.2].";
            }

            uses port-config-objects;

            leaf local-functions-supported {
                type functions-supported;
                description
                 "The OAM functions supported on this Ethernet like interface.
                   OAM consists of separate functional sets beyond the basic
discovery
                 process which is always required. These functional groups can
                be supported independently by any implementation. These values
                are communicated to the peer via the local configuration field
                    of Information OAMPDUs.";
            }
            description
                "EOAM port operational status.";
        } // End container port-status

        container peer-information {

            leaf peer-status {
                type peer-status;
```

```
                description
                    " Operational status of the peer OAM interface.";
            }

            leaf peer-mac-address {
                type yang:mac-address;
                description
                    "The MAC address of the peer OAM entity. The MAC address
is derived
                     from the most recently received OAMPDU. This value is
initialized
                     to all zeros (0x000000000000). This value is invalid if
the wwpLeosOamPeerStatus
                   is inactive. An OAMPDU is indicated by a valid frame with (1)
                     destination MAC address equal to that of the reserved MAC
address
                   for Slow Protocols (See 43B of [802.3ah]), (2) a lengthOrType
                   field equal to the reserved type for Slow Protocols, (3) and a
                    Slow Protocols subtype equal to that of the subtype reserved
for
                     OAM.";
            }

            leaf peer-vendor-oui {
                type oui;
                description
                  "The OUI of the OAM peer as reflected in the latest Information
                     OAMPDU received with a Local Information TLV. The OUI can
be used
                   to identify the vendor of the remote OAM entity. This value is
                    initialized to all zeros (0x000000). This value is considered
                    invalid if the peer-status is inactive. An Information OAMPDU
                   is indicated by a valid frame with (1) destination MAC address
                    equal to that of the reserved MAC address for Slow Protocols
(See
                     43B of [802.3ah]), (2) a lengthOrType field equal to the
reserved
                   type for Slow Protocols, (3) a Slow Protocols subtype equal to
                    that of the subtype reserved for OAM, and (4) an OAM code that
                     equals the code reserved for Information OAMPDUs.";
            }

            leaf vendor-info {
                type uint32;
                description
                  "The Vendor Info of the OAM peer as reflected in the latest
Information
                     OAMPDU received with a Local Information TLV. The vendor
information
                    field is within the Local Information TLV, and can be used to
                     determine additional information about the peer entity.
The format
                     of the vendor information is unspecified within the 32-
bit field.
                  This value is initialized to all zeros (0x00000000). This value
                     is invalid if the peer-status is inactive. An Information
OAMPDU
                   is indicated by a valid frame with (1) destination MAC address
                    equal to that of the reserved MAC address for Slow Protocols
(See
```


Copyright© 2022 Ciena® Corporation

```
                    43B of [802.3ah]), (2) a lengthOrType field equal to the
reserved
                type for Slow Protocols, (3) a Slow Protocols subtype equal to
                that of the subtype reserved for OAM, and (4) an OAM code that
                    equals the code reserved for Information OAMPDUs. ";
            }

            leaf peer-mode {

                type enumeration {
                    enum active {
                        description
                            "OAM peer is in active mode.";
                    }
                    enum passive {
                        description
                            "OAM peer is in passive mode.";
                    }
                    enum unknown {
                        description
                            "peer-mode is initialized to 'unknown' and is
                            not valid if peer-status is inactive.";
                    }
                }
                description
                    "The mode of the OAM peer as reflected in the latest
Information
                 OAMPDU received with a Local Information TLV. The mode of the
                  peer can be determined from the Configuration field in the
Local
                    Information TLV of the last Information OAMPDU received
from the
               peer. This value is initialized to unknown(3), and is not valid
                    if the peer-status is inactive. An Information OAMPDU
                is indicated by a valid frame with (1) destination MAC address
                 equal to that of the reserved MAC address for Slow Protocols
(See
                    43B of [802.3ah]), (2) a lengthOrType field equal to the
reserved
                type for Slow Protocols, (3) a Slow Protocols subtype equal to
                that of the subtype reserved for OAM, and (4) an OAM code that
                    equals the code reserved for Information OAMPDUs.";
            }

            leaf peer-max-pdu-size {
                type uint32;
                units bytes;
                description
                "The maximum size of OAMPDU supported by the peer as reflected
                    in the latest Information OAMPDU received with a Local
Information
                 TLV. Ethernet OAM on this interface must not use OAMPDUs that
                  exceed this size. The maximum OAMPDU size can be determined
from
               the PDU Configuration field of the Local Information TLV of the
                last Information OAMPDU received from the peer. This value is
                    initialized to 64, and is invalid if the peer-status is
inactive.
                   An Information OAMPDU is indicated by a valid frame with
(1) destination
```


Copyright© 2022 Ciena® Corporation

```
                MAC address equal to that of the reserved MAC address for Slow
                  Protocols (See 43B of [802.3ah]), (2) a lengthOrType field
equal
                 to the reserved type for Slow Protocols, (3) a Slow Protocols
                subtype equal to that of the subtype reserved for OAM, and (4)
                   an OAM code that equals the code reserved for Information
OAMPDUs.";
            }

            leaf peer-configuration-revision {
                type uint32;
                description
               "The configuration revision of the OAM peer as reflected in the
                latest OAMPDU. This attribute is changed by the peer whenever
                   it has a local configuration change for Ethernet OAM this
interface.
                    This value is initialized to all zeros (0x00000000), and
is invalid
                if the peer-status is inactive. The configuration revision can
                be determined from the Revision field of the Local Information
                   TLV of the most recently received Information OAMPDU with
a Local
                Information TLV. An Information OAMPDU is indicated by a valid
                    frame with (1) destination MAC address equal to that of
the reserved
                  MAC address for Slow Protocols (See 43B of [802.3ah]), (2)
a lengthOrType
               field equal to the reserved type for Slow Protocols, (3) a Slow
                    Protocols subtype equal to that of the subtype reserved
for OAM,
                    and (4) an OAM code that equals the code reserved for
Information
                    OAMPDUs.";
            }

            leaf peer-functions-supported {
                type functions-supported;
                description
                   "The OAM functions supported on the peer. OAM consists of
separate
                  functionality sets above the basic discovery process. This
value
                indicates the capabilities of the peer OAM entity with respect
                 to these functions. This value is initialized so all bits are
                clear, and is invalid if the wwpLeosOamPeerStatus is inactive.
                   The capbilities of the OAM peer can be determined from the
configuration
                    field of the Local Information TLV of the most recently
received
               Information OAMPDU with a Local Information TLV. An Information
                 OAMPDU is indicated by a valid frame with (1) destination MAC
                   address equal to that of the reserved MAC address for Slow
Protocols
                 (See 43B of [802.3ah]), (2) a lengthOrType field equal to the
                reserved type for Slow Protocols, (3) a Slow Protocols subtype
                 equal to that of the subtype reserved for OAM, and (4) an OAM
                code that equals the code reserved for Information OAMPDUs. ";
            }
            description
                "Operational status of the EOAM peer.";
```


Copyright© 2022 Ciena® Corporation

```
        } // End container peer-status

        container loopback-port-status {

            leaf loopback-status {
                type enumeration {
                    enum no-loopback;
                    enum initiating-loopback;
                    enum remote-loopback;
                    enum terminating-loopback;
                    enum local-loopback;
                    enum unknown;
                }
                description
                    "The loopback status of the OAM entity.  This status is
                    determined by a combination of the local parser and
                multiplexer states, the remote parser and multiplexer states,
                    as well as by the actions of the local OAM client.  When
                   operating in normal mode with no loopback in progress, the
                    status reads no-loopback.

                  If the OAM client has sent a Loopback OAMPDU and is waiting
                 for a response, where the local parser and multiplexer states
                    are DISCARD (see [802.3ah, 57.2.11.1]), the status is
                'initiating-loopback'.  In this case, the local OAM entity has
                 yet to receive any acknowledgement that the remote OAM entity
                    has received its loopback command request.

                If the local OAM client knows that the remote OAM entity is in
                 loopback mode (via the remote state information as described
                 in [802.3ah, 57.2.11.1, 30.3.6.1.15]), the status is remote-
loopback.
                   If the local OAM client is in the process of terminating
the remote
                  loopback [802.3ah, 57.2.11.3, 30.3.6.1.14], with its local
                    multiplexer and parser states in DISCARD, the status is
                    terminating-loopback.  If the remote OAM client has put
the local
                    OAM entity in loopback mode as indicated by its local
parser state,
                    the status is local-loopback.

                    The unknown status indicates the parser and multiplexer
                combination is unexpected.  This status may be returned if the
                OAM loopback is in a transition state but should not persist.

                    The values of this attribute correspond to the following
                values of the local and remote parser and multiplexer states.

                                                LclPrsr   LclMux    RmtPrsr   RmtMux
                   no-loopback                    FWD       FWD       FWD       FWD
                  initiating-loopback          DISCARD   DISCARD     FWD       FWD
                   remote-loopback            DISCARD     FWD      LPBK     DISCARD
                   terminating-loopback       DISCARD   DISCARD    LPBK     DISCARD
                   local-loopback               LPBK    DISCARD   DISCARD     FWD
                  unknown                      ***   any other combination   ***";
            }
            description
                "EOAM loopback port status.";
        } // End container loopback-port-status
```

```
        container port-statistics {

            leaf information-transmitted {
                type yang:counter32;
                units frames;
                description
                   "A count of the number of Information OAMPDUs transmitted
on this
                interface. An Information OAMPDU is indicated by a valid frame
                with (1) destination MAC address equal to that of the reserved
                  MAC address for Slow Protocols (See 43B of [802.3ah]), (2)
a lengthOrType
               field equal to the reserved type for Slow Protocols, (3) a Slow
                    Protocols subtype equal to that of the subtype reserved
for OAM,
                    and (4) an OAMPDU code that equals the OAM Information
code. Discontinuities
                    of this counter can occur at re-initialization of the
management
                  system, and at other times as indicated by the value of the
                    ifCounterDiscontinuityTime.";
            }

            leaf information-received {
                type yang:counter32;
                units frames;
                description
                "A count of the number of Information OAMPDUs received on this
                interface. An Information OAMPDU is indicated by a valid frame
                with (1) destination MAC address equal to that of the reserved
                  MAC address for Slow Protocols (See 43B of [802.3ah]), (2)
a lengthOrType
               field equal to the reserved type for Slow Protocols, (3) a Slow
                    Protocols subtype equal to that of the subtype reserved
for OAM,
                    and (4) an OAMPDU code that equals the OAM Information
code. Discontinuities
                    of this counter can occur at re-initialization of the
management
               system, and at other times as indicated by the value of the
                    ifCounterDiscontinuityTime.";
            }

            leaf unique-event-notification-transmitted {
                type yang:counter32;
                units frames;
                description
                 "A count of the number of unique Event OAMPDUs transmitted on
               this interface. Event notifications may be sent in duplicate to
               increase the probability of successfully being received, given
                 the possibility that a frame may be lost in transit. An Event
                  Notification OAMPDU is indicated by a valid frame with (1)
destination
                MAC address equal to that of the reserved MAC address for Slow
                  Protocols (See 43B of [802.3ah]), (2) a lengthOrType field
equal
                 to the reserved type for Slow Protocols, (3) a Slow Protocols
                subtype equal to that of the subtype reserved for OAM, and (4)
                 an OAMPDU code that equals the OAM Event code. A unique Event
```


Copyright© 2022 Ciena® Corporation

```
                    Notification OAMPDU is indicated as an Event Notification
OAMPDU
                      with a Sequence Number field that is distinct from the
previously
                      transmitted Event Notification OAMPDU Sequence Number.
Discontinuities
                      of this counter can occur at re-initialization of the
management
                    system, and at other times as indicated by the value of the
                      ifCounterDiscontinuityTime";
            }

            leaf unique-event-notification-received {
                type yang:counter32;
                units frames;
                description
                "A count of the number of unique Event OAMPDUs received on this
                interface. Event notification OAMPDUs may be sent in duplicate
                  to increase the probability of successfully being received,
given
                 the possibility that a frame may be lost in transit. An Event
                  Notification OAMPDU is indicated by a valid frame with (1)
destination
                MAC address equal to that of the reserved MAC address for Slow
                  Protocols (See 43B of [802.3ah]), (2) a lengthOrType field
equal
                 to the reserved type for Slow Protocols, (3) a Slow Protocols
                subtype equal to that of the subtype reserved for OAM, and (4)
                 an OAMPDU code that equals the OAM Event code. A unique Event
                  Notification OAMPDU is indicated as an Event Notification
OAMPDU
                    with a Sequence Number field that is distinct from the
previously
                    received Event Notification OAMPDU Sequence Number.
Discontinuities
                    of this counter can occur at re-initialization of the
management
                    system, and at other times as indicated by the value of the
                    ifCounterDiscontinuityTime.";
            }

            uses loopback-statistics;

            leaf variable-request-transmitted {
                type yang:counter32;
                units frames;
                description
                "A count of the number of Variable Request OAMPDUs transmitted
                on this interface. A Variable Request OAMPDU is indicated by a
                 valid frame with (1) destination MAC address equal to that of
                    the reserved MAC address for Slow Protocols (See 43B of
[802.3ah]),
                    (2) a lengthOrType field equal to the reserved type for
Slow Protocols,
                  (3) a Slow Protocols subtype equal to that of the subtype
reserved
                 for OAM, and (4) an OAMPDU code that equals the OAM Variable
Request
                    code. Discontinuities of this counter can occur at re-
initialization
```


Copyright© 2022 Ciena® Corporation

```
                    of the management system, and at other times as indicated
by the
                     value of the ifCounterDiscontinuityTime.";
                reference
                    "[802.3ah], 30.3.6.1.28.";
            }

            leaf variable-request-received {
                type yang:counter32;
                units frames;
                description
                "A count of the number of Variable Request OAMPDUs received on
                   this interface. A Variable Request OAMPDU is indicated by
a valid
                    frame with (1) destination MAC address equal to that of
the reserved
                  MAC address for Slow Protocols (See 43B of [802.3ah]), (2)
a lengthOrType
               field equal to the reserved type for Slow Protocols, (3) a Slow
                    Protocols subtype equal to that of the subtype reserved
for OAM,
                 and (4) an OAMPDU code that equals the OAM Variable Request
code.
               Discontinuities of this counter can occur at re-initialization
                   of the management system, and at other times as indicated
by the
                    value of the ifCounterDiscontinuityTime.";
            }

            leaf variable-response-transmitted {
                type yang:counter32;
                units frames;
                description
               "A count of the number of Variable Response OAMPDUs transmitted
                on this interface. A Variable Response OAMPDU is indicated by
               a valid frame with (1) destination MAC address equal to that of
                    the reserved MAC address for Slow Protocols (See 43B of
[802.3ah]),
                    (2) a lengthOrType field equal to the reserved type for
Slow Protocols,
                   (3) a Slow Protocols subtype equal to that of the subtype
reserved
                for OAM, and (4) an OAMPDU code that equals the OAM Variable
Response
                    code. Discontinuities of this counter can occur at re-
initialization
                   of the management system, and at other times as indicated
by the
                    value of the ifCounterDiscontinuityTime.";
            }

            leaf variable-response-received {
                type yang:counter32;
                units frames;
                description
               "A count of the number of Variable Response OAMPDUs received on
                    this interface. A Variable Response OAMPDU is indicated
by a valid
                    frame with (1) destination MAC address equal to that of
the reserved
```


Copyright© 2022 Ciena® Corporation

```
                    MAC address for Slow Protocols (See 43B of [802.3ah]), (2)
a lengthOrType
                  field equal to the reserved type for Slow Protocols, (3) a Slow
                      Protocols subtype equal to that of the subtype reserved
for OAM,
                   and (4) an OAMPDU code that equals the OAM Variable Response
code.
                  Discontinuities of this counter can occur at re-initialization
                     of the management system, and at other times as indicated
by the
                      value of the ifCounterDiscontinuityTime.";
              }

              leaf organization-specific-transmitted {
                  type yang:counter32;
                  units frames;
                  description
                      "A count of the number of Organization Specific OAMPDUs
transmitted
                 on this interface. An Organization Specific OAMPDU is indicated
                  by a valid frame with (1) destination MAC address equal to that
                      of the reserved MAC address for Slow Protocols (See 43B
of [802.3ah]),
                      (2) a lengthOrType field equal to the reserved type for
Slow Protocols,
                     (3) a Slow Protocols subtype equal to that of the subtype
reserved
                      for OAM, and (4) an OAMPDU code that equals the OAM
Organization
                     Specific code. Discontinuities of this counter can occur
at re-initialization
                    of the management system, and at other times as indicated
by the
                      value of the ifCounterDiscontinuityTime.";
              }

              leaf organization-specific-received {
                  type yang:counter32;
                  units frames;
                  description
                      "A count of the number of Organization Specific OAMPDUs
received
                 on this interface. An Organization Specific OAMPDU is indicated
                  by a valid frame with (1) destination MAC address equal to that
                      of the reserved MAC address for Slow Protocols (See 43B
of [802.3ah]),
                      (2) a lengthOrType field equal to the reserved type for
Slow Protocols,
                     (3) a Slow Protocols subtype equal to that of the subtype
reserved
                      for OAM, and (4) an OAMPDU code that equals the OAM
Organization
                     Specific code. Discontinuities of this counter can occur
at re-initialization
                    of the management system, and at other times as indicated
by the
                      value of the ifCounterDiscontinuityTime.";
              }

              leaf unsupported-codes-transmitted {
```


Copyright© 2022 Ciena® Corporation

```
                type yang:counter32;
                units frames;
                description
                    "A count of the number of Organization Specific OAMPDUs
received
               on this interface. An Organization Specific OAMPDU is indicated
                by a valid frame with (1) destination MAC address equal to that
                    of the reserved MAC address for Slow Protocols (See 43B
of [802.3ah]),
                    (2) a lengthOrType field equal to the reserved type for
Slow Protocols,
                   (3) a Slow Protocols subtype equal to that of the subtype
reserved
                    for OAM, and (4) an OAMPDU code that equals the OAM
Organization
                    Specific code. Discontinuities of this counter can occur
at re-initialization
                   of the management system, and at other times as indicated
by the
                    value of the ifCounterDiscontinuityTime.";
            }

            leaf unsupported-codes-received {
                type yang:counter32;
                units frames;
                description
                 "A count of the number of OAMPDUs received on this interface
with
                    an unsupported op-code. An unsupported op-code OAMPDU is
indicated
                by a valid frame with (1) destination MAC address equal to that
                    of the reserved MAC address for Slow Protocols (See 43B
of [802.3ah]),
                    (2) a lengthOrType field equal to the reserved type for
Slow Protocols,
                   (3) a Slow Protocols subtype equal to that of the subtype
reserved
                  for OAM, and (4) an OAMPDU code that equals the opcode for
a function
                    that is not supported by the device. Discontinuities of
this counter
               can occur at re-initialization of the management system, and at
                    other times as indicated by the value of the
ifCounterDiscontinuityTime.";
            }

            leaf frames-lost-due-to-oam {
                type yang:counter32;
                units frames;
                description
                 "A count of the number of OAMPDUs received on this interface
with
                    an unsupported op-code. An unsupported op-code OAMPDU is
indicated
                by a valid frame with (1) destination MAC address equal to that
                    of the reserved MAC address for Slow Protocols (See 43B
of [802.3ah]),
                    (2) a lengthOrType field equal to the reserved type for
Slow Protocols,
```


Copyright© 2022 Ciena® Corporation

```
                    (3) a Slow Protocols subtype equal to that of the subtype
reserved
                   for OAM, and (4) an OAMPDU code that equals the opcode for
a function
                     that is not supported by the device. Discontinuities of
this counter
                can occur at re-initialization of the management system, and at
                     other times as indicated by the value of the
ifCounterDiscontinuityTime.";
            }

            leaf duplicate-event-notifications-transmitted {
                type yang:counter32;
                units frames;
                description
                 "A count of the number of duplicate Event OAMPDUs transmitted
                   on this interface. Event notification OAMPDUs may be sent
in duplicate
                  to increase the probability of successfully being received,
given
                  the possibility that a frame may be lost in transit. An Event
                   Notification OAMPDU is indicated by a valid frame with (1)
destination
                 MAC address equal to that of the reserved MAC address for Slow
                   Protocols (See 43B of [802.3ah]), (2) a lengthOrType field
equal
                  to the reserved type for Slow Protocols, (3) a Slow Protocols
                 subtype equal to that of the subtype reserved for OAM, and (4)
                   an OAMPDU code that equals the OAM Event code. A duplicate
Event
                    Notification OAMPDU is indicated as an Event Notification
OAMPDU
                      with a Sequence Number field that is identical to the
previously
                      transmitted Event Notification OAMPDU Sequence Number.
Discontinuities
                      of this counter can occur at re-initialization of the
management
                   system, and at other times as indicated by the value of the
                      ifCounterDiscontinuityTime.";
            }

            leaf duplicate-event-notifications-received {
                type yang:counter32;
                units frames;
                description
                 "A count of the number of duplicate Event OAMPDUs received on
                   this interface. Event notification OAMPDUs may be sent in
duplicate
                  to increase the probability of successfully being received,
given
                  the possibility that a frame may be lost in transit. An Event
                   Notification OAMPDU is indicated by a valid frame with (1)
destination
                 MAC address equal to that of the reserved MAC address for Slow
                   Protocols (See 43B of [802.3ah]), (2) a lengthOrType field
equal
                  to the reserved type for Slow Protocols, (3) a Slow Protocols
                 subtype equal to that of the subtype reserved for OAM, and (4)
```


Copyright© 2022 Ciena® Corporation

```
                    an OAMPDU code that equals the OAM Event code. A duplicate
Event
                     Notification OAMPDU is indicated as an Event Notification
OAMPDU
                       with a Sequence Number field that is identical to the
previously
                       received Event Notification OAMPDU Sequence Number.
Discontinuities
                       of this counter can occur at re-initialization of the
management
                     system, and at other times as indicated by the value of the
                       ifCounterDiscontinuityTime.";
                }
                description
                    "EOAM port statistics.";
            } // End container port-statistics
            description
                "EOAM port operational information.";
        } // End augment

        container port-event-log {
            config false;

            list port-entry {
                key "port-number index";

                leaf port-number {
                    type uint32;
                }

                leaf index {
                    type uint32 {
                        range 1..4294967295;
                    }
                    description
                    "An arbitrary integer for identifying individual events within
                        the event log.";
                }

                leaf timestamp {
                    type yang:timestamp;
                    description
                        "The value of sysUpTime at the time of the logged event.
For locally
                       generated events, the time of the event can be accurately
retrieved
                    from sysUpTime. For remotely generated events, the time of the
                     event is indicated by the reception of the Event Notification
                    OAMPDU indicating the event occurred on the peer. A system may
                        attempt to adjust the timestamp value to more accurately
reflect
                      the time of the event at the peer OAM entity by using other
information,
                        such as that found in the timestamp found of the Event
Notification
                    TLVs, which provides an indication of the relative time between
                        events at the peer entity.";
                }

                leaf oui {
```


Copyright© 2022 Ciena® Corporation

```
                type oui;
                description
                "The OUI of the entity defining the object type. All IEEE 802.3
                    defined events (as appearing in [802.3ah] except for the
Organizationally
                     Unique Event TLVs) use the IEEE 802.3 OUI of 0x0180C2.
Organizations
                defining their own Event Notification TLVs include their OUI in
                     the Event Notification TLV which gets reflected here.";
            }

            leaf event-type {
                type event;
            }

            leaf location {
                type event-location;
            }

            leaf window-hi {
                type uint32;
                description
                    "If the event represents a threshold crossing event, the
two objects
                     window-hi and window-lo form an unsigned 64-bit integer
yielding
                the window over which the value was measured for the threshold
                   crossing event (e.g. 5, when 11 occurrences happened in 5
seconds
                 while the threshold was 10). The two objects are combined as:
                window = ((2^32) * window-hi) + window-lo Otherwise, this value
                     is returned as all Fs (0xFFFFFFFF) and adds no useful
information.";
                reference
                    "[802.3ah], 30.3.6.1.37 and 57.5.3.2.";
            }

            leaf window-lo {
                type uint32;
                description
                    "If the event represents a threshold crossing event, the
two objects
                     window-hi and window-lo form an unsigned 64-bit integer
yielding
                the window over which the value was measured for the threshold
                   crossing event (e.g. 5, when 11 occurrences happened in 5
seconds
                 while the threshold was 10). The two objects are combined as:
                window = ((2^32) * window-hi) + window-lo Otherwise, this value
                     is returned as all Fs (0xFFFFFFFF) and adds no useful
information.";
                reference
                    "[802.3ah], 30.3.6.1.37 and 57.5.3.2.";
            }

            leaf threshold-hi {
                type uint32;
                description
                    "If the event represents a threshold crossing event, the
two objects
```


Copyright© 2022 Ciena® Corporation

```
                threshold-hi and threshold-lo form an unsigned 64-bit integer
                yielding the value that was crossed for the threshold crossing
                event (e.g. 10, when 11 occurrences happened in 5 seconds while
                    the threshold was 10). The two objects are combined as:
threshold
                = ((2^32) * threshold-hi) + threshold-lo Otherwise, this value
                    is returned as all Fs (0xFFFFFFFF) and adds no useful
information.";
                reference
                    "[802.3ah], 30.3.6.1.37 and 57.5.3.2.";
            }

            leaf threshold-lo {
                type uint32;
                description
                   "If the event represents a threshold crossing event, the
two objects
                threshold-hi and threshold-lo form an unsigned 64-bit integer
                yielding the value that was crossed for the threshold crossing
                   event (e.g. 10, when 11 occurrences happened in 5 seconds
while
                    the threshold was 10). The two objects are combined as:
threshold
                = ((2^32) * threshold-hi) + threshold-lo Otherwise, this value
                    is returned as all Fs (0xFFFFFFFF) and adds no useful
information.";
                reference
                    "[802.3ah], 30.3.6.1.37 and 57.5.3.2.";
            }

            leaf value {
                type yang:counter64;
                description
                  "If the event represents a threshold crossing event, this
value
                    indicates the value of the parameter within the given
window that
                generated this event (e.g. 11, when 11 occurrences happened in
                    5 seconds while the threshold was 10). Otherwise, this
value is
                  returned as all Fs (0xFFFFFFFFFFFFFFFF) and adds no useful
information.";
                reference
                    "[802.3ah], 30.3.6.1.37 and 57.5.3.2.";
            }

            leaf running-total {
                type yang:counter64;
                description
                    "This value represents the total number of times this
occurrence
                has happened since the last reset (e.g. 3253, when 3253 symbol
                errors have occurred since the last reset, which has resulted
                    in 51 symbol error threshold crossing events since the
last reset).";
                reference
                    "[802.3ah], 30.3.6.1.37 and 57.5.3.2.";
            }

            leaf total {
```


Copyright© 2022 Ciena® Corporation

```
                type uint32;
                description
               "This value represents the total number of times one or more of
                these occurrences have resulted in an event (e.g. 51 when 3253
                    symbol errors have occurred since the last reset, which
has resulted
                     in 51 symbol error threshold crossing events since the
last reset).";
                reference
                    "[802.3ah], 30.3.6.1.37 and 57.5.3.2.";
            }
         } // End list
         description
            "This table records a history of the events that have occurred
          at the Ethernet OAM level. These events can include locally detected
           events, which may result in locally generated OAMPDUs, and remotely
             detected events, which are detected by the OAM peer entity and
           signaled to the local entity via Ethernet OAM. Ethernet OAM events
             can be signaled by Event Notification OAMPDUs or by the flags
             field in any OAMPDU.";
   } // End container port-event-log

    rpc loopback-command {
        description
            "Initiate or terminate remote loopback with an OAM peer.";
        input {
            leaf loopback-command {
                type enumeration {
                    enum start-remote-loopback {
                        description
                          "start-remote-loopback causes the local OAM client
to send a loopback
                             OAMPDU to the OAM peer with the loopback enable
flags set. These
                             setting only take effect is the oper-status for
the Ethernet port
                       is operational.";
                    }
                    enum stop-remote-loopback {
                        description
                           "stop-remote-loopback causes the local OAM client
to send a loopback
                             OAMPDU to the OAM peer with the loopback enable
flags cleared.
                            These setting only take effect is the oper-status
for the Ethernet
                             port is operational.";

                    }
                }
            }
        }
        output {
            uses loopback-statistics;
        }
    } // End rpc loopback-command

    notification link-event {

        leaf log-port {
```


Copyright© 2022 Ciena® Corporation

```
            type uint32;
            description
                "An Ethernet port on the switch.";
        }
        leaf log-type {
            type event;
        }

        leaf log-location {
            type event-location;
        }
    } // End notification link-event
} // End EOAM module
```


Copyright© 2022 Ciena® Corporation

ciena-ieee-rstp.yang

```
module ciena-ieee-rstp {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-ieee-rstp";
    prefix "rstp";

    import iana-if-type {
        prefix "ianaift";
    }

    import ietf-interfaces {
        prefix "if";
    }

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
                 Hanover, Maryland 21076
                 U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
        "This YANG module contains the schema
         definition for RSTP.";

    revision "2016-10-25" {
        description
            "Added rstp-port-config for lag interfaces.";
    }

    revision "2015-03-14" {
        description
            "Initial Version.";
        reference "IEEE Std 802.1D";
    }

    container global-config {
        description
            "Global RSTP configuration settings.";
        leaf admin-enabled {
            type boolean;
            description
                "Global enable/disable for RSTP.";
        }
    }

    augment "/if:interfaces/if:interface" {
        when "if:type = 'ianaift:ethernetCsmacd' or
              if:type = 'ianaift:ieee8023adLag'" {
            description
                "Augment only if the interface type
                 is ethernetCSmacd or ieee8023adLag.";
        }
        description
            "Augmenting ietf-interface module to add
```


Copyright© 2022 Ciena® Corporation

```
                configuration parameters for RSTP.";

            container port-config {
                description
                    "per port configurable variables.";
                leaf admin-enabled {
                    type boolean;
                    description
                      "Per-port enable/disable for RSTP.";
                }
            }
        }
    }
```


Copyright© 2022 Ciena® Corporation

ciena-ietf-igmp-deviation.yang

```
module ciena-ietf-igmp-deviation {
    namespace "http://www.ciena.com/ns/yang/ciena-ietf-igmp-dev";
    prefix "ciena-ietf-igmp-dev";

    import ciena-ietf-igmp {
        prefix "igmp";
    }

    revision "2016-07-13" {
        description
            "Initial version";
    }

    deviation /igmp:multicast-services/igmp:instance/igmp:igmp-snooping/
igmp:robustness {
        deviate replace {
            type uint32 {
                range "1..5";
            }
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-ietf-igmp.yang

```
module ciena-ietf-igmp {
    namespace "http://www.ciena.com/ns/yang/igmp";
    prefix "igmp";

    import ietf-yang-types {
        prefix yang;
        revision-date 2010-09-24;
    }
    import ietf-inet-types {
        prefix inet;
        revision-date 2010-09-24;
    }

    /* May need to change the following yang module names. */

    import ciena-mef-fd {
        prefix "mef-fd";
    }
    import ciena-mef-fp {
        prefix "mef-fp";
    }

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
        Hanover, Maryland 21076
        U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
        "This module contains the YANG definition for Ciena's global
        configuration of Internet Group Management Protocol (IGMP),
                      IGMP Snooping, Proxy and Multicast features.

                          Copyright (c) 2016 Ciena Corporation.  All rights
                          reserved.

                          All information contained herein is, and remains
                      the property of Ciena Corporation. Dissemination of this
                      information or reproduction of this material is strictly
                    forbidden unless prior written permission is obtained from
                          Ciena Corporation.";

    revision "2016-03-21" {
        description
            "Initial Version.";
        reference
            "RFC 1112: Host Extensions for IP Multicasting (IGMPv1)
            RFC 2236: Internet Group Management Protocol, Version 2
            (IGMPv2)
            RFC 3376: Internet Group Management Protocol, Version 3
            (IGMPv3)
            RFC 4604: Using Internet Group Management Protocol Version 3
```

```
            (IGMPv3) and Multicast Listener Discovery Protocol
            Version 2 (MLDv2) for Source-Specific Multicast
            IETF magma snoop, draft-ietf-magma-snoop-12.txt:
            Considerations for IGMP and MLD Snooping Switches
            IETF magma igmp proxy, draft-ietf-magma-igmp-proxy-6.txt:
            IGMP/MLD Proxying";
    }

    /*
     * Type definitions
     */

    typedef state {
        type enumeration {
            enum disable {
                description
                    "Disable multicast service.";
            }
            enum enable {
                description
                    "Enable multicast service.";
            }
        }

        description
            "State of multicast service or snooping.";
    }

    identity igmp-compatibility-mode {
        description
            "Base identity IGMP version based compatibility mode.";
    }

    identity  v1 {
        base igmp-compatibility-mode;
        description
            "IGMP version v1.";
    }

    identity v2 {
        base igmp-compatibility-mode;
        description
            "IGMP version v2.";
    }

    identity v3 {
        base igmp-compatibility-mode;
        description
            "IGMP version v3.";
    }

    identity multicast-service-state {
        description
            "Base identity of the current state of the
            multicast service.";
    }

    identity active {
        base multicast-service-state;
        description
```


Copyright© 2022 Ciena® Corporation

```
            "Specify the group is activ.";
    }

    identity query {
        base multicast-service-state;
        description
            "Specify that the query message has been sent.";
    }

    identity active-linger {
        base multicast-service-state;
        description
            "Specify that the member count is zero and the
            leave message is not sent to upstream router.";
    }

    identity linger {
        base multicast-service-state;
        description
            "Specify that the member count is zero and the
            leave message is sent to upstream router.";
    }

    identity multicast-traffic-source {
        description
            "Base identity of the source for this multicast
            group traffic.";
    }

    identity router {
        base multicast-traffic-source;
        description
            "Represents the multicast router.";
    }

    identity server {
        base multicast-traffic-source;
        description
            "Represents the actual multicast server from where
            traffic is originated.";
    }

    identity multicast-service-type {
        description
            "Base identity of the type of the multicast service.";
    }

    identity  static {
        base multicast-service-type;
        description
            "Represents the static multicast group.";
    }

    identity dynamic {
        base multicast-service-type;
        description
            "Represents the dynamic multicast group.";
    }

    identity filter-mode {
```


Copyright© 2022 Ciena® Corporation

```
        description
            "Base identity of igmp v3 source address filter mode
            for the multicast group.";
    }

    identity  include {
        base filter-mode;
        description
            "Include membership.";
    }

    identity exclude {
        base filter-mode;
        description
            "Include membership.";
    }

    identity server-topology-mode {
        description
            "Base identity of the server topology mode.";
    }

    identity centralized {
        base server-topology-mode;
        description
            "Centralized server mode.";
    }

    identity distributed {
        base server-topology-mode;
        description
            "Distributed server mode.";
    }

    identity igmp-leave-mode {
        description
            "Base identity of the igmp leave mode for multicast service..";
    }

    identity fast {
        base igmp-leave-mode;
        description
            "Fast leave.";
    }

    identity inquisitive {
        base igmp-leave-mode;
        description
            "Inquisitive leave.";
    }

    grouping fd-ref {
        description
            "Represents a forwarding domain.";

        leaf fd-name {
            type leafref {
                path "/mef-fd:fds/mef-fd:fd/mef-fd:name";
            }
```


Copyright© 2022 Ciena® Corporation

```
            description
                "A reference to a forwarding domain.";
        }
    }

    //
    // Data definitions related to IGMP
    //

    container multicast-services {
        description
            "Igmp multicast services.";

        leaf global-admin-state {
            type state;
            default enable;

            description
                "Specifies the global multicast admin state.";
        }

        leaf global-snoop-state {
            type state;
            default disable;

            description
                "Specifies the igmp global snooping state.";
        }

        list instance {
            key "fd-name";

            description
                "Specifies a list of multicast service instances.";

            uses fd-ref;

            leaf admin-state {
                type state;
                default enable;

                description
                    "Specifies the admin state of multicast-service on a
                    instance.";
            }

            leaf umf-state {
                type enumeration {
                    enum drop {
                        description
                           "Causes the device to drop all unresolved messages
                            on service instance.";
                    }
                    enum flood {
                        description
                       "Causes the device to flood all unresolved messages to
                            all interfaces on service instance.";
                    }
                }
                default flood;
```


Copyright© 2022 Ciena® Corporation

```
        description
            "Specifies how the unknown multicast frames are handled
            on instance.";
    }

    container server-interface {
        description
            "Specifies the list of authorized server ports.";

        leaf-list interface {
            type mef-fp:fp-ref;

            description
                "
                Server interface from which the igmp query messages
                are received, and joins/leaves messages are sent.";
        }
    }

    container igmp-snooping {

        description
         "Specifies the configuration parameters for igmp snooping.";

        leaf admin-state {
            type state;
            default enable;

            description
                "Specifies the admin state of igmp snooping for
                multicast service.";
        }

        leaf active-linger-timeout {
            type uint32 {
                range "0..300";
            }
            units "seconds";
            default "30";

            description
                "Specifies the active linger timeout.";
        }

        leaf default-router-interface {
            type mef-fp:fp-ref;

            description
                "Specify the default router interface for
                multicast service.";
        }

        leaf fork-mode {
            type enumeration {
                enum on {
                    description
                        "Forking mode on.";
                }
```

```
            enum off {
                description
                    "Forking mode off.";
            }
        }
        default off;

        description
            "Specify the igmp forking mode state.";
    }

    leaf last-member-query-interval {
        type uint32 {
            range "10..100";
        }
        units "deci-seconds(Tenths of a sec)";
        default "10";

        description
            "Specify the last membership query interval for
            multicast service.";
    }
    leaf leave-mode {
        type identityref {
            base igmp-leave-mode;
        }
        default fast;

        description
           "Specify the igmp leave mode for multicast service.";
    }

    leaf linger-timeout {
        type uint32 {
            range "10..300";
        }
        units "seconds";
        default "120";

        description
           "Specify the amount of time that a multicast filter is
            in place after the last host has left. In essence it
            configures how long to filter the traffic after all
          the hosts have left the group. This can tune the system
          as to how long the router has to remove the stream from
            the subnet.";
    }

    leaf server-topology {
        type identityref {
            base server-topology-mode;
        }
        default centralized;

        description
            "Specify the server topology mode.";
    }

    leaf priority {
        type uint32 {
```


Copyright© 2022 Ciena® Corporation

```
                range "0..7";
            }
            default 7;

            description
                "Specifies the priority of igmp packets for
                this multicast service.";
        }

        leaf querier-compatibility-mode {
            type identityref {
                base igmp-compatibility-mode;
            }
            default v3;

            description
                "Specifies the query engine compatibility mode.";
        }

        leaf query-delay {
            type uint32 {
                range "1..100";
            }
            units "deci-seconds(Tenths of a sec)";
            default "10";

            description
                "Specifies the time to wait before sending the next
                group specific query downstream.";
        }

        leaf query-engine-state {
            type enumeration {
                enum on {
                    description
                        "Query engine on.";
                }
                enum off {
                    description
                        "Query engine off.";
                }
            }
            default off;

            description
                "Specifies the query engine state for a given
                multicast service. ";
        }

        leaf query-interval {
            type uint32 {
                range "10..31744";
            }
            units "seconds";
            default "125";

            description
                "The interval period between each general query
                sent to the router. This parameter specifies how
                often to repeat the group specific query.
```


Copyright© 2022 Ciena® Corporation

```
            Default value is 125 seconds.  Note that the actual
            rate may be slower when the switch becomes busy,
                 so this parameter should be thought of as a
                     minimum time interval.";
    }

    leaf query-source-ip-address {
        type inet:ipv4-address;

        description
         "This represents the source IP address for the multicast
            service.";
    }


    leaf query-response-interval {
        type uint32 {
            range "10..31744";
        }
        units "deci-seconds(Tenths of a sec)";
        default "100";

        description
            "The amount of time to receive a membership
            report after a query message is sent.
            This value goes directly into the query packet.";
    }

    leaf rapid-recovery-mode {
        type enumeration {
            enum on {
                description
                   "Generates and accepts query packets with zero
                     source ip address.";
            }
            enum off {
                description
                 "Does not generate and accept query packets with
                    zero source ip address.";
            }
        }
        default off;

        description
          "Specifies whether the rapid recovery mode is set to on
            or off.";
    }

    leaf robustness {
        type uint32 {
            range "1..7";
        }
        default 2;

        description
            "Specify the number of replies to send for
            each multicast group.";
    }

    container router-group-address {
```


Copyright© 2022 Ciena® Corporation

```
                    description
                       "Specify a range of multicast group addresses that are
                       originated from router. If it is undefined, all leave,
                                  join messages are forwarded to router. If
it is defined,
                                   leave/join messages are filtered.";

                    leaf start-ip-address {
                        type inet:ipv4-address;

                        description
                            "Specify a start multicast address.";
                    }

                    leaf end-ip-address {
                        type inet:ipv4-address;

                        description
                            "Specify an end multicast address.";
                    }
                }

                leaf router-query-interval {
                    type uint32 {
                        range "10..31744";
                    }
                    units "seconds";
                    default "250";

                    description
                      "Specify the router query interval for multicast service.";
                }
            }

            container static-groups {
                description
                    "Specify the static multicast groups.";
                leaf-list address {
                    type inet:ipv4-address;

                    description
                       "Specifies the ip Address of static multicast group.";
                }
            }

            container static-group-exclude-interface {
                description
                    "Specify the list of interfaces excluded from a static
multicast group.";
                leaf-list interface {
                    type mef-fp:fp-ref;

                    description
                        "Specifies the interface to be excluded from staic
                        multicast group.";
                }
            }
        }
    }
```


Copyright© 2022 Ciena® Corporation

```
container multicast-services-state {
    config false;
    description
        "multicast services state data";

    list instance-info {
        key "fd-name";

        description
            "Specifies a list of multicast service instances.";

        uses fd-ref;

        leaf oper-state {
            type state;

            description
                "Represents the operational state of multicast-service on
                 a instance.";
        }

        container igmp-snooping-state {

            description
               "Specifies the operational parameters for igmp snooping.";

            leaf operational-state {
                type state;

                description
                 "Represents the operational state of igmp snooping on the
                    multicast service";
            }

            leaf router-source-mac-address {
                type yang:mac-address;

                description
                    "Specifies the router's MAC address.";
            }

            leaf router-source-ip-address {
                type inet:ipv4-address;

                description
                    "Specifies the router's IP address.";
            }

            leaf router-interface {
                type mef-fp:fp-ref;

                description
                    "Specifies the interface from where router queries
                    are coming.";
            }

            leaf report-send-interface {
                type mef-fp:fp-ref;

                description
```


Copyright© 2022 Ciena® Corporation

```
                "Specifies the interface to where IGMP reports are sent.
                   Not always the same as the router interface.";
            }

            leaf router-compatibility-mode {
                type identityref {
                    base igmp-compatibility-mode;
                }
                description
                    "Specifies the IGMP version of the router.";
            }

            leaf report-send-interface-is-meshvc {
                type enumeration {
                    enum yes {
                        description
                            "Interface is mesh vc.";
                    }
                    enum no {
                        description
                            "Interface is not mesh vc.";
                    }
                    enum undefined {
                        description
                            "Undefined.";
                    }
                }

                description
                  "Specifies whether the current report send interface is
                    a meshvc or not.";
            }
        }

        list group-information {
            key "address";

            description
                "Specifies the list of all multicast groups.";

            leaf address {
                type inet:ipv4-address;

                description
                    "The multicast group ip address.";
            }

            leaf state {
                type identityref {
                    base multicast-service-state;
                }

                description
                    "The current state of the multicast service.";
            }

            leaf type {
                type identityref {
                    base multicast-service-type;
                }
```


Copyright© 2022 Ciena® Corporation

```
    description
        "The type of the multicast service.";
}

leaf source {
    type identityref {
        base multicast-traffic-source;
    }

    description
        "The source for this multicast group traffic.";
}

leaf member-count {
    type uint32;

    description
        "Number of member interfaces of multicast group.";
}

leaf up-time {
    type yang:timeticks;

    description
      "Specify the time when a multicast group is discovered,
        in seconds.";
}

leaf group-compatibility-mode {
    type identityref {
        base igmp-compatibility-mode;
    }

    description
     "Represents the group compatibility mode for group based
     on the lowest version of igmp membership report recieved
        for this multicast group.";
}

leaf group-timer {
    type yang:timeticks;

    description
       "Specify the time left (seconds) before we decide that
        there are no more members of a group.";
}

leaf igmpv1-host-present-timer {
    type yang:timeticks;

    description
      "Specify the timeout period of an igmpv1 host. It is the
         remaining time (seconds) from now, before the group
     compatibility mode switches back to V2/V3 on the group.";
}

leaf igmpv2-host-present-timer {
    type yang:timeticks;
```


Copyright© 2022 Ciena® Corporation

```
                description
                 "Specify the timeout period of an igmpv2 host. It is the
                    remaining time (seconds) from now, before the group
                  compatibility mode switches back to V3 on this group.";
            }

            leaf source-filter-mode  {
                type identityref {
                    base filter-mode;
                }

                description
                 "The igmp v3 source address filter mode for the multicast
                    group.";
            }

            leaf number-of-source-addresses {
                type uint32;

                description
                  "Number of source addresses for this multicast group.";
            }

            list group-source-records {
                key "source-ip-address";
                description
                    "Specify dynamic multicast source information.";

                leaf source-ip-address {
                    type inet:ipv4-address;

                    description
                        "The ip address of source record for the igmp v3
                        multicast group.";
                }

                leaf source-timer {
                    type yang:timeticks;

                    description
                    "Specify the time left to delete source address record
                        from the source table, in seconds";
                }
            }
        }

        list group-members {
            key "address interface";
            description
                "Specify all interface members of multicast group.";

            leaf address {
                type inet:ipv4-address;

                description
                    "The multicast group ip address.";
            }

            leaf interface {
                type mef-fp:fp-ref;
```

```
            description
                "Represents the multicast group member";
        }
    }

    container packet-statistics {
        description
            "Specifies the packet statistics parameters.";

        leaf static-group-count {
            type uint16;

            description
                "The total number of static multicast groups.";
        }

        leaf dynamic-group-count {
            type uint16;

            description
                "The total number of dynamic mcast groups.";
        }

        leaf router-group-count {
            type uint16;

            description
                "Active multicast groups of the router which require
                query reply.";
        }

        leaf linger-group-count {
            type uint16;

            description
                "The total number of groups in linger state.";
        }

        leaf total-group-members {
            type uint16;

            description
                "The total number of group members for all groups in
                a multicast service.";
        }

        leaf join-messages-received {
            type yang:counter32;

            description
                "Number of join messages received.";
        }

        leaf leave-messages-received {
            type yang:counter32;

            description
                "Number of leave messages received.";
        }
```


Copyright© 2022 Ciena® Corporation

```
leaf version3-membership-reports-received {
    type yang:counter32;

    description
        "Number of IGMP V3 membership reports recieved.";
}

leaf version3-membership-IsIn-reports-received {
    type yang:counter32;

    description
        "Number of V3 membership IS_IN reports recieved.";
}

leaf version3-membership-IsEx-reports-received {
    type yang:counter32;

    description
        "Number of V3 membership IS_EX reports recieved.";
}

leaf version3-membership-ToIn-reports-received {
    type yang:counter32;

    description
        "Number of V3 membership TO_IN reports recieved.";
}

leaf version3-membership-ToEx-reports-received {
    type yang:counter32;

    description
        "Number of V3 membership TO_EX reports recieved.";
}

leaf version3-membership-allow-reports-received {
    type yang:counter32;

    description
        "Number of V3 membership Allow reports recieved.";
}

leaf version3-membership-block-reports-received {
    type yang:counter32;

    description
        "Number of V3 membership Block reports recieved.";
}

leaf query-messages-received {
    type yang:counter32;

    description
        "Number of queries received.";
}

leaf query-messages-discarded {
    type yang:counter32;
```


Copyright© 2022 Ciena® Corporation

```
        description
            "Number of queries discarded.";
    }

    leaf router-query-timeouts {
        type yang:counter32;

        description
            "Number of timeouts waiting for the router to
            send query.";
    }

    leaf query-reply-timeouts {
        type yang:counter32;

        description
            "Specifies the number of query reply timeouts.
            Timeouts sending reports to the router.";
    }

    leaf join-messages-discarded-on-router-interface {
        type yang:counter32;

        description
         "Number of messages discarded by the router. This happens
            when join comes in from a router interface.";
    }

    leaf join-messages-discarded-because-no-router-interface {
        type yang:counter32;

        description
          "Number of messages discarded by the host. This happens
            when a join/leave comes in from a interface, but
            no router interface is established.";
    }

    leaf bad-checksum-messages-received {
        type yang:counter32;

        description
            "Number of packets received with bad
            IP checksum.";
    }

    leaf l2-l3-mismatch-packets-received {
        type yang:counter32;

        description
            "Number of IGMP Layer2 / Layer 3 mismatch packets
            received.";
    }

    leaf unknown-packet-type-received {
        type yang:counter32;

        description
            "Number of unknown (unknown type ) IGMP
            packets received.";
    }
```


Copyright© 2022 Ciena® Corporation

```
                leaf resource-exceeded {
                    type yang:counter32;

                    description
                        "Specifies the number of resource exceeded errors.";
                }

                leaf query-with-source-ip-zero-discarded {
                    type yang:counter32;

                    description
                        "Specifies the number of queries discarded with a
                        zero source IP because of rapid-recovery off.";
                }

                leaf compatibility-mode-discards {
                    type yang:counter32;

                    description
                     "For products that support IGMPv3, this object specifies
                        the total number of frames discarded because of
                        compatibility mode mismatches. Products that do not
                        support IGMPv3 will not return this object.";
                }
            }
        }
    }

    //
    // RPCs
    //

    rpc clear-statistics {
        description
            "Administratively clears the statistics of a group.";

        input {
            uses fd-ref;
        }
    }

    rpc debug-event-logging {
        description
            "Set or unset debug events.";

        input {
            leaf event-logging {
                type enumeration {
                    enum on {
                        description
                            "Debug events on.";
                    }
                    enum off {
                        description
                            "Debug events off.";
                    }
                }

                description
```


Copyright© 2022 Ciena® Corporation

```
                    "Command to turn debugs on or off.";
            }
        }
    }

    //
    // Notifications
    //

    /* TODO: Need to check whether there can be any. */
} /* end of module */
```


Copyright© 2022 Ciena® Corporation

ciena-ietf-ip-deviation-sds.yang

```
module ciena-ietf-ip-deviation-sds {
  namespace
    "http://www.ciena.com/ns/yang/ciena-ietf-ip-deviation-sds";

  prefix ciena-ietf-ip-deviation-sds;

  import ietf-interfaces {
    prefix if;
  }

  import ietf-ip {
    prefix ip;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module defines YANG deviations between
     Ciena SAOS SDS equipment and the IETF IP YANG
     implementation.

     Copyright (c) 2017 Ciena Corporation.  All rights
     reserved.

     All information contained herein is, and remains
     the property of Ciena Corporation. Dissemination of this
     information or reproduction of this material is strictly
     forbidden unless prior written permission is obtained from
     Ciena Corporation.";

  revision 2017-06-07 {
    description
      "Initial version";
    reference
      "009-3273-007 (39XX 51XX SAOS 6.16 Administration & Security)
       Standard Revision A

       380-1874-301_(8700_SAOS_8.4_AdministrationSecurity)RevA
       8700 Packetwave Platform Administration and Security, SAOS 8.4,
       380-1874-301, 8700, Rev A";
  }

  /*
   * Not supported
   */

  // Operational data for IP interfaces will not be supported in the current
  // release of this SAOS SDS product.
```


Copyright© 2022 Ciena® Corporation

```
  //
  // Remove the IPv4 operational container
  deviation /if:interfaces-state/if:interface/ip:ipv4 {
    deviate not-supported;
  }

  // Operational data for IP interfaces will not be supported in the current
  // release of this SAOS SDS product.
  //
  // Remove the IPv6 operational container
  deviation /if:interfaces-state/if:interface/ip:ipv6 {
    deviate not-supported;
  }

  // Currently don't support rw neighbor information
  deviation /if:interfaces/if:interface/ip:ipv6/ip:neighbor {
    deviate not-supported;
  }

  // Currently don't dup-addr-detect-transmits rw information
 deviation /if:interfaces/if:interface/ip:ipv6/ip:dup-addr-detect-transmits
{
    deviate not-supported;
  }

  // Currently don't autoconf rw information
  deviation /if:interfaces/if:interface/ip:ipv6/ip:autoconf {
    deviate not-supported;
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-ietf-snmp-deviation.yang

```
module ciena-ietf-snmp-deviation {
  namespace "http://ciena.com/ns/yang/ciena-ietf-snmp-deviation";
  prefix snmp-dev;

  import ietf-snmp {
    prefix snmp;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
    E-mail:  yang@ciena.com
    Postal:  7035 Ridge Road
             Hanover, Maryland 21076
             U.S.A.
    Phone:   +1 800-921-1144
    Fax:     +1 410-694-5750";

  description
    "This YANG module defines Ciena deviations from the ietf-snmp.yang module.

    Copyright (c) 2016 Ciena Corporation.  All rights reserved.

          All information contained herein is, and remains
          the property of Ciena Corporation. Dissemination of this
          information or reproduction of this material is strictly
          forbidden unless prior written permission is obtained from
          Ciena Corporation.";

  revision "2016-09-22"
  {
    description
      "Initial version";
    reference
      "RFC 6020: YANG - A Data Modeling Language for
       the Network Configuration Protocol (NETCONF).
       No specific reference; standard not available.";
  }

  // SAOS will use its own version of the target container
  deviation /snmp:snmp/snmp:target {
    deviate not-supported;
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-ietf-syslog-deviation.yang

```
module ciena-ietf-syslog-deviation {
  namespace "http://ciena.com/ns/yang/ciena-ietf-syslog-deviation";
  prefix ciena-ietf-syslog-deviation;

  import ietf-syslog {
    prefix syslog;
  }
  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This module contains the Ciena proprietary YANG deviations
     of the standard ietf-syslog YANG module.

     Copyright (c) 2016 Ciena Corporation.  All rights reserved.

     All information contained herein is, and remains
     the property of Ciena Corporation. Dissemination of this
     information or reproduction of this material is strictly
     forbidden unless prior written permission is obtained from
     Ciena Corporation.";

  revision "2016-03-01" {
    description
      "Initial version";
    reference
      "";
  }
//  Yang node /syslog:syslog/syslog:log-actions/syslog:console
//  is removed via feature-disable of feature ietf-syslog:console-action

//  Yang node /syslog:syslog/syslog:log-actions/syslog:buffer
//  is removed via feature-disable of feature ietf-syslog:buffer-action

  deviation /syslog:syslog/syslog:log-actions/syslog:file {
    deviate not-supported;
  }

  deviation /syslog:syslog/syslog:log-actions/syslog:remote/
syslog:destination {
    deviate not-supported;
  }

//  Yang node /syslog:syslog/syslog:log-actions/syslog:terminal
//  is removed via feature-disable of feature ietf-syslog:terminal-action

//  Yang node /syslog:syslog/syslog:log-actions/syslog:session
//  is removed via feature-disable of feature ietf-syslog:session-action
}
```

Copyright© 2022 Ciena® Corporation

ciena-ietf-system-deviation.yang

```
module ciena-ietf-system-deviation {
  namespace "http://www.ciena.com/ns/yang/ciena-ietf-system-deviation";
  prefix sys-dev;

  import ietf-system {
    prefix sys;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module defines Ciena's deviations from the standard
     system model.

     Copyright (c) 2016 Ciena Corporation.  All rights reserved.

     All information contained herein is, and remains the property of
     Ciena Corporation. Dissemination of this information or
     reproduction of this material is strictly forbidden unless prior
     written permission is obtained from Ciena Corporation.";

  revision 2017-06-06 {
    description
      "Initial version";
    reference
      "RFC 6020: YANG - A Data Modeling Language for
       the Network Configuration Protocol (NETCONF).
       No specific reference; standard not available.";
  }

  //
  // Configuration deviations
  //

  // SAOS allows characters for a hostname not allowed by
  // a "domain-name"
  deviation /sys:system/sys:hostname {
    deviate replace {
      type string {
        length "2..63";
      }
    }
  }

  // SAOS will use its version of the NTP model
  // Yang node /sys:system/sys:ntp
  // is removed via feature-disable of feature ietf-system:ntp
```


Copyright© 2022 Ciena® Corporation

```
  // SAOS will use its version of system authentication Yang node
  // /sys:system/sys:authentication is removed via feature-disable of
  // feature ietf-system:authentication

  // SAOS will use its version of the DNS resolver
  deviation /sys:system/sys:dns-resolver {
    deviate not-supported;
  }

  // SAOS will use its version of the RADIUS client
  // Yang node /sys:system/sys:radius
  // is removed via feature-disable of feature ietf-system:radius

  //
  // Operational deviations
  //

  //
  // RCP deviations
  //

  // SAOS does not have the concept of shutting down
  deviation /sys:system-shutdown {
    deviate not-supported;
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-ietf-twamp-deviation.yang

```
module ciena-ietf-twamp-deviation {

    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-ietf-twamp-dev";
    prefix "ciena-ietf-twamp-dev";

    import ciena-ietf-twamp { prefix ciena-ietf-twamp; }

    revision 2015-04-21 {
        description "Initial revision.";
    }

    //
    //  Changes to supported config data.
    //

    //
    //  Config data that is not supported in SAOS 6.14
    //
    deviation /ciena-ietf-twamp:twamp/ciena-ietf-twamp:twampServer/ciena-
ietf-twamp:count {
        deviate not-supported;
    }

    deviation /ciena-ietf-twamp:twamp/ciena-ietf-twamp:twampServer/ciena-
ietf-twamp:maxCount {
        deviate not-supported;
    }

    deviation /ciena-ietf-twamp:twamp/ciena-ietf-twamp:twampServer/ciena-
ietf-twamp:modes {
        deviate not-supported;
    }

    deviation /ciena-ietf-twamp:twamp/ciena-ietf-twamp:twampServer/ciena-
ietf-twamp:keyChain {
        deviate not-supported;
    }

    //
    //  Read-only data is not supported in SAOS 6.14
    //
    deviation /ciena-ietf-twamp:twamp/ciena-ietf-twamp:twampServer/ciena-
ietf-twamp:salt {
        deviate not-supported;
    }

    deviation /ciena-ietf-twamp:twamp/ciena-ietf-twamp:twampServer/ciena-
ietf-twamp:serverIv {
        deviate not-supported;
    }

    deviation /ciena-ietf-twamp:twamp/ciena-ietf-twamp:twampServer/ciena-
ietf-twamp:challenge {
        deviate not-supported;
    }

    deviation /ciena-ietf-twamp:twamp/ciena-ietf-twamp:twampServer/ciena-
ietf-twamp:twampServerCtrlConnection {
```


Copyright© 2022 Ciena® Corporation

```
        deviate not-supported;
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-ietf-twamp.yang

```
module ciena-ietf-twamp {
   namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-ietf-twamp";
    //namespace need to be assigned by IANA
   prefix "twamp";

   import ietf-inet-types {
     prefix inet;
   }

   organization
     "IETF IPPM (IP Performance Metrics) Working Group";

   contact
     "draft-cmzrjp-ippm-twamp-yang@tools.ietf.org";

   description "TWAMP Data Model";

   revision "2015-03-06" {
     description "Initial version. RFC5357 is covered.
       RFC5618, RFC5938 and RFC6038 are not covered.";
   }

   feature controlClient {
     description "This feature relates to the device functions as the
       TWAMP Control-Client.";
   }

   feature server {
     description "This feature relates to the device functions as the
       TWAMP Server.";
   }

   feature sessionSender {
     description "This feature relates to the device functions as the
       TWAMP Session-Sender.";
   }

   feature sessionReflector {
     description "This feature relates to the device functions as the
       TWAMP Session-Reflector.";
   }

   grouping maintenanceStatistics {
     leaf sentPackets {
       config "false";
       type uint32;
     }
     leaf rcvPackets {
       config "false";
       type uint32;
     }
     leaf lastSentSeq {
       config "false";
       type uint32;
     }
     leaf lastRcvSeq {
       config "false";
       type uint32;
```


Copyright© 2022 Ciena® Corporation

```
    }
  }

  container twamp {
    container twampClient {
      presence "twampClient";
      if-feature controlClient;
      leaf clientAdminState {
        mandatory "true";
        type boolean;
        description "Indicates whether this device is allowed to run
          TWAMP to initiate control/test sessions";
      }

      list modePreferenceChain {
        key "priority";
        unique "mode";

        leaf priority {
          type uint16;
        }

        leaf mode {
          type enumeration {
            enum unauthenticated {
              value "1";
            }
            enum authenticated {
              value "2";
              }
            enum encrypted {
              value "4";
            }
            enum unauthtestencrpytcontrol {
              value "8";
            }
            enum individualsessioncontrol {
              value "16";
              }
            enum reflectoctets {
              value "32";
            }
            enum symmetricalsize {
              value "64";
            }
          }
        }
      }

      list keyChain {
        key "keyId";
        leaf keyId {
          type string {
          }
        }
        leaf secretKey {
          type string;
        }
      }
```


Copyright© 2022 Ciena® Corporation

```
list twampClientCtrlConnection {
  key "ctrlConnectionName";
  leaf ctrlConnectionName {
    type "string";
    description "A unique name used as a key to identify this
      individual TWAMP control connection on the
      Control-Client device.";
  }
  leaf clientIp {
    type inet:ip-address;
  }
  leaf serverIp {
    config "true";
    type inet:ip-address;
  }
  leaf serverTcpPort {
    type inet:port-number;
  }
  leaf dscp{
    type inet:dscp;
    description "The DSCP value to be placed in the IP header
      of the TWAMP TCP Control packets generated
      by the Control-Client";
  }
  leaf keyId {
    type string {
    }
  }
  leaf dkLen {
    type uint32;
  }
  leaf clientTcpPort {
    config "false";
    type inet:port-number;
  }
  leaf serverStartTime {
    config "false";
    type uint64;
  }
  leaf ctrlConnectionState {
    config "false";
    type enumeration {
      enum active {
        description "Control session is active.";
      }
      enum idle {
        description "Control session is idle.";
      }
    }
  }
  leaf selectedMode {
    config "false";
    type enumeration {
      enum unauthenticated {
        value "1";
      }
      enum authenticated {
        value "2";
        }
      enum encrypted {
```


Copyright© 2022 Ciena® Corporation

```
            value "4";
          }
          enum unauthtestencrpytcontrol {
            value "8";
          }
          enum individualsessioncontrol {
            value "16";
            }
          enum reflectoctets {
            value "32";
          }
          enum symmetricalsize {
            value "64";
          }
        }
      }
      leaf token {
        config "false";
        type string {
          length "1..64";
        }
        description "64 octets, containing the concatenation of a
          16-octet challenge, a 16-octet AES Session-key used
          for encryption, and a 32-octet HMAC-SHA1 Session-key
          used for authentication";
      }
      leaf clientIv{
        config "false";
        type string {
          length "1..16";
        }
        description "16 octets, Client-IV is generated randomly
          by the Control-Client.";
      }

      list twampSessionRequest {
        key "testSessionName";
        leaf testSessionName {
          type "string";
        }
        leaf senderIp {
          type inet:ip-address;
        }
        leaf senderUdpPort {
          type inet:port-number;
        }
        leaf reflectorIp {
          type inet:ip-address;
        }
        leaf reflectorUdpPort {
          type inet:port-number;
        }
        leaf timeout {
          type uint64;
          description "The time Session-Reflector MUST wait after
              receiving a Stop-Session message";
        }
        leaf paddingLength {
          type uint32{
            range "64..1500";
```


Copyright© 2022 Ciena® Corporation

```
            }
            description "The number of bytes of padding that should
                be added to the UDP test packets generated by the
                sender.";
          }
          leaf startTime {
            type uint64;
          }
          leaf repeat  {
            type boolean;
          }
          leaf repeatInterval  {
            type uint32;
            when "../repeat='true'";
            description "Repeat interval (in minutes)";
          }
          leaf pmIndex {
            type uint16;
            description "Numerical index value of a Registered
                Metric in the Performance Metric Registry";
          }
          leaf testSessionState {
            config "false";
            type enumeration {
              enum ok {
                value 0;
                description "Test session is accepted.";
              }
              enum failed {
                value 1;
                description "Failure, reason unspecified
                    (catch-all).";
              }
              enum internalError {
                value 2;
                description "Internal error.";
              }
              enum notSupported {
                value 3;
                description "Some aspect of request is not
                    supported.";
              }
              enum permanentResLimit {
                value 4;
                description "Cannot perform request due to
                    permanent resource limitations.";
              }
              enum tempResLimit {
                value 5;
                description "Cannot perform request due to
                    temporary resource limitations.";
              }
            }
          }
          leaf sid{
            config "false";
            type string;
          }
        }
      }
```


Copyright© 2022 Ciena® Corporation

```
}

container twampServer {
  presence "twampServer";
  if-feature server;
  leaf serverAdminState{
    type boolean;
    mandatory "true";
    description "Indicates whether this device is allowed to run
      TWAMP to respond to control/test sessions";
  }
  leaf serverTcpPort {
    type inet:port-number;
    default "862";
  }
  leaf servwait {
    type uint32 {
      range 1..604800;
    }
    default 900;
    description "SERVWAIT (TWAMP Control (TCP) session timeout),
      default value is 900";
  }
  leaf dscp {
    type inet:dscp;
    description "The DSCP value to be placed in the IP header
      of the TWAMP TCP Control packets generated by the Server";
  }
  leaf count {
    type uint32 {
      range 1024..4294967295;
    }
  }
  leaf maxCount {
    type uint32 {
      range 1024..4294967295;
    }
    default 32768;
  }
  leaf modes {
    type bits {
      bit unauthenticated {
        position 0;
      }
      bit authenticated {
        position 1;
      }
      bit encrypted {
        position 2;
      }
      bit unauthtestencryptcontrol {
        position 3;
      }
      bit individualsessioncontrol {
        position 4;
      }
      bit reflectoctets {
        position 5;
      }
      bit symmetricalsize {
```


Copyright© 2022 Ciena® Corporation

```
      position 6;
    }
  }
}
leaf salt{
  config "false";
  type string {
  }
    description "Salt MUST be generated pseudo-randomly";
}
leaf serverIv {
  config "false";
  type string {
    length "1..16";
  }
  description "16 octets, Server-IV is generated randomly
    by the Control-Client.";
}
leaf challenge {
  config "false";
  type string {
  }
  description "Challenge is a random sequence of octets
    generated by the Server";
}

list keyChain {
  key "keyId";
  leaf keyId {
    type string {
    }
  }
  leaf secretKey {
    type string;
  }
}

list twampServerCtrlConnection {
  key "clientIp clientTcpPort serverIp serverTcpPort";
  config "false";
  leaf clientIp {
    type inet:ip-address;
  }
  leaf clientTcpPort {
    type inet:port-number;
  }
  leaf serverIp {
    type inet:ip-address;
  }
  leaf serverTcpPort {
    type inet:port-number;
  }
  leaf serverCtrlConnectionState {
    type enumeration {
      enum "active";
      enum "servwait";
    }
  }
  leaf dscp {
    type inet:dscp;
```


Copyright© 2022 Ciena® Corporation

```
    description "The DSCP value used in the header of the TCP
      control packets sent by the Server for this control
      connection. This will usually be the same value as is
      configured for twampServer:dscp under the twampServer.
      However, in the event that the user re-configures
      twampServer:dscp after this control connection is already
      in progress, this read-only value will show the actual
      dscp value in use by this control connection.";
  }
  leaf selectedMode {
    type enumeration {
      enum unauthenticated {
        value "1";
      }
      enum authenticated {
        value "2";
        }
      enum encrypted {
        value "4";
      }
      enum unauthtestencrpytcontrol {
        value "8";
      }
      enum individualsessioncontrol {
        value "16";
        }
      enum reflectoctets {
        value "32";
      }
      enum symmetricalsize {
        value "64";
      }
    }
    description "The mode that was chosen for this control
      connection as set in the Mode field of the Set-Up-Response
      message.";
  }
  leaf keyId {
    type string {
    }
    description "The keyId value that is in use by this control
      connection.";
  }
  leaf dkLen {
    type uint32;
    description "The dkLen value that is in use by this control
      connection. This will usually be the same value as is
      configured under twampServer. In the event that the user
      re-configured twampServer:dkLen after this control
      connection is already in progress, this read-only value
      will show the actual dkLen that is in use for this
      control connection.";
  }
  leaf count {
    type uint32 {
      range 1024..4294967295;
    }
    description "The count value that is in use by this control
      connection. This will usually be the same value as is
      configured under twampServer. However, in the event that
```


Copyright© 2022 Ciena® Corporation

```
          the user re-configured twampServer:count after this control
          connection is already in progress, this read-only value
          will show the actual count that is in use for this
          control connection.";
      }
      leaf maxCount {
        type uint32 {
          range 1024..4294967295;
        }
        description "The maxCount value that is in use by this
          control connection. This will usually be the same value
          as is configured under twampServer. However, in the event
          that the user re-configured twampServer:maxCount after
          this control connection is already in progress, this
          read-only value will show the actual maxCount that is
          in use for this control connection.";
      }
    }
  }

  container twampSessionSender {
    if-feature sessionSender;
    list twampSenderTestSession {
      key "testSessionName";
      leaf testSessionName {
        type string;
        description "A unique name for this test session to be used
          as a key for this test session by the Session-Sender
          logical entity.";
      }
      leaf ctrlConnectionName {
        config "false";
          type "string";
          description "The name of the parent control connection
            that is responsible for negotiating this test session.";
      }
      leaf dscp {
        type inet:dscp;
        description "The DSCP value to be placed in the header of
          TWAMP UDP test packets generated by the sender.";
      }
      leaf dot1dPriority {
        type uint8 {
          range "0..7";
        }
      }
      leaf fillMode {
        type enumeration {
          enum zero;
          enum random;
        }
        default zero;
      }
      leaf numberOfPackets {
        type uint32;
        description "The overall number of UDP test packets to be
          transmitted by the sender for this test session.";
      }
      choice packetDistribution {
        case fixed {
```


Copyright© 2022 Ciena® Corporation

```
            leaf fixedInterval {
              type uint32;
            }
            leaf fixedIntervalUnits  {
              type enumeration {
                enum seconds;
                enum milliseconds;
                enum microseconds;
                enum nanoseconds;
              }
            }
          }
          case poisson {
            leaf lambda {
              type uint32;
            }
            leaf lambdaUnits {
              type uint32;
            }
            leaf maxInterval {
              type uint32;
            }
            leaf truncationPointUnits {
              type enumeration {
                enum seconds;
                enum milliseconds;
                enum microseconds;
                enum nanoseconds;
              }
            }
          }
        }
        leaf senderSessionState {
          config "false";
          type enumeration {
            enum setup {
              description "Test session is active.";
            }
            enum failure {
              description "Test session is idle.";
            }
          }
        }
        uses maintenanceStatistics;
      }
    }

    container twampSessionReflector {
      if-feature sessionReflector;
      leaf refwait {
        config "true";
        type uint32 {
          range 1..604800;
        }
        default 900;
        description "REFWAIT(TWAMP test session timeout),
          the default value is 900";
      }

      list twampReflectorTestSession {
```


Copyright© 2022 Ciena® Corporation

```
          key "senderIp senderUdpPort reflectorIp reflectorUdpPort";
          config "false";
          leaf sid {
            type string;
          }
          leaf senderIp {
            type inet:ip-address;
          }
          leaf senderUdpPort {
            type inet:port-number;
          }
          leaf reflectorIp {
            type inet:ip-address;
          }
          leaf reflectorUdpPort {
            type inet:port-number;
          }
          leaf parentConnectionClientIp {
            type inet:ip-address;
          }
          leaf parentConnectionClientTcpPort {
            type inet:port-number;
          }
          leaf parentConnectionServerIp {
            type inet:ip-address;
          }
          leaf parentConnectionServerTcpPort {
            type inet:port-number;
          }
          leaf dscp {
            type inet:dscp;
            description "The DSCP value placed in the header of TWAMP
              UDP test packets generated by the Session-Sender.";
          }
          uses maintenanceStatistics;
        }
      }
    }
  }
```


Copyright© 2022 Ciena® Corporation

ciena-ip-acl-sds.yang

```
module ciena-ip-acl-sds {
  namespace "http://www.ciena.com/ns/yang/ciena-sds-ip-acl";
  prefix "ciena-ip-acl-sds";

  import ietf-inet-types {
    prefix "inet";
  }

  import ietf-yang-types {
    prefix yang;
  }

  import ciena-types {
    prefix ciena;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
    E-mail:  yang@ciena.com
    Postal:  7035 Ridge Road
             Hanover, Maryland 21076
             U.S.A.
    Phone:   +1 800-921-1144
    Fax:     +1 410-694-5750";

  description
    "This YANG module defines the nodes for sw ip-acl
     for the NETCONF protocol for 6x.

     Single Blade.

    Copyright (c) 2016 Ciena Corporation.  All rights
         reserved.

         All information contained herein is, and remains
         the property of Ciena Corporation. Dissemination of this
         information or reproduction of this material is strictly
         forbidden unless prior written permission is obtained from
         Ciena Corporation.";

  revision "2018-08-28"
  {
      description
          "Updated description of src-ip, dest-ip leaf's
           w.r.t the non-supported configuration of these leafs.";
      reference
          "RFC 6020: YANG - A Data Modeling Language for
              the Network Configuration Protocol (NETCONF).
              No specific reference; standard not available.";
  }

  revision "2017-01-31"
  {
      description
          "Initial version";
```


Copyright© 2022 Ciena® Corporation

```
    reference
        "RFC 6020: YANG - A Data Modeling Language for
            the Network Configuration Protocol (NETCONF).
            No specific reference; standard not available.";
}


/*
 * Groupings
 */

grouping ciena-ip-acl-statistics {
  description
    "ip-acl counters.";

  leaf rx-frames-hit {
    type yang:counter64;

    description
      "Rx frames hit.";
  }

  leaf rx-frames-bad-port {
    type yang:counter64;

    description
      "Rx frames with bad port.";
  }

  leaf rx-frames-bad-dscp {
    type yang:counter64;

    description
      "Rx frames with bad dscp.";
  }

  leaf rx-frames-bad-proto {
    type yang:counter64;

    description
      "Rx frames with bad protocol.";
  }
}

/*
 * Configuration data nodes
 */

container ciena-ip-acl {
  description
    "Configuration data for sw ip-acl.";

  container ip-acl-global {
    description
      "Global config for sw ip-acl.";

    leaf admin-state {
      type ciena:admin-state;

      default disabled;
```


Copyright© 2022 Ciena® Corporation

```
        description
          "The global admin state of cpu-rate-limiting.";
      }
    }

    list ip-acl-entry {
      key "src-ip dest-ip";

      description
        "The list of configured ip-acl entries on the device.";

      leaf src-ip {
        type inet:ip-prefix;

        description
          "Source IPv4/IPv6 Address.
           Multicast or Loopback Address shall be rejected.
           IPv6 Address are only accepted in Compressed format -
2001:db8:0000:00::0/64 shall be rejected, 2001:db8::/64 shall only be
allowed.";
      }

      leaf dest-ip {
        type inet:ip-prefix;

        description
          "Destination IPv4/IPv6 Address.
           Multicast or Loopback Address shall be rejected.
           IPv6 Address are only accepted in Compressed format -
2001:db8:0000:00::0/64 shall be rejected, 2001:db8::/64 shall only be
allowed.";
      }

      // At CLI this is a list of port ranges, but it is
      // held internally in SAOS as a bit mask
      leaf port-ranges {
        type string;
        default "all";
        description
          "list of ports or port ranges separated by commas.
           E.g. 1-2,5,7-8.";
      }

      // At CLI this is a list of dscp ranges, but it is
      // held internally in SAOS as a bit mask
      leaf dscp-ranges {
        type string;
        default "all";
        description
          "List of dscp values or dscp value ranges separated by commas.
           E.g. 0-1,5,60-63";
      }

      // At CLI this is a list of protocols, but it is
      // held internally in SAOS as a bit mask
      // The value "all" is mutually exclusive to all
      // other protocols.
      choice protocol {
        description
```


Copyright© 2022 Ciena® Corporation

```
        "Match for protocol can be all OR a list of protocols.";
      default all;
      case all {
        leaf all {
          type enumeration {
            enum all;
          }
          default all;
          description
            "all protocols";
        }
      }

      case protocols {
        leaf protocol-mask {
          type bits {
            bit icmp {
              description "icmp protocol";
            }
            bit tcp {
              description "tcp protocol";
            }
            bit udp {
              description "udp protocol";
            }
          }
        description
          "protocol mask";
        }
      }
    }
  }
}

/*
 * Operational state data nodes
 */

container ciena-ip-acl-state {
  config false;

  description
    "Operational data for sw ip-acl.";

  container ip-acl-global {
    description
      "Global operational data for sw ip-acl.";

    leaf oper-state {
      type ciena:oper-state;
      description
        "The global oper state of sw ip-acl.";
    }

    leaf entries-in-use {
      type uint8 {
        range "0..64";  // range is platform dependent
      }

      description
```


Copyright© 2022 Ciena® Corporation

```
          "Number of ACL entries in use.";
      }

      leaf entries-max {
        type uint8 {
          range "0..64";  // range is platform dependent
        }

        description
          "Maximum number of ACL entries on this platform.";
      }

      leaf rx-frames-no-match {
        type yang:counter64;

        description
          "Rx frames with no match.";
      }

      uses ciena-ip-acl-statistics;
    }

    list ip-acl-entry {
      key "src-ip dest-ip";

      description
        "The list of configured ip-acl entries on the device.";

      leaf src-ip {
        type inet:ip-prefix;

        description
          "Source network.";
      }

      leaf dest-ip {
        type inet:ip-prefix;

        description
          "Destination network.";
      }

      uses ciena-ip-acl-statistics;
    }
  }

  /*
   * RPC
   */

  rpc ciena-ip-acl-clear {
    description
      "Clear the ip-acl statistics.";

    output {
      leaf status {
        type string;
        description
          "Status of the clear statistics operation.";
      }
```


Copyright© 2022 Ciena® Corporation

```
    }
  }

  rpc ciena-ip-acl-import {
    description
      "Import existing interfaces to access control list.";

    output {
      leaf status {
        type string;
        description
          "Status of the import operation.";
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-itu-g8032-deviation.yang

```
    module ciena-itu-g8032-deviation {
    namespace "http://www.ciena.com/ns/yang/ciena-itu-g8032-dev";
    prefix "ciena-itu-g8032-dev";

    import ciena-itu-g8032 {
        prefix "g8032";
    }

    import ciena-mef-fd {
      prefix "mef-fd";
    }

    revision 2017-08-10 {
        description
          "Removed deviation that restricted special characters in g8032:ring-
name and g8032:instance-name";
   }

    revision "2016-06-10" {
        description
            "Initial version";
    }

    typedef mef-fd-name {
        type leafref {
            path "/mef-fd:fds/mef-fd:fd/mef-fd:name";
        }
    }

    deviation /g8032:g8032-rings/g8032:g8032-ring/g8032:ring-description{
        deviate not-supported;
    }

    deviation /g8032:g8032-rings/g8032:notification-enabled{
        deviate not-supported;
    }

    deviation /g8032:g8032-rings/g8032:g8032-ring/g8032:ring-ports/
g8032:ring-port {
        deviate replace {
            min-elements 2;
            max-elements 2;
        }
    }

    deviation /g8032:g8032-rings/g8032:g8032-ring/g8032:erp-instances/
g8032:erp-instance/g8032:data-member-vids {
        deviate replace {
            type mef-fd-name;
      }
    }

    deviation /g8032:g8032-rings/g8032:g8032-ring/g8032:erp-instances/
g8032:erp-instance/g8032:wtr-timer {
        deviate not-supported;
    }
```


Copyright© 2022 Ciena® Corporation

```
    deviation /g8032:g8032-rings/g8032:g8032-ring/g8032:erp-instances/
g8032:erp-instance/g8032:guard-timer {
        deviate not-supported;
    }

    deviation /g8032:g8032-rings/g8032:g8032-ring/g8032:erp-instances/
g8032:erp-instance/g8032:hold-off-time {
        deviate not-supported;
    }

    deviation /g8032:g8032-rings/g8032:g8032-ring/g8032:erp-instances/
g8032:erp-instance/g8032:instance-description {
        deviate not-supported;
    }

    deviation /g8032:g8032-rings/g8032:g8032-ring/g8032:erp-instances/
g8032:erp-instance/g8032:raps-priority {
        deviate not-supported;
    }

    deviation /g8032:g8032-rings/g8032:g8032-ring/g8032:erp-instances/
g8032:erp-instance/g8032:ports/g8032:port {
        deviate replace {
            min-elements 2;
           max-elements 2;
        }
    }

    deviation /g8032:g8032-rings/g8032:g8032-ring/g8032:erp-instances/
g8032:erp-instance/g8032:raps-vid {
        deviate replace {
            mandatory false;
        }
    }

    deviation /g8032:g8032-rings/g8032:g8032-ring/g8032:erp-instances/
g8032:erp-instance/g8032:raps-level {
        deviate replace {
            mandatory false;
        }
    }

    deviation /g8032:g8032-rings/g8032:g8032-ring/g8032:erp-instances/
g8032:erp-instance/g8032:erp-instance-construct {
        deviate replace {
            mandatory false;
        }
    }


    deviation /g8032:g8032-rings/g8032:g8032-ring/g8032:erp-instances/
g8032:erp-instance/g8032:ports {
       deviate delete {
              must "../erp-instance-construct != 'major-ring' or " +
                 "count(port) = 2" {
              error-message "A major ring must have 2 ring ports";
              description "A major ring must have 2 ring ports";
             }
        }
    }
```


Copyright© 2022 Ciena® Corporation

```
    deviation /g8032:g8032-rings/g8032:g8032-ring/g8032:ring-id {
       deviate replace {
                   type uint8 {
                       range "1..255";
             }
         }
    }

}
```


Copyright© 2022 Ciena® Corporation

ciena-itu-g8032.yang

```
module ciena-itu-g8032 {
  namespace "http://www.ciena.com/ns/yang/g8032";
  prefix g8032;

  import ietf-interfaces {
    prefix "if";
  }

  import ieee-dot1q-types  {
    prefix "dot1qtypes";
  }

  import ieee-types {
    prefix "ieee";
  }

  import ietf-yang-types {
    prefix "yang";
  }

  organization
    "Cisco Systems, Inc.,
     Ciena Corporation";

  contact
    "Cisco Systems, Inc.
      - Customer Service
      - Postal: 170 W Tasman Drive, San Jose, CA 95134
      - Tel: +1 1800 553-NETS
      - E-mail: cs-yang@cisco.com

    Ciena Corporation
      - Web URL: http://www.ciena.com/
      - E-mail:  yang@ciena.com
      - Postal:  7035 Ridge Road
                 Hanover, Maryland 21076
                 U.S.A.
      - Phone:   +1 800-921-1144
      - Fax:     +1 410-694-5750
    ";

  description
    "This YANG module describes configuration and operational
    data model for ITU.T G.8032 Ethernet Ring Protocol (ERP).

    G.8032 provides a loop avoidance and protection switching
    mechanism in ethernet ring network.

    An Ethernet ring node (which is assumed to be compliant to an
    IEEE 802.1Q bridge) is connected to adjacent Ethernet ring
    nodes participating in the same Ethernet ring, using two
    independent links. A ring link is bounded by two adjacent
    Ethernet ring nodes and a port for a ring link is called a
    ring port. The minimum number of Ethernet ring nodes in
    an Ethernet ring is two.

    An Automatic Protection Switching(APS) protocol is used to
    coordinate the protection actions over the ring.";
```


Copyright© 2022 Ciena® Corporation

```
  revision 2016-04-06 {
    description
      "This YANG model describes the configuration and
      operational state information in support of G.8032.";
    reference
      "ITU-T G.8032/Y.1344 (08/2015)";
  }

  // Ring types associated with the ERP instance

  identity erp-instance-type {
    description
      "Represents the configured ERP ring type.";
  }

  identity sub-ring-no-virtual-channel {
    base erp-instance-type;
    description
      "Identify for a sub-ring without a virtual channel.";
    reference
      "ITU-T G.8032/Y.1344 (08/2015); Section 9.7.2, 10.1.14";
  }

  identity sub-ring-virtual-channel {
    base erp-instance-type;
    description
      "Identify for a sub-ring with a virtual channel.";
    reference
      "ITU-T G.8032/Y.1344 (08/2015); Section 9.7.1";
  }

  identity major-ring {
    base erp-instance-type;
    description
      "Identify for a major-ring.";
    reference
      "ITU-T G.8032/Y.1344 (08/2015); Section 9.3";
  }

  // G.8032 protocol version types

  identity g8032-version-type {
    description
      "The G.8032 protocol version.";
  }

  identity g8032-version1 {
    base g8032-version-type;
    description
      "ITU-T G.8032 version 1";
    reference
      "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.13";
  }

  identity g8032-version2 {
    base g8032-version-type;
    description
```


Copyright© 2022 Ciena® Corporation

```
      "ITU-T G.8032 version 2";
    reference
      "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.13";
  }

  // Ring state types

  identity ring-state-type {
    description
      "Protocol states for a ring instance.";
  }

  identity initializing {
    base ring-state-type;
    description
      "Ring protocol state machine is initializing and it is not
      yet operational.";
    reference
      "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.2 (Table 10-2)";
  }

  identity idle {
    base ring-state-type;
    description
      "Ring is operational in normal state, RPL link is blocked
      and no failure or administrative condition exists. Owner
      node is transmitting R-APS (NR, RB) PDUs.";
    reference
      "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.2 (Table 10-2)";
  }

  identity pending {
    base ring-state-type;
    description
      "Ring does not have any failure or administrative
      condition however RPL is blocked. In case of revertive
      mode operation wait-to-restore timer is running.";
    reference
      "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.2 (Table 10-2)";
  }

  identity protection {
    base ring-state-type;
    description
      "One or more failure condition exists on the ring. Failed
      nodes are transmitting R-APS (SF) PDUs. Upon reaching to
      protection state ring unblocks RPL link to allow user
      traffic.";
    reference
      "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.2 (Table 10-2)";
  }

  identity manual-switch {
    base ring-state-type;
    description
      "User has manually switched ring. This is usually done to
      perform maintenance operation on the ring node.";
    reference
      "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.2 (Table 10-2)";
  }
```


Copyright© 2022 Ciena® Corporation

```
identity force-switch {
  base ring-state-type;
  description
    "User has force-switched ring. This state overrides
    protection state.";
  reference
    "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.2 (Table 10-2)";
}

// Port state types

identity port-state-type {
  description
    "States associated with a ring port.";
}

identity forwarding {
  base port-state-type;
  description
    "Ring port is in normal state and ready to forward
    traffic ring controlled data traffic.";
}

identity blocking {
  base port-state-type;
  description
    "Ring port is blocked and it is not forwarding any data
    traffic. In normal operations a port is blocked if it is
    a RPL port.";
}

identity failed {
  base port-state-type;
  description
    "Ring port has local failure. Port is blocked and it is
    not forwarding any traffic.";
}

identity local-ms {
  base port-state-type;
  description
    "Ring port has been blocked by administrative manual
    switch action.";
}

identity local-fs {
  base port-state-type;
  description
    "Ring port has been blocked by administrative force switch
    action.";
}

// FOP event types

identity fop-event-type {
  description
    "Represents Failure of Protocol events";
}
```


Copyright© 2022 Ciena® Corporation

```
identity fop-pm {
  base fop-event-type;
  description
    "Failure of Protocol, Provisioning Mismatch (FOP-PM)
    event.

    This event is generated when a node ID mismatch is
    detected by ERP control process by comparing the node ID
    of the RPL owner and the node ID in a received
    R-APS(NR, RB) frame.";
  reference
    "ITU-T G.8032/Y.1344 (08/2015); Section 10.4";
}

identity fop-pm-clear {
  base fop-event-type;
  description
    "This event is generated when Protocol, Provisioning
    Mismatch (FOP-PM) defect is cleared.";
  reference
    "ITU-T G.8032/Y.1344 (08/2015); Section 10.4";
}

identity fop-to {
  base fop-event-type;
  description
    "Failure of Protocol, Timeout (FOP-TO) event.

    This event is generated when ERP control process does not
    receive periodic R-APS frame for a defined timeout
    interval (>= 3.5 times R-APS interval).";
  reference
    "ITU-T G.8032/Y.1344 (08/2015); Section 10.4";
}

identity fop-to-clear {
  base fop-event-type;
  description
    "This event is generated when Protocol, Timeout (FOP-TO)
    defect is cleared";
  reference
    "ITU-T G.8032/Y.1344 (08/2015); Section 10.4";
}

typedef rpl-type {
  type enumeration {
    enum none {
      description
          "No RPL.";
    }
    enum owner {
      description
          "An RPL owner.";
    }
    enum neighbor {
      description
          "An RPL neighbor owner.";
    }
    enum next-neighbor {
      description
```


Copyright© 2022 Ciena® Corporation

```
            "Represents next-neighbour node in the ring.";
      }
    }
    description
      "The disposition of the RPL ownership associated with a ring
      Interface.";
    reference
      "ITU-T G.8032/Y.1344 (08/2015); Section 10.1";
  }

  typedef reversion-mode {
    type enumeration {
      enum revertive {
        description
            "Represents revertive mode operation of a G.8032
            ethernet ring.";
      }
      enum non-revertive {
        description
            "Represents non-revertive mode operation of a
            G.8032 ethernet ring.";
      }
    }
    description
      "This typedef defines G.8032 ring protection operation mode.";
    reference
      "ITU-T G.8032/Y.1344 (08/2015); Section 9.1, 10.2.3.1";
  }

  typedef ring-name-type {
    type string {
      length "1..255";
    }
    description
      "The ring name type identifier.";
  }

  typedef ring-id-type {
    type uint8 {
      range "1..239";
    }
    default 1;
    description
      "G.8032 Ring ID";
    reference
      "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.1";
  }

  typedef ring-port-type {
    type enumeration {
      enum port0 {
        description
          "Port 0 of the ring node.";
      }
      enum port1 {
        description
          "Port 1 of the ring node.";
      }
    }
    description
```


Copyright© 2022 Ciena® Corporation

```
    "Each ring instance has maximum of two ports. This definition
    labels these ports as port0 and port1 for identifications.";
  reference
    "ITU-T G.8032/Y.1344 (08/2015); Section 10.1 (Figure 10-1)";
}

typedef hold-off-timer-type {
  type uint16 {
    range "0..10000";
  }
  units "milliseconds";
  default "0";
  description
    "The hold-off timer is used to coordinate the timing of
    protection switches.

    When a new defect or more severe defect occurs (new SF), this
    event is not to be reported immediately to protection switching
    if the provisioned hold-off timer value is non-zero. Instead,
    the hold-off timer is started. When the hold-off timer expires,
    the trail that started the timer is checked as to whether a
    defect still exists. If one does exist, that defect is
    reported to protection switching.

    Hold off timer values SHOULD be specified in multiple of 100.";
  reference
    "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.8";
}

typedef guard-timer-type {
  type uint16 {
    range "10..2000";
  }
  units "milliseconds";
  default "500";
  description
    "The guard timer is used to prevent Ethernet ring nodes from
    acting upon outdated R-APS messages and prevents the
    possibility of forming a closed loop.

    This timer period SHOULD be greater than the maximum expected
    forwarding delay in which an R-APS message traverses the
    entire ring.

    The period of the guard timer MAY be configured by the operator
    in 10 ms steps between 10 ms and 2 seconds.";
  reference
    "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.5";
}

typedef wtr-timer-type {
  type uint8 {
    range "1..12";
  }
  units "minutes";
  default "5";
  description
    "In revertive mode the 'wait to restore' (WTR) timer
    is used to prevent frequent operation of the protection
    switching due to intermittent signal failure defects.
```


Copyright© 2022 Ciena® Corporation

```
      When recovering from a Signal fail, the delay timer
      must be long enough to allow the recovering network to
      become stable.

      WTR timer is activated on RPL owner node. When WTR
      timer is expired RPL owner node initiates the reversion
      process by transmitting an R-APS (NR, RB) message.";
    reference
      "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.4";
  }

  grouping ring-identifier {
    description
      "This grouping defines ring-name and instance-name
      combination.";
    leaf ring-name {
      type leafref {
        path "/g8032-rings/g8032-ring/ring-name";
      }
      mandatory true;
      description
        "G.8032 ring name.";
    }

    leaf instance-name {
      type leafref {
        path "/g8032-rings/" +
             "g8032-ring[ring-name = current()/../ring-name]/" +
             "erp-instances/erp-instance/instance-name";
      }
      mandatory true;
      description
        "ERP instance name.";
    }
  } // ring-identifier

  grouping aps-statistics {
    description
      "R-APS statistics";
    leaf tx-raps {
      type yang:counter64;
      mandatory true;
      description
        "Number of R-APS messages transmitted";
    }
    leaf rx-raps {
      type yang:counter64;
      mandatory true;
      description
        "Number of R-APS messages received";
    }
    leaf tx-raps-nrrb {
      type yang:counter64;
      description
        "Number of R-APS(NR,RB) messages transmitted";
    }
    leaf rx-raps-nrrb {
      type yang:counter64;
      description
```


Copyright© 2022 Ciena® Corporation

```
        "Number of R-APS(NR,RB) messages received";
    }
    leaf tx-raps-nr {
      type yang:counter64;
      description
        "Number of R-APS(NR) messages transmitted";
    }
    leaf rx-raps-nr {
      type yang:counter64;
      description
        "Number of R-APS(NR) messages received";
    }
    leaf tx-raps-sf {
      type yang:counter64;
      description
        "Number of R-APS(SF) messages transmitted";
    }
    leaf rx-raps-sf {
      type yang:counter64;
      description
        "Number of R-APS(SF) messages received";
    }
    leaf tx-raps-fs {
      type yang:counter64;
      description
        "Number of R-APS(FS) messages transmitted";
    }
    leaf rx-raps-fs {
      type yang:counter64;
      description
        "Number of R-APS(FS) messages received";
    }
    leaf tx-raps-ms {
      type yang:counter64;
      description
        "Number of R-APS(MS) messages transmitted";
    }
    leaf rx-raps-ms {
      type yang:counter64;
      description
        "Number of R-APS(MS) messages received";
    }
  } // aps-statistics

  container g8032-rings {
    description
      "Contains all the G.8032 system specific related
      configuration.";

    leaf ring-system-control {
      type enumeration {
        enum disabled {
          description
            "Disables G.8032 ring protection on a system wide
            basis. There may be occasions during troubleshooting
            that requires the operator to globally disable.";
        }
        enum enabled {
          description
            "Enables G.8032 ring protection on a system wide
```


Copyright© 2022 Ciena® Corporation

```
            basis.";
        }
      }
      default "enabled";
      description
        "Setting this control allows the enablement or disablement
        of the G.8032 protection switching on a system wide
        basis.";
    }

    leaf notification-enabled {
      type boolean;
      default "false";
      description
        "Enable/disable ring status change notification. If true,
        any change in ring state will be notified via
        ring-state-change notification.";
    }

    leaf r-aps-version {
      type identityref {
        base g8032-version-type;
      }
      description
        "The version of the ERP R-APS protocol being used.";
      reference
        "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.13";
    }

    list g8032-ring {
      key "ring-name";
      description
        "Provides configuration nodes in support of ERP instances
        on the device.";

      leaf ring-name {
        type ring-name-type;
        description
          "G.8032 ring name.";
      }

      leaf ring-id {
        type ring-id-type;
        default "1";
        description
          "The Ring ID.";
        reference
          "ITU-T G.8032 (08/2015); Section 10.1.1";
      }

      leaf ring-description {
        type string {
          length "1..255";
        }
        description
          "Textual description of the ring.";
      }

      container ring-ports {
        description
```


Copyright© 2022 Ciena® Corporation

```
      "Ring port configurations.";
    list ring-port {
      key "port-id";
      min-elements 1;
      max-elements 2;
      description
        "List of the ring ports.
         If all of the ERP instances for this ring are sub-rings
         that terminate at this node, this list may have only 1
         entry.";

      leaf port-id {
        type ring-port-type;
        description
          "Port identifier (port0 or port1).";
      }

      leaf interface {
        type if:interface-ref;
        mandatory true;
        description
          "Port interface reference of the Ring.";
      }
    } // ring-port
  } // ring-ports

  container erp-instances {
    description
      "Collection of ring instances.";
    list erp-instance {
      key "instance-name";
      description
        "There can be multiple ERP instances associated with a
        given G.8032 ring. Each ERP instance must have a
        separate set of data member VLAN identifiers.";

      leaf instance-name {
        type ring-name-type;
        description
          "The ERP instance name.";
      }

      leaf instance-description {
        type string {
          length "1..255";
        }
        description
          "Textual description of the instance.";
      }

      leaf raps-vid {
        type ieee:vlanid;
        mandatory true;
        description
          "The VLAN identifier used by the ERP instance for
          ring APS control messages.";
        reference
          "ITU-T G.8032/Y.1344 (08/2015); Section 9.5";
      }
```


Copyright© 2022 Ciena® Corporation

```
leaf raps-level {
  type uint8 {
    range "0..7";
  }
  mandatory true;
  description
    "The Maintenance Entity Level used by the ERP instance
     for ring APS control messages.";
  reference
    "ITU-T G.8032/Y.1344 (08/2015); Section 10.3";
}

leaf raps-priority {
  type dot1qtypes:priority-type;
  description
    "The priority assigned to the R-APS PDUs.";
}

leaf-list data-member-vids {
  type dot1qtypes:vid-range;
 //min-elements 1;
  description
    "The range of data member VLAN identifiers protected
    by the ERP instance.";
}

leaf erp-instance-construct {
  type identityref {
    base erp-instance-type;
  }
  mandatory true;
  description
    "Type of ERP instance.";
  reference
    "ITU-T G.8032/Y.1344 (08/2015); Section 9.3, 9.7";
}

leaf reversion {
  type reversion-mode;
  default "revertive";
  description
    "Configuration node specifying the reversion
    capabilities of this ERP instance.";
  reference
    "ITU-T G.8032/Y.1344 (08/2015); Section 9.1, 10.2.3.1";
}

leaf wtr-timer {
  type wtr-timer-type;
  description
    "The wait-to-restore timer associated with the ERP
    instance.";
  reference
    "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.4";
}

leaf guard-timer {
  type guard-timer-type;
  description
    "The guard timer is used to prevent Ethernet ring nodes
```


Copyright© 2022 Ciena® Corporation

```
          from acting upon outdated R-APS messages and prevents
          the possibility of forming a closed loop.";
        reference
          "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.5";
      }

      leaf hold-off-time {
        type hold-off-timer-type;
        description
          "The hold-off timer is used to coordinate the
          timing of protection switches.";
        reference
          "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.8";
      }

      container ports {
        must "../erp-instance-construct != 'major-ring' or " +
             "count(port) = 2" {
          error-message "A major ring must have 2 ring ports";
          description "A major ring must have 2 ring ports";
            }
        description
          "Port specific configurations for ring instances.";
        list port {
          key "port-id";
          min-elements 1;
          max-elements 2;
          description
            "List of the ring ports.
             If this is a major ring, or this node is in a
             sub-ring but is not the interconnection node where
             the sub-ring terminates, then this list must have 2
             entries.
             If this is a sub-ring (with or without a virtual
             channel), and this is an interconnection node where
             the sub-ring terminates, then this list must have
             only 1 entry.";

          leaf port-id {
            type leafref {
              path "../../../../../ring-ports/ring-port/port-id";
            }
            description
              "Port id of the ring instance. This leaf
              identifies one of the two ring ports.";
          }

          leaf rpl {
            type rpl-type;
            default "none";
            description
              "RPL port role for the ring instance.";
            reference
              "ITU-T G.8032/Y.1344 (08/2015); Section 10.1";
          }

        } // port
      } // ports
    } // erp-instance
  } // erp-instances
```


Copyright© 2022 Ciena® Corporation

```
    } // g8032-ring
  } // g8032-rings

  container g8032-rings-state {
    config false;
    description
      "Contains all the G.8032 system specific related
      operational state nodes.";

    leaf max-instances-supported {
      type uint32;
      mandatory true;
      description
        "Maximum number of G.8032 rings instances supported on the
        device.";
    }

    list g8032-ring {
      key "ring-name";
      description
        "Provides configuration nodes in support of ERP instances
        on the device.";

      leaf ring-name {
        type leafref {
          path "/g8032-rings/g8032-ring/ring-name";
        }
        description
          "G.8032 ring name.";
      }

      container erp-instances {
        description
          "Collection of ring instances.";
        list erp-instance {
          key "instance-name";
          description
            "There can be multiple ERP instances associated with a
            given G.8032 ring. Each ERP instance must have a
            separate set of data member VLAN identifiers.";

          leaf instance-name {
            type leafref {
              path "/g8032-rings/g8032-ring[ring-name = " +
                   "current()/../../../ring-name]/" +
                   "erp-instances/erp-instance/instance-name";
            }
            description
              "The G.8032 ERP instance name.";
          }

          leaf wtb-timer {
            type uint16 {
              range "5010..7000";
            }
            units "milliseconds";
            description
              "The WTB timer is used when clearing Forced switch and
              Manual switch commands. As multiple Forced switch
              commands are allowed to co-exist in an Ethernet ring,
```


Copyright© 2022 Ciena® Corporation

```
    the WTB timer ensures that clearing of a single Forced
    switch command does not trigger the re-blocking of the
    RPL. The timer is defined to be 5 seconds longer than
    the Guard Timer.";
  reference
    "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.4";
}

container ports {
  description
    "Operational state for the ring instance ports.";
  list port {
    key port-id;
    min-elements 1;
    max-elements 2;
    description
      "Collection of ring port states.";

    leaf port-id {
      type leafref {
        path "/g8032-rings/" +
             "g8032-ring[ring-name = " +
                "current()/../../../../../ring-name]/" +
             "erp-instances/" +
             "erp-instance[instance-name = " +
                "current()/../../../instance-name]/" +
             "ports/port/port-id";
      }
      description
        "Port id of the ring instance. This leaf identifies
        one of the two ring ports.";
    }

    container flush-logic-remote {
      description
        "Remote node-id, bpr values, used by flush logic
        of the ERP control process.";
      reference
        "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.10";

      leaf node-id {
        type ieee:mac-address;
        mandatory true;
        description
          "Node ID of the last R-APS message received
          over the ring port.

          During initialization this value is set to
          00:00:00:00:00:00 by ERP control process.";
        reference
          "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.10";
      }

      leaf bpr {
        type uint8 {
          range "0 | 1";
        }
        mandatory true;
        description
          "Blocked port reference (BPR) of the last
```


Copyright© 2022 Ciena® Corporation

```
                    R-APS message received over the ring port.

                    During initialization this value is set to 0
                    by ERP control process.";
                  reference
                    "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.10";
                }
              } // flush-logic-remote

              leaf port-state {
                type identityref {
                  base port-state-type;
                }
                mandatory true;
                description
                  "Ring port state";
              }

              container r-aps-statistics {
                description
                  "R-APS packet statistics";
                uses aps-statistics;
              }
            } // port
          } // ports

          leaf ring-instance-state {
            type identityref {
              base ring-state-type;
            }
            mandatory true;
            description
              "Ring instance state. This represent protocol state
              machine state.";
            reference
              "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.2
              (Table 10-2)";
          }
        } // erp-instance
      } // erp-instances
    } // g8032-ring
  } // g8032-rings-state

  /*
   * Notifications
   */

  notification ring-state-change-event {
    description
      "Notify ring state change events.";
    uses ring-identifier {
      description
        "Uniquely identifies a ring instance.";
    }

    leaf current-state {
      type identityref {
        base ring-state-type;
      }
      mandatory true;
```


Copyright© 2022 Ciena® Corporation

```
      description
        "New state of the ring instance.";
      reference
        "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.2 (Table 10-2)";
    }

    leaf previous-state {
      type identityref {
        base ring-state-type;
      }
      mandatory true;
      description
        "Previous state of the ring instance.";
      reference
        "ITU-T G.8032/Y.1344 (08/2015); Section 10.1.2 (Table 10-2)";
    }
  } // ring-state-change-event

  notification fop-event {
    description
      "Failure of Protocol (FOP) events.";
    reference
      "ITU-T G.8032/Y.1344 (08/2015); Section 10.4, and " +
      "ITU-T G.8021/Y.1341 (04/2015); Section 6.1.4.3";
    uses ring-identifier {
      description
        "Uniquely identifies a ring instance.";
    }

    leaf event-type {
      type identityref {
        base fop-event-type;
      }
      description
        "G.8032 Failure of Protocol event types.";
    }
  } // fop-event

  /*
   * Remote Procedure Calls
   */

  rpc clear-statistics {
    description
      "Clear ring statistics.";

    input {
      uses ring-identifier {
        description
          "Uniquely identifies ring instance.";
      }
    }

    output {
      leaf errstr {
        type string;
        description
          "Error message from the device if RPC failed.";
      }
    }
```


Copyright© 2022 Ciena® Corporation

```
  } // clear-statistics

  rpc clear-ring {
    description
      "Clear an existing Forced-Switch (FS) or Manual-Switch (MS)
      condition on the ring port.";
    reference
      "ITU-T G.8032/Y.1344 (08/2015); Sections 8, 10.1.9,
      10.2.4.1, 10.2.5.1";

    input {
      uses ring-identifier {
        description
          "Uniquely identifies ring instance.";
      }
    }

    output {
      leaf errstr {
        type string;
        description
          "Error message from the device if RPC failed.";
      }
    }
  } // clear-ring

  rpc switch-ring {
    description
      "Perform administrative action on the ring (instance).";

    input {
      uses ring-identifier {
        description
          "Uniquely identifies ring instance.";
      }

      leaf port-id {
        type leafref {
          path "/g8032-rings/" +
               "g8032-ring[ring-name = current()/../ring-name]/" +
               "erp-instances/erp-instance[instance-name = " +
               "current()/../instance-name]/ports/port/port-id";
        }
        mandatory true;
        description
          "Ring port to perform switch operation.";
      }

      leaf operation {
        type enumeration {
          enum manual {
            description
              "Trigger manual switch. This operation is allowed
              only when a ring instance is in idle or pending state.

              This action allows administrator to temporarily block
              another ring port and unblock RPL link to perform
              maintenance operation in the ring.";
            reference
              "ITU-T G.8032/Y.1344 (08/2015); Section 10.2.4";
```


Copyright© 2022 Ciena® Corporation

```
          }
          enum forced {
            description
              "Trigger force switch. This operation overrides all
              other state of the ring.";
            reference
              "ITU-T G.8032/Y.1344 (08/2015); Section 10.2.5";
          }
        }
        mandatory true;
        description
          "G.8032 administrative operation types.";
        reference
          "ITU-T G.8032/Y.1344 (08/2015); Section 8, 10.1.9";
      }
    } // input

    output {
      leaf errstr {
        type string;
        description
          "Error message from the device if RPC failed, including
          failures where switch operation is not permitted on the
          ring instance under protocol restrictions.";
      }
    } // output
  } // switch-ring

} // ietf-g8032
```


Copyright© 2022 Ciena® Corporation

ciena-l2cp-profile.yang

```
module ciena-l2cp-profile {
   namespace "http://www.metroethernetforum.org/ns/yang/ciena-l2cp-profile";
    prefix "ciena-l2cp";

    import ietf-yang-types {
        prefix yang;
        revision-date 2010-09-24;
    }

    import ciena-mef-l2cp-profile {
        prefix "mef-l2cp";
    }

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
                 Hanover, Maryland 21076
                 U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
        "This YANG module defines Ciena's L2 Control Protocol
         configuration additions.";

    revision "2017-01-04" {
        description
             "Updated description of mef-ce2 global-l2cp-mode that
              IS-IS MACs block will not inherit disposition of all-bridges-
block";
    }

    revision "2015-05-13" {
        description
            "Initial version";
    }


    typedef l2pt-mac-ref {
        type leafref {
           path "/ciena-l2cp:l2pt-macs/ciena-l2cp:l2pt-mac/ciena-l2cp:name";
        }
        description
            "This type is used by the data models that need to reference
             configured L2PT MAC address instances.";
    }

    typedef fixed-cos-map-ref {
        type leafref {
            path "/ciena-l2cp:fixed-cos-maps/ciena-l2cp:fixed-cos-map/ciena-
l2cp:priority";
        }
        description
            "This type is used by the data models that need to reference
```


Copyright© 2022 Ciena® Corporation

```
            fixed priority to CoS mapping.";
    }

    container l2pt-macs {

        description "Contains table of L2PT Translation MAC addresses.";

        list l2pt-mac {
            key "name";

            leaf name {
                type string;

                description
                    "User assigned name of L2PT Translation MAC address to
                     be stamped over L2 control protocol's standard defined
                     Destination MAC at egress.";
            }

            leaf l2pt-mac-address {
                type yang:mac-address;

                description
                    "This stores the L2PT Translation MAC address.";
            }
        }

    }

    augment "/mef-l2cp:l2cp-profiles/mef-l2cp:l2cp-profile" {

        leaf l2pt-mac-name {
            type ciena-l2cp:l2pt-mac-ref;

            description
                "Reference to a L2PT translation MAC instance.";
        }

    }

    container fixed-cos-maps {

        description "Contains table of fixed priority to CoS mappings.";

        list fixed-cos-map {
            key "priority";

            leaf priority {
                type enumeration {
                    enum high {
                       description
                       "High fixed priority that can be assigned to forwarded
                            untagged L2 control frames.";
                    }
                    enum low {
                        description
                        "Low fixed priority that can be assigned to forwarded
                             untagged L2 control frames.";
                    }
                    enum medium {
```


Copyright© 2022 Ciena® Corporation

```
                        description
                      "Medium fixed priority that can be assigned to forwarded
                            untagged L2 control frames.";
                    }
                }

                description
                    "System chosen name of fixed priority.";
            }

            leaf fixed-cos {
                type uint8 {
                    range "0..7";
                }
                description
                   "This denotes CoS value assigned to the fixed priority.";
            }
        }
    }

    augment "/mef-l2cp:l2cp-profiles/mef-l2cp:l2cp-profile" {

        leaf fixed-cos-priority {
            type ciena-l2cp:fixed-cos-map-ref;

            description
                "Reference to a fixed priority to CoS mapping instance.";
        }

    }

    augment "/mef-l2cp:l2cp-profiles/mef-l2cp:l2cp-profile" {

        list ciena-protocol-disposition {
            key "protocol";

            leaf protocol {
                type enumeration {
                    enum cisco-cdp {
                        description "Cisco Discovery Protocol (CDP). MAC DA
01-00-0C-CC-CC-CC.
                                     LLC/SNAP AA-AA-03-00-00-0C-20-00.";
                    }
                    enum cisco-dtp {
                        description "Cisco Dynamic Trunking Protocol (DTP).
MAC DA 01-00-0C-CC-CC-CC.
                                     LLC/SNAP AA-AA-03-00-00-0C-20-04.";
                    }
                    enum cisco-pagp {
                       description "Cisco Port Aggregation Protocol (PAGP).
MAC DA 01-00-0C-CC-CC-CC.
                                     LLC/SNAP AA-AA-03-00-00-0C-01-04.";
                    }
                    enum cisco-pvst {
                      description "Cisco Per VLAN Spanning Tree (PVST). MAC
DA 01-00-0C-CC-CC-CD.
                                     LLC/SNAP AA-AA-03-00-00-0C-01-0B.";
                    }
                    enum cisco-stp-uplink-fast {
```


Copyright© 2022 Ciena® Corporation

```
                        description "Cisco STP UplinkFast. MAC DA 01-00-0C-
CD-CD-CD.
                                     LLC/SNAP AA-AA-03-00-00-0C-20-0A.";
                    }
                    enum cisco-udld {
                    description "Cisco Unidirectional Link Detection (UDLD).
MAC DA 01-00-0C-CC-CC-CC.
                                     LLC/SNAP AA-AA-03-00-00-0C-01-11.";
                    }
                    enum cisco-vtp {
                       description "Cisco VLAN Trunk Protocol (VTP). MAC DA
01-00-0C-CC-CC-CC.
                                     LLC/SNAP AA-AA-03-00-00-0C-20-03.";
                    }
                    enum gmrp {
                      description "GARP Multicast Registration Protocol. MAC
DA 01-80-C2-00-00-20.";
                    }
                    enum gvrp {
                       description "GARP VLAN Registration Protocol. MAC DA
01-80-C2-00-00-21.";
                    }
                    enum vlan-bridge {
                    description "Cisco VLAN Bridge. MAC DA 01-00-0C-CD-CD-CE.
                                     LLC/SNAP AA-AA-03-00-00-0C-01-0C.";
                    }
                    enum all-bridges-block {
                        description "All Bridges Block. MAC DA 01-80-C2-00-
00-10 to 01-80-C2-00-00-1F.";
                    }
                }
            }

            leaf untagged-disposition {
                type mef-l2cp:DispositionType;

                description
                    "Disposition assigned to untagged form of L2 control
protocol PDUs.";
            }

            leaf tagged-disposition {
                type mef-l2cp:DispositionType;

                description
                  "Disposition assigned to tagged form of L2 control protocol
PDUs.";
            }

            leaf l2pt-mac-translation {
                type boolean;
                default "false";

                description
                   "This tells the tunneling method to be used for forwarded
untagged L2 control
                    frames. This is applicable if L2 control protocol is
assigned forward disposition
                    for untagged frames. True value indicates L2PT tunnel
method while False value
```


Copyright© 2022 Ciena® Corporation

```
                        indicates transparent tunnel method.";
            }
        }

    }

   augment "/mef-l2cp:l2cp-profiles/mef-l2cp:l2cp-profile/mef-l2cp:protocol-
disposition" {
        leaf l2pt-mac-translation {
            type boolean;
            default "false";

            description
                "This tells the tunneling method to be used for forwarded
untagged L2 control
                frames. This is applicable if L2 control protocol is assigned
forward disposition
                for untagged frames. True value indicates L2PT tunnel method
while False value
                indicates transparent tunnel method.";
        }

    }

    augment "/mef-l2cp:l2cp-profiles" {
        leaf global-l2cp-mode {
            type enumeration {
                enum mef-ce1 {
                    description
                        "MEF-CE1 Mode. Under this mode, user is not able to
configure dispositions
                         of bridge-rsvd-0b0f and bridge-rsvd-0c0d protocols
independently and
                       these protocols always inherit dispositions of bridge-
block.";
                }
                enum mef-ce2 {
                    description
                        "MEF-CE2 Mode. Under this mode, user is allowed to
configure dispositions
                         of bridge-rsvd-0b0f and bridge-rsvd-0c0d protocols
independent of bridge-block.
                         Additionally IS-IS MAC block(01:80:C2:00:00:14/
01:80:C2:00:00:15) will not inherit
                         disposition of all-bridges-block. IS-IS MACs shall
be treated as data frames.";
                }
            }
            default "mef-ce1";

            description
                "Global L2CP mode. It determines the behavior of bridge-rsvd-
0b0f and bridge-rsvd-0c0d protocols.";
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-log-sds.yang

```
module ciena-log-sds {
  namespace "http://www.ciena.com/ns/yang/ciena-log-sds";
  prefix "ciena-log-sds";

  import ciena-types {
    prefix ciena;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module defines the nodes for
     System Logging for the NETCONF protocol for 6x.

     Single Blade.

     Copyright (c) 2016 Ciena Corporation.  All rights reserved.

     All information contained herein is, and remains
     the property of Ciena Corporation. Dissemination of this
     information or reproduction of this material is strictly
     forbidden unless prior written permission is obtained from
     Ciena Corporation.";
   revision "2017-08-29" {
    description
      "Added in a new manager";
       reference
       "";
      }
  revision "2016-09-29" {
    description
      "Initial version";
    reference
      "RFC 6020: YANG - A Data Modeling Language for
       the Network Configuration Protocol (NETCONF).
       No specific reference; standard not available.";
  }

  /*
   * NOTES:
   * 1.
   *
   *
   */

  /*
   * Typedefs
   */
```

```
typedef manager-name {
  type enumeration {
    enum acl-mgr {
      value 1;
      description
        "The   manager.";
    }
    enum ais-mgr {
      value 2;
      description
        "The   manager.";
    }
    enum alarmconfig-mgr {
      value 3;
      description
        "The   manager.";
    }
    enum alarm-mgr {
      value 4;
      description
        "The   manager.";
    }
    enum arp-mgr {
      value 5;
      description
        "The   manager.";
    }
    enum bcast-containment-mgr {
      value 6;
      description
        "The   manager.";
    }
    enum benchmark-mgr {
      value 7;
      description
        "The   manager.";
    }
    enum bfd-mgr {
      value 8;
      description
        "The   manager.";
    }
    enum blade-mgr {
      value 9;
      description
        "The   manager.";
    }
    enum cfm-mgr {
      value 10;
      description
        "The   manager.";
    }
    enum chassis-mgr {
      value 11;
      description
        "The   manager.";
    }
    enum config-mgr {
      value 12;
```


Copyright© 2022 Ciena® Corporation

```
  description
    "The   manager.";
}
enum dc-mgr {
  value 13;
  description
    "The   manager.";
}
enum dhcp-mgr {
  value 14;
  description
    "The   manager.";
}
enum dhcpv6-mgr {
  value 15;
  description
    "The   manager.";
}
enum dnld-mgr {
  value 16;
  description
    "The   manager.";
}
enum dnsc-mgr {
  value 17;
  description
    "The   manager.";
}
enum dot1x-mgr {
  value 18;
  description
    "The   manager.";
}
enum event-mgr {
  value 19;
  description
    "The   manager.";
}
enum flow-mgr {
  value 20;
  description
    "The   manager.";
}
enum igmp-mgr {
  value 21;
  description
    "The   manager.";
}
enum interface-mgr {
  value 22;
  description
    "The   manager.";
}
enum isis-mgr {
  value 23;
  description
    "The   manager.";
}
enum l2-protocol-mgr {
  value 24;
```


Copyright© 2022 Ciena® Corporation

```
      description
        "The   manager.";
    }
    enum lacp-mgr {
      value 25;
      description
        "The   manager.";
    }
    enum ldp-mgr {
      value 26;
      description
        "The   manager.";
    }
    enum lldp-mgr {
      value 27;
      description
        "The   manager.";
    }
    enum logging-mgr {
      value 28;
      description
        "The   manager.";
    }
    enum mpls-ldp-mgr {
      value 29;
      description
        "The   manager.";
    }
    enum mpls-mgr {
      value 30;
      description
        "The   manager.";
    }
    enum mstp-mgr {
      value 31;
      description
        "The   manager.";
    }
    enum ndp-mgr {
      value 32;
      description
        "The   manager.";
    }
    enum netconf-helper-mgr {
      value 33;
      description
        "The   manager.";
    }
    enum ntpc-mgr {
      value 34;
      description
        "The   manager.";
    }
    enum ntwrk-mgr {
      value 35;
      description
        "The   manager.";
    }
    enum oam-mgr {
      value 36;
```


Copyright© 2022 Ciena® Corporation

```
    description
      "The   manager.";
  }
  enum ospf-mgr {
    value 37;
    description
      "The   manager.";
  }
  enum perfmon-mgr {
    value 38;
    description
      "The   manager.";
  }
  enum pfg-mgr {
    value 39;
    description
      "The   manager.";
  }
  enum port-mgr {
    value 40;
    description
      "The   manager.";
  }
  enum prtc-mgr {
    value 41;
    description
      "The   manager.";
  }
  enum radius-mgr {
    value 42;
    description
      "The   manager.";
  }
  enum raps-mgr {
    value 43;
    description
      "The   manager.";
  }
  enum rmon-mgr {
    value 44;
    description
      "The   manager.";
  }
  enum routing-domain-mgr {
    value 45;
    description
      "The   manager.";
  }
  enum rstp-mgr {
    value 46;
    description
      "The   manager.";
  }
  enum rsvp-mgr {
    value 47;
    description
      "The   manager.";
  }
  enum rtm-mgr {
    value 48;
```


Copyright© 2022 Ciena® Corporation

```
      description
        "The   manager.";
    }
    enum security-mgr {
      value 49;
      description
        "The   manager.";
    }
    enum snmp-mgr {
      value 50;
      description
        "The   manager.";
    }
    enum software-mgr {
      value 51;
      description
        "The   manager.";
    }
    enum spidd-mgr {
      value 52;
      description
        "The   manager.";
    }
    enum ssh-mgr {
      value 53;
      description
        "The   manager.";
    }
    enum sub-port-mgr {
      value 54;
      description
        "The   manager.";
    }
    enum sync-mgr {
      value 55;
      description
        "The   manager.";
    }
    enum tacacs-mgr {
      value 56;
      description
        "The   manager.";
    }
    enum tdm-mgr {
      value 57;
      description
        "The   manager.";
    }
    enum traffic-profiling-mgr {
      value 58;
      description
        "The   manager.";
    }
    enum twamp-mgr {
      value 59;
      description
        "The   manager.";
    }
    enum unknown0 {
      value 60;
```


Copyright© 2022 Ciena® Corporation

```
        description
          "The   manager.";
      }
      enum unknown1 {
        value 61;
        description
          "The   manager.";
      }
      enum unknown2 {
        value 62;
        description
          "The   manager.";
      }
      enum unknown3 {
        value 63;
        description
          "The   manager.";
      }
      enum unknown4 {
        value 64;
        description
          "The   manager.";
      }
      enum unknown5 {
        value 65;
        description
          "The   manager.";
      }
      enum virtual-circuit-mgr {
        value 66;
        description
          "The   manager.";
      }
      enum vlli-mgr {
        value 67;
        description
          "The   manager.";
      }
      enum xcvr-mgr {
        value 68;
        description
          "The   manager.";
      }
      enum xftp-mgr {
        value 69;
        description
          "The   manager.";
      }
      enum all-mgrs {
        description
          "All managers.";
      }
    }
    description
      "Manager names
       NOTE: This list is based on the sorted order of managers, not
       their actual manager ID's";
  }

  /*
```


Copyright© 2022 Ciena® Corporation

```
   * Configuration data nodes
   */
  container log {
    description
      "The log configuration.";

    leaf flash-admin-state {
      type ciena:admin-state;
      description
        "The flash admin state of the log.";
    }

    leaf ram-admin-state {
      type ciena:admin-state;
        description
          "The RAM admin state of the log.";
    }

    list log-filter {
      key "destination filter-name";
      description
        "The list of log filters.";

      leaf destination {
        type enumeration {
          enum flash {
            description
              "The flash destination.";
          }
          enum ram {
            description
              "The RAM destination.";
          }
        }
        description
          "The destination for the log.";
      }

      leaf filter-name {
        type string {
          length "1..16";
        }
        description
          "Log filter name.";
      }

      leaf admin-state {
        type ciena:admin-state;
        description
          "Log filter admin state.";
      }

      leaf-list severity {
        type enumeration {
          enum cleared {
            description "";
          }
          enum indeterminate {
            description "";
          }
```


Copyright© 2022 Ciena® Corporation

```
      enum critical {
        description "";
      }
      enum major {
        description "";
      }
      enum minor {
        description "";
      }
      enum warning {
        description "";
      }
      enum config {
        description "";
      }
      enum info {
        description "";
      }
      enum debug {
        description "";
      }
      enum local {
        description "";
      }
      enum all {
        description
          "All supported severities.";
      }
    }
    min-elements 1;
    description
      "An event severity.";
    reference
      "ITU X.733";
  }

  container port-list {
    leaf-list port-number {
      type string {
        pattern
          '[i]?[0-9]{1,4}|[0-9]{1,4}[./][0-9]{1,4}';
      }
      description
        "A port number.  The number may be of the form iN, N, N.N, or
         N/N, where N is a sequence of one to four decimal digits,
         so that the range of N is 0-9999.";
    }
    description
      "The list of ports.";
  }

  list manager-categories {
    key "manager-name";
    description
      "Category list by manager.";

    leaf manager-name {
      type manager-name;
      description
        "The manager name.";
```


Copyright© 2022 Ciena® Corporation

```
}

leaf-list category {
  type enumeration {
    enum critical {
      description
        "The critical category.";
    }
    enum major {
      description
        "The major category.";
    }
    enum minor {
      description
        "The minor category.";
    }
    enum warning {
      description
        "The warning category.";
    }
    enum config {
      description
        "The config category.";
    }
    enum info {
      description
        "The info category.";
    }
    enum debug {
      description
        "The debug category.";
    }
    enum error {
      description
        "The error category.";
    }
    enum options {
      description
        "The options category.";
    }
    enum pkt {
      description
        "The pkt category.";
    }
    enum sm {
      description
        "The sm category.";
    }
    enum timers {
      description
        "The timers category.";
    }
    enum resolve {
      description
        "The resolve category.";
    }
    enum monitor {
      description
        "The monitor category.";
    }
```


Copyright© 2022 Ciena® Corporation

```
        enum notif {
          description
            "The notif category.";
        }
        enum muxsm {
          description
            "The muxsm category.";
        }
        enum ptxsm {
          description
            "The ptxsm category.";
        }
        enum rxsm {
          description
            "The rxsm category.";
        }
        enum slsm {
          description
            "The slsm category.";
        }
        enum txsm {
          description
            "The txsm category.";
        }
        enum other {
          description
            "The other category.";
        }
        enum pdu {
          description
            "The txsm category.";
        }
        enum statechange {
          description
            "The statechange category.";
        }
        enum events {
          description
            "The events category.";
        }
        enum egresspkt {
          description
            "The egresspkt category.";
        }
        enum ingresspkt {
          description
            "The ingresspkt category.";
        }
        enum accounting {
          description
            "The accounting category.";
        }
        enum all {
          description
            "All categories.";
        }
      }
      description
        "The list of manager event categories for the log filter.";
    }
```


Copyright© 2022 Ciena® Corporation

```
      }

    } // end list log filter

  } // end log container

  /*
   * Operational state data nodes
   */
  container log-state {
    description
      "The log state.";

    list log-filter {
      key "filter-name";
      description
        "A list of log filters.";

      leaf filter-name {
        type string;
        description
          "Log filter name.";
      }

      leaf index {
        type int32;
        description
          "Log filter index.";
      }

      leaf oper-state {
        type int32; //ciena:oper-state;
        description
          "Log filter operational state.";
      }

      leaf no-ports {
        type int32;
        description
          "The number of ports included in the filter.";
      }

    }  // end list

    // Need to review how transfer of files is done in netconf.
    // Should the file transfer logic be in ciena-types as a standard way to
transfer files?
   // The Binary type is not supported in NETCONF development scripts... right
now.
    container log-file {
      description
        "The log file.";
      // log flash view / log ram view
      leaf log-contents { // or log-file?
        type int32; // type binary. changed from binary to see if causing a
build issue?
        description
          "The flash/ram log file. Not tested.
```


Copyright© 2022 Ciena® Corporation

```
          The assumption is that his leaf will be able to send an entire file
to the user.
           This leaf is attempting to send the entire log output from 'log
flash view'. OR
          -- what files does the 'log flash view' command read? when displayed
on the CLI?

           The log files can be extremely long.
          Are there concerns about netconf server excessive load / possible
processing delay.
           -- can one request block other netconf server requests?

         The binary built-in type represents any binary data, i.e., a sequence
of octets.
          Binary values are encoded with the base64 encoding scheme. Base64
binary-to-text encoding schemes.
           Base 64 Encoding https://tools.ietf.org/html/rfc4648#section-4.";
      } // end security-log-... leaf

      // logging view destination <TAB>
      // flash      ram
      leaf destination {
        type enumeration {
          enum flash {
            description
              "The flash log.";
          }
          enum ram {
            description
              "The RAM log.";
          }
        }
        description
          "The destination log.";
      }

      leaf argument-type {
        type enumeration {
          enum tail { //  tail <NUMBER: 1..100>
            description
              "The number of lines shown at end of file.";
          }
          enum keyword {  // keyword <String[1..16]>
            description
              "The keyword string.";
          }
        }
        description
          "The argument types.";
      } // end log-argument-type leaf

      leaf argument-value {
        type string;
        description
          "The argument value.
         This value will be taken as a string, and converted to the type value.
           The value will be checked for the format expected by the argument
type.
          This expects that each argument type only has one argument value.";
      }
```


Copyright© 2022 Ciena® Corporation

```
    } // end log-file container

  } // end log-state container

  /*
   * RPC
   */

  rpc log-send {
    description
      "Send a debug log message.";

    input {
      leaf msg {
        type string {
          length "1..127";
        }
        description
          "The message string.";
      }
      leaf severity {
        type int32; // ciena:event-severity;
        description
          "The severity of the debug message.";
      }
    }
    output {
      leaf status {
        type string;
        description
          "Result of operation.";
      }
    }
  } // end rpc log-send

  rpc log-reset-to-factory-defaults {
    description
      "Reset the log to factory defaults.";

    output {
      leaf status {
        type string;
        description
          "Results of operation.";
      }
    }
  }

  rpc log-clear {
    description
      "Clear the log.";

    input {
      leaf destination {
        type string; // should be enum.
        description
          "The log destination.";
      }
    }
```


Copyright© 2022 Ciena® Corporation

```
    output {
      leaf status {
        type string;
        description
          "Results of operation.";
      }
    }
  }

} // end ciena log module
```


Copyright© 2022 Ciena® Corporation

ciena-logical-port.yang

```
module ciena-logical-port {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-logical-port";
    prefix "ciena-logical-port";

    import ciena-mef-logical-port {
        prefix "mef-logical-port";
    }

    import ciena-mef-l2cp-profile {
        prefix "mef-l2cp";
    }

    import ciena-mef-egress-qos {
        prefix "mef-egress-qos";
    }

    import ciena-types {
        prefix "ciena";
    }

    import ciena-pfg-profile-sds {
        prefix "ciena-pfg";
    }

    import ciena-mef-pfg-profile {
        prefix "mef-pfg";
    }

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
                 Hanover, Maryland 21076
                 U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
        "This YANG module defines Ciena's management data definition for the
         management of a Logical Port.";

    revision "2017-12-05" {
        description
           "Added leaf vlan-ether-type-policy to allow VLAN ethertype policy
            configuration on a logical port";
        reference
            "RFC 6020: YANG - A Data Modeling Language for
             the Network Configuration Protocol (NETCONF).
             No specific reference; standard not available.";
    }

    revision "2017-11-15" {
        description
            "Added enum value ignore to egress-frame-cos-policy leaf.
             Removed mac-learning and mac-learn-limit.
```


Copyright© 2022 Ciena® Corporation

```
             Added resolved-cos-remark.
             Added egress-filter, egress-tpid and l2-transform leaves.
             Added new enum dot1d-tag2-cos under cos-policy leaf.
            Reference to PFG profiles and forwarding groups have been added.
            Ports are assigned forwarding groups for each PFG profile. While
             PFG profiles (i.e.- 'univ' and 'mgmt') will be used to specify
           which forwarding groups traffic of logical ports will be forwarded
to.";
        reference
            "RFC 6020: YANG - A Data Modeling Language for
             the Network Configuration Protocol (NETCONF).
             No specific reference; standard not available.";
    }

    revision "2016-05-06" {
        description
           "Reference to L2 control frame tunneling profile has been added.
            Additionally this adds the fixed color leaf.";
        reference
            "RFC 6020: YANG - A Data Modeling Language for
             the Network Configuration Protocol (NETCONF).
             No specific reference; standard not available.";
    }

    revision "2015-07-20" {
        description
            "Initial version";
        reference
            "RFC 6020: YANG - A Data Modeling Language for
             the Network Configuration Protocol (NETCONF).
             No specific reference; standard not available.";
    }

    /*
     * typedefs
     */

    typedef transform-mode {
        type enumeration {
          enum ipush-epop {
            description
              "i-push,e-pop";
          }
          enum ipush-epop-stamp {
            description
              "i-push,e-pop:stamp";
          }
          enum istamp-push-ematch-pop-stamp {
            description
              "i-stamp:push,e-match-pop:stamp";
          }
        }
        description
          "transform actions";
    }

    /*
     * Configuration data nodes
     */
   augment "/mef-logical-port:logical-ports/mef-logical-port:logical-port" {
```


Copyright© 2022 Ciena® Corporation

```
        container ciena-logical-port {
            leaf acceptable-frame-type {
                type enumeration {
                    enum tagged-only;
                    enum untagged-only;
                    enum all;
                }
            }

            leaf fixed-cos {
                type uint32;

                description
                    "Fixed cos value assigned when the cos policy
                     is set to fixed-cos.";
            }

            leaf fixed-color {
                type enumeration {
                    enum green;
                    enum yellow;
                }

                description
                    "Fixed color value assigned when the cos policy
                     is set to fixed-cos.";
            }

            leaf cos-policy {
                type enumeration {
                    enum dot1d-tag1-cos {
                        description
                            "If the port resolved CoS policy is set to dot1d-
tag1-cos then outer VLAN tag PCP/CFI value
                             is mapped to the port resolved CoS mapping table
to derive the R-CoS and R-Color.";
                    }
                    enum dot1d-tag2-cos {
                        description
                            "If the port resolved CoS policy is set to dot1d-
tag2-cos then inner VLAN tag PCP/CFI value
                              is mapped to the resolved CoS mapping table to
derive the R-CoS and R-Color.";
                    }
                    enum fixed-cos {
                        description
                            "If the port resolved CoS policy is set to fixed-
cos then the fixed R-CoS/R-Color is applied
                           to the frame from the fixed resolved CoS and fixed
resolved Color value for the port.";
                    }
                    enum l3-dscp-cos {
                        description
                            "If the port resolved CoS policy is set to l3-
dscp-cos then the layer 3 type of service(TOS)
                             byte is mapped to the port resolved CoS mapping
table to derive the R-CoS and R-Color.";
                    }
                }
```


Copyright© 2022 Ciena® Corporation

```
    description
        "Sets the method used to assign cos values to
         incomming packets.";
}

leaf egress-frame-cos-policy {
    type enumeration {
        enum ignore;
        enum rcos-to-l2-outer-pcp-map;
    }
    default ignore;
}

leaf l2cp-profile {
   type mef-l2cp:l2cp-profile-ref;
   description
    "Reference to a Layer 2 Control Protocol Tunneling Profile.";
}

leaf cos-queue-map {
    type mef-egress-qos:cos-queue-map-ref;
    description
       "Reference to queue-map.";
}

leaf egress-filter {
    type ciena:on-off-state;
    default off;
    description
      "Limit flooding from NNI to UNI";
}

leaf l2-transform {
    type transform-mode;
    default ipush-epop;
    description
      "L2 transform actions available on the port";
}

leaf egress-tpid {
    type enumeration {
      enum 0x8100 {
        description
          "VLAN tagged.";
      }
      enum 0x88a8 {
        description
          "Provider bridging.";
      }
      enum 0x9100 {
        description
          "Proprietary.";
      }
    }
    default 0x8100;
    description
      "TPID to use for VLAN based Egress TPID stamping.";

}
```


Copyright© 2022 Ciena® Corporation

```
            leaf resolved-cos-remark-l2 {
                type boolean;
                default false;
                description
                   "resolved COS remark";
            }

            container pfg-params {
                list params {
                    key profile-name;
                    leaf profile-name {
                        type mef-pfg:pfg-profile-ref;
                        description
                            "This represents the name of l2 based PFG profile
                              (univ or mgmt) for this port.";
                    }

                    leaf forwarding-group {
                        type identityref {
                            base ciena-pfg:frwd-grps;
                        }
                        description
                            "This represents the forwarding group (A or B)
                             traffic of this port will be forwarded to .";
                    }
                }
            }

            leaf egress-tpid-policy {
                type enumeration {
                    enum all {
                        description
                         "TPID used for frames egressing through the logical-port
                           will be using the value set in the egress-tpid leaf.";
                    }
                    enum vlan-tpid {
                        description
                         "TPID used for frames egressing through the logical-
port will be
                              using the value set in leaf vlan-tpid of ciena-
fd yang model.";
                    }
                }
                default all;
                description
                    "Egress frames TPID policy.";
            }
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-mac-management-deviation-sds.yang

```
module ciena-mac-management-deviation-sds {
    namespace "http://www.ciena.com/ns/yang/ciena-mac-management-deviation-
sds";
    prefix "mac-mgmt-dev";

    import ciena-mef-mac-management {
        prefix "mef-mac-management";
    }

    revision 2017-08-25 {
        description "Initial revision";
    }

    deviation /mef-mac-management:mac-management/mef-mac-management:mac-
maximum-table-size {
        deviate not-supported;
    }
    deviation /mef-mac-management:fdbs/mef-mac-management:fdb/mef-mac-
management:description {
        deviate not-supported;
    }
    deviation /mef-mac-management:fdbs/mef-mac-management:fdb/mef-mac-
management:mac-learning {
        deviate not-supported;
    }
    deviation /mef-mac-management:static-macs/mef-mac-management:mac-entry/
mef-mac-management:mac-address-mask {
        deviate not-supported;
    }
    deviation /mef-mac-management:static-macs/mef-mac-management:mac-entry/
mef-mac-management:destination-interface/mef-mac-management:logical-port {
        deviate not-supported;
    }
    deviation /mef-mac-management:static-macs/mef-mac-management:mac-entry/
mef-mac-management:filter-action {
        deviate not-supported;
    }
   deviation /mef-mac-management:dynamic-mac-flush/mef-mac-management:input/
mef-mac-management:flush-scope/mef-mac-management:flow-point {
        deviate not-supported;
    }
   deviation /mef-mac-management:dynamic-mac-flush/mef-mac-management:input/
mef-mac-management:flush-scope/mef-mac-management:all {
        deviate not-supported;
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-management-interfaces-sds.yang

```
module ciena-management-interfaces-sds {
  namespace "http://www.ciena.com/ns/yang/ciena-management-interfaces-sds";
  prefix "ciena-mgmt-interfaces";

  import ietf-inet-types {
    prefix "inet";
  }

  import ietf-yang-types {
    prefix "yang";
  }

  import ciena-types {
    prefix ciena;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
    E-mail:  support@ciena.com
    Postal:  7035 Ridge Road
             Hanover, Maryland 21076
             U.S.A.
    Phone:   +1 800-921-1144
    Fax:     +1 410-694-5750";

  description
    "This YANG module defines the nodes for
     the Ciena Management Interface
     and associated notifications for the NETCONF protocol in 6x.

    Copyright (c) 2016 Ciena Corporation.  All rights
         reserved.

         All information contained herein is, and remains
         the property of Ciena Corporation. Dissemination of this
         information or reproduction of this material is strictly
         forbidden unless prior written permission is obtained from
         Ciena Corporation.";

  revision "2016-05-03"
  {
    description
      "Initial version";
    reference
      "RFC 6020: YANG - A Data Modeling Language for
          the Network Configuration Protocol (NETCONF).
          No specific reference; standard not available.";
  }

  /* NOTES/Q:
   *
   * NOTE 1. Supporting only IPv4/IPv6 local and remote interfaces, not ip
interfaces.
   *
   *      6x: muliple ipv6 interfaces
```

```
   *          up to 13 addresses
   *           one ipv4, one link-local, eight ipv6 auto-created,
   *
   *           two manually configured ipv6. (int local/remote add ip <ipv6-
addr>)
   *
   * NOTE 2. local/remote needs to show source of configuration specified by
the ip address.
   *      Manual, SLAAC, DHCP, Local
   *
   *
   * Still need to review 8x version.
   *     8x: differences. Should we add these to mgmt interfaces?
   *         interface backplane
   *         interface shutdown interface
   */

  /*
   * Features
   */

  /*
   * Typedefs
   *
   */

  /*
   * Groupings
   */

  /*
   * Configuration data nodes
   */

  container mgmt-interfaces {
    description
      "Management Interface Configuration in 6x.";

      list interface {
        key "name interface-type";

        description
          "Interface List.
           This is a generic interface to apply to any interface.";

          // name - should be operational data.
          // when oper. data is supported this should be 'config false' here.
          // but its needed as a key to reference different interfaces.
          leaf name {
            type string;
            description
              "The management interface name for this interface.";
          }

          leaf interface-type {
            type enumeration {
              enum local {
                description
                  "The local interface.";
              }
```


Copyright© 2022 Ciena® Corporation

```
              enum remote {
                description
                  "The remote interface.";
              }
              enum ip-interface {
                description
                  "The ip interface.";
              }
              enum loopback {
                description
                  "The loopback interface.";
              }
            }
            description
              "The interface type.";
          }

         // ipv4 and ipv6 can also show 'Not configured' show will that be
reflected here?
          // just means that no leafs will be returned?

            leaf ipv4-address {
              type inet:ip-address;
              description
                "Inet address associated with this interface.";
            }

            leaf ipv4-prefix-length {
              type int32 {
                range "0..32";
              }
              description
                "Specifies the address prefix length.";
            }

            leaf ipv6-primary-address {
              type inet:ip-address;
              description
                "Inet address associated with this interface.";
            }

            leaf ipv6-primary-prefix-length {
              type int32;
              description
                "Specifies the address prefix length.";
            }

            leaf ipv6-secondary-address {
              type inet:ip-address;
              description
                "Inet address associated with this interface.";
            }

            leaf ipv6-secondary-prefix-length {
              type int32;
              description
                "Specifies the address prefix length.";
            }
```


Copyright© 2022 Ciena® Corporation

```
         // assuming only shows manually assigned addresses; the operational
data part will show
          // address added through SLAAC/router advertisements...

          leaf admin-state {
            type ciena:admin-state;
            description
              "The admin state of this management interface.";
          }

          /*
          // This will be removed when code changed to vlan-domain, and vs-
domain nodes.
          leaf domain {
            type int32;
            description
              "Specifies the management interface domain. Is this only for
remote?
               Only found in remote interface.
               Types? unknown, vlan, vs?
               Seems like used for VLAN membership?
               Assuming that virtual-switch also gets set here.
               seems that we set the vlan here";
          }
          */

          /*
          leaf vlan-domain {
            type int32;
            description
              "The vlan membership domain.";
          }

          leaf vs-domain {
            type string;
            description
              "The virtual switch domain.";
          }
          */

          choice domain
          {
            leaf vlan-name {
              type int32;
              description
                "The vlan membership name.";
            }

            leaf vs-name {
              type string;
              description
                "The virtual switch name.";
            }
          }

          leaf priority {
            type int32 {
              range "0..7";
            }
            description
```


Copyright© 2022 Ciena® Corporation

```
              "The 802.1p priority value for the management traffic on this
interface.";
            }

            leaf ingress-acl {
              type string;
              description
             "The name of the ingress ACL attached to the management interface.";
            }

            leaf ipv4-forwarding {
              type ciena:on-off-state;
              description
                "Interface IPv4 Forwarding.";
            }

            leaf mtu {
              type int32;
              description
                "The interface mtu.";
            }

          // need to keep ipv6 part here; assuming that auto-created addresses
are displayed here.
          // and not any manually configured...

            leaf ipv4-gateway {
              type inet:ip-address;
                description
                  "IPv4 Gateway Address.";
            }

            leaf ipv6-gateway {
              type inet:ip-address;
                description
                  "IPv6 Gateway Address.";
            }

        } // end list interface

        container tcp-ip-stack-operational-state {
          description
            "TCP/IP/STACK OPERATIONAL STATE.";

          // interface set ip-forwarding on/off
          leaf ipv4-forwarding {
            type ciena:on-off-state;
            description
              "The forwarding state.";
          }

          // interface set default-dscp
          leaf default-dscp {
            type int32;
            description
              "The default DSCP.";
          }

          // interface set mgmt-port-interface <local/remote>
          leaf mgmt-port-interface {
```


Copyright© 2022 Ciena® Corporation

```
          type enumeration {
            enum local {
              description
                "Local Interface.";
            }
            enum remote {
              description
                "Remote Interface.";
            }
          }
          description
            "This object indicates the way in which traffic at the MGMT port
is used.";
        }

      } // container tcp-ip-stack-operational-state

      // interface ipv4-stack show
      container ipv4-stack-state {
        description
          "IPV4 Stack State.";

        leaf icmp-accept-redirects {
          type ciena:on-off-state;
          description
           "This object indicates the ICMP accept-redirects setting associated
with this address
              type represented by cienaCesInetAddrType.  When on, the
corresponding IP stack will
              act upon received ICMP redirect messages.  When off, the
corresponding IP stack
              will not act upon received ICMP redirect messages.";
        }

        leaf icmp-echo-ignore-broadcasts {
          type ciena:on-off-state;
          description
            "This object indicates the ICMP echo-ignore-broadcasts setting
associated with this address
              type represented by cienaCesInetAddrType.  This setting refers
to whether the device will
              ignore ICMP Echo Request messages which are directed toward a
multicast or broadcast
              destination address.";
        }

        leaf icmp-port-unreachable {
          type ciena:on-off-state;
          description
           "This object indicates the ICMP port-unreachable setting associated
with this address
              type represented by cienaCesInetAddrType.  When on, the
corresponding IP stack will
             send ICMP port-unreachable messages when appropriate to do so.
When off, the
              corresponding IP stack will not send ICMP port-unreachable
messages.";
        }

      }  // end ipv4-stack-state
```


Copyright© 2022 Ciena® Corporation

```
      // interface ipv6-stack show
      container ipv6-stack-state {
        description
          "IPV6 Stack State";

        leaf admin-state {
          type ciena:admin-state;
          description
            "IPv6 Stack Admin State.";
        }

        leaf stack-preference {
          type enumeration {
            enum ipv4 {
              description
                "The ipv4 stack preference.";
            }
            enum ipv6 {
              description
                "The ipv6 stack preference.";
            }
          }
          description
            "Stack Preference.";
        }

        leaf accept-router-advertisements {
          type ciena:on-off-state;
          description
            "Accept Router Advertisements.";
        }

        leaf icmp-accept-redirects {
           type ciena:on-off-state;
           description
          "This object indicates the ICMP accept-redirects setting associated
with this address
              type represented by cienaCesInetAddrType.  When on, the
corresponding IP stack will
              act upon received ICMP redirect messages.  When off, the
corresponding IP stack
              will not act upon received ICMP redirect messages.";
        }

        leaf icmp-echo-ignore-broadcasts {
          type ciena:on-off-state;
          description
            "This object indicates the ICMP echo-ignore-broadcasts setting
associated with this address
             type represented by cienaCesInetAddrType.  This setting refers
to whether the device will
             ignore ICMP Echo Request messages which are directed toward a
multicast or broadcast
             destination address.";
        }

        leaf icmp-port-unreachable {
          type ciena:on-off-state;
          description
```

```
          "This object indicates the ICMP port-unreachable setting associated
with this address
             type represented by cienaCesInetAddrType.  When on, the
corresponding IP stack will
            send ICMP port-unreachable messages when appropriate to do so.
When off, the
             corresponding IP stack will not send ICMP port-unreachable
messages.";
        }

      } // end ipv6-stack-state

      container ipv4-routing-table {
        description
          "The IPv4 Routing Table.";

        list routes {
          key "destination gateway";
          description
            "A IPv4 route entry.";

          leaf destination {
            type union {
              type inet:host;
              type inet:ip-prefix;
            }
            description
              "The destination address.";
          }

          leaf gateway {
            type inet:ip-address;
            description
              "The gateway address.";
          }

          leaf metric {
            type int32 {
              range "1..255";
            }
            description
              "The route metric.";
          }

          leaf interface {
            type enumeration {
              enum none {
                description
                  "None specified.";
              }
              enum local {
                description
                  "Local Interface.";
              }
              enum remote {
                description
                  "Remote Interface.";
              }
            }
            description
```


Copyright© 2022 Ciena® Corporation

```
          "The route interface.";
      }
    }
  }

  container ipv6-routing-table {
    description
      "The IPv6 Routing Table.";

    list routes {
      key "destination gateway";
      description
        "A IPv6 route entry.";

      leaf destination {
        type union {
          type inet:host;
          type inet:ip-prefix;
        }
        description
          "The destination address.";
      }

      leaf gateway {
        type inet:ip-address;
        description
          "The gateway address.";
      }

      leaf metric {
        type int32 {
          range "1..255";
        }
        description
          "The route metric.";
      }

      leaf interface {
        type enumeration {
          enum none {
            description
              "None specified.";
          }
          enum local {
            description
              "Local Interface.";
          }
          enum remote {
            description
              "Remote Interface.";
          }
        }
        description
          "The route interface.";
      }
    }
  }

  // interface tcp-stack show
  container tcp-stack-state {
```


Copyright© 2022 Ciena® Corporation

```
        description
          "This object indicates the current setting of the TCP timestamps
option used by
           the TCP stack of the device.";

        leaf tcp-timestamps {
        type ciena:on-off-state;
        description
          "This object indicates the current setting of the TCP timestamps
option used by
           the TCP stack of the device.";
        }

      } // end container tcp stack state

      container serial-console {
        description
          "Serial Console.";

        leaf serial-login {
          type ciena:admin-state;
          description
            "Serial Login Admin State.";
        }

      }

  } // end container mgmt-interfaces

  /*
   * Operational state data nodes
   */

  container mgmt-interfaces-state {
    config false;
    description
      "Management Interface State for 6x.";

    // local/remote interface.
    //container interface-operational-state {
    //  description
    //    "Operational State of the Interface.";

    list interface {

      description
        "Interface State List.";

      leaf name {
        type string;
        description
          "The management interface name for this interface.";
      }

      leaf index {
        type int32;
        description
          "Interface Index.";
      }
```


Copyright© 2022 Ciena® Corporation

```
leaf oper-state {
  type ciena:oper-state;
  description
    "The oper state of this management interface. ";
}

leaf mac-address {
  type yang:mac-address;
  description
    "Interface Mac Address.";
}

leaf domain-id {
  type int32;
  description
    "Specifies the management interface domain index.
   When domain represents 'vs', this object represents the vs index.
   When domain represents 'vlan , this object represents the vlan id.";
}

leaf domain-vs-name {
  type string;
  description
    "The virtual switch name associated with the domain-id.";
}

leaf mtu {
  type int32;
  description
    "Specifies the interface path MTU.";
}

leaf rtd-index {
  type int32;
  description
    "The interface rtd index.";
}

leaf system-ifindex {
  type int32;
  description
    "The interface system ifindex.";
}

leaf interface-type {
  type string;
  description
    "The interface type.";
}

 leaf ldp-state {
  type ciena:admin-state;
  description
    "The LDP State.";
}

leaf interface-rsvp {
  type ciena:admin-state;
  description
    "Interface RSVP.";
```


Copyright© 2022 Ciena® Corporation

```
      }

      leaf tunnel-entities {
        type string;
        description
          "The interface tunnel entities.";
      }

      leaf l2vpn-association {
        type string;
        description
          "The interface L2VPN Association.";
      }

      leaf vccv-profile-association {
        type string;
        description
          "";
      }

      leaf ip-bfd-session {
        type string;
        description
          "The interface IP BFD Session.";
      }

      leaf ospf {
        type string;
        description
          "The interface OSPF attachment state.";
      }

      leaf isis {
        type string;
        description
          "The interface isis attachment state.";
      }

      leaf static-arp {
        type string;
        description
          "The interface static arp attachment state.";
      }

      leaf ptp {
        type string;
        description
          "The interface PTP attachment state.";
      }

      leaf role {
        type string;
        description
          "The interface role state.";
      }

   // } // end container int-oper-state

    // need to keep ipv6 part here; assuming that auto-created addresses are
displayed here.
```


Copyright© 2022 Ciena® Corporation

```
    // and not any manually configured...

    //container admin-interface {
    //  description
    //    "Operational State of the Admin Interface.";

      // ipv4 and ipv6 can also show 'Not configured' show will that be
reflected here?
      // just means that no leafs will be returned?

      container ipv6 {
        description
          "IPv6 Addresses.";

        list addresses {
          key address;
          description
            "IPv6 Addresses.";

          leaf address {
            type inet:ip-address;
            description
              "Inet address associated with this interface.";
          }

          leaf prefix-length {
            type int32;
            description
              "Specifies the address prefix length.";
          }

        } // end list

      } // end container ipv6



    //} // end container admin interface

    container operational-interfaces {
      description
        "Operational Interface Data.";

      container ipv4 {
        description
          "IPv4";

        leaf address {
          type inet:ip-address;
          description
            "Inet address associated with this interface.";
        }

        leaf prefix-length {
          type int32;
          description
            "Specifies the address prefix length.";
        }

        leaf source {
```


Copyright© 2022 Ciena® Corporation

```
          type string;
          description
            "Address Source.";
        }

      }

      container ipv6 {
        description
          "IPv6";

        list addresses {
          description "IPv6 Addresses.";

          leaf address {
            type inet:ip-address;
            description
              "Inet address associated with this interface.";
          }

          leaf prefix-length {
            type int32;
            description
              "Specifies the address prefix length.";
          }

          leaf source {
            type string;
            description
              "IPv6 Source Address.";
          }

        } // end list addresses

      } // end container ipv6

    } // end oper-interface

  } // end list interface

    container gateway {
      description
        "Operational State of Gateway.";

      container ipv4 {
        description
          "IPv4";

        // Assuming only the manual, and the dhcp addresses here.

        // move to config data area.
        container manual {
          description
            "Manual Set Gateway.";

          /* are these the oper-status states?
             selected(1),
             disabled(2),
             enabled(3),
             unavailable(4),
```


Copyright© 2022 Ciena® Corporation

```
            standby(5)
                  */

          leaf oper-status {
            type ciena:admin-state;
            description
              "Operational Status of the gateway.";
          }

          /* Are these the sources?
                 unknown(1),
                  dhcpv4(2),
                  userConfiguredPrimary(3),
                  userConfiguredBackup(4),
                  auto(5)
                  */

          leaf source {
           type string; // should be enum? what are all the values for source?
            description
              "Gateway Source. How was the gateway set?";
          }

        } // end container manual

        container dhcp {
          description
            "DHCP Gateway.";

          leaf address {
            type inet:ip-address;
            description
              "IPv4 Address.";
          }

          leaf oper-status {
            type ciena:admin-state;
            description
              "Oper Status.";
          }

          /* Are these the sources?
                 unknown(1),
                  dhcpv4(2),
                  userConfiguredPrimary(3),
                  userConfiguredBackup(4),
                  auto(5)
                  */

          leaf source {
           type string; // should be enum? what are all the values for source?
            description
              "Gateway Source.";
          }

        } // end container dhcp

      } // end container ipv4

      container ipv6 {
```


Copyright© 2022 Ciena® Corporation

```
        description
          "IPv6 Address.";

       // IPv6 allows for multiple gateways, but only one manually set gateway.
        // Assuming there will be a bunch of addresses here..

        // assuming that there will be gateway ipv6 addresses added here by
router advertisement.
        // add to config data area;
        // part that will only show the only manually configured address.
        // and here will show any network learned address ; excluding the
manual created address...
        // but still keep this here, it will exist in two places.

        list configuration-addresses {
          description
            "IPV6 GATEWAY CONFIGURATION List.";

        // ??
        // how is the operational state of the manually entered addresses
        // going to be represented here?
        // should there be a container here ;
        // I guess they, the manually entered addresses, will also be shown
here...
        // and can be get/and/set in config data too...


          leaf address {
             type inet:ip-address;
             description
               "IPv6 Address;";
           }

           /* Are theses the state values?
              selected(1),
             disabled(2),
             enabled(3),
             unavailable(4),
             standby(5)
                   */

           leaf oper-state {
             type ciena:oper-state;
             description
               "Address Oper State.";
           }

           leaf source {
           type string; // should be enum? what are all the values for source?
Manual...
             description
               "Address Source.";
           }

        } // end list addresses

        list operational-addresses {
          description
            "IPV6 OPERATIONAL GATEWAY.";
```


Copyright© 2022 Ciena® Corporation

```
          leaf address {
             type inet:ip-address;
             description
               "IPv6 Address.";
           }

           leaf interface {
             type string; // should this be an enum? local/remote/...
             description
               "The gateway the interace is using.";
           }

           /*      unknown(1),
                   dhcpv4(2),
                   userConfiguredPrimary(3),
                   userConfiguredBackup(4),
                   auto(5)
                   */
           leaf source {
           type string; // should be enum? what are all the values for source?
             description
               "How the gateway address was learned.";
           }

           leaf metric {
             type int32;
             description
               "Gateway metric.";
           }

       } // end list addresses

     } // end container ipv6
   } // end container gateway
 } // end container mgmt-interfaces-state

 /*
  * RPC Operations
  */

 // <rpc template>

 /*
 rpc <template> {
   description
     "";

   input {

   }
   output {
     leaf status {
       type string;
       description
         "Result of operation.";
     }
   }
 }
 */
```


Copyright© 2022 Ciena® Corporation

```
/*
 * Notifications
 */
// <notification template>

notification mgmt-address-change-notification {
  description
   "A address-change-notification is sent when the
      inet address of the associated interface changes.";

  container mgmt-address-change-notification {
    description
      "Management Address Change Notification.";

    leaf index {
      type int32;
      description
        "Index.";
    }

    leaf name {
      type string;
      description
        "Interface Name.";
    }

    leaf address {
      type inet:ip-address;
      description
        "Interface Address.";
    }

    leaf prefix-length {
      type int32;
      description
        "Address Prefix Length.";
    }

    leaf global-mac-address {
      type yang:mac-address;
      description
        "Global Mac Address.";
    }

    leaf global-severity {
      type string; // need to check what severity this is?
      description
        "Global Severity.";
    }
  }
} // end notification address-change-notification

notification mgmt-address-added-notification {
  description
   "A address-added-notification is sent when a new interface
       is added.";

  container mgmt-address-added-notification  {
    description
      "Management Address Added Notification.";
```


Copyright© 2022 Ciena® Corporation

```
      leaf index {
        type int32;
        description
          "Index.";
      }

      leaf name {
        type string;
        description
          "Interface Name.";
      }

      leaf address {
        type inet:ip-address;
        description
          "Interface Address.";
      }

      leaf prefix-length {
        type int32;
        description
          "Interface Prefix Length.";
      }

      leaf global-mac-address {
        type yang:mac-address;
        description
          "Global Mac Address.";
      }

      leaf global-severity {
        type string; // need to check what severity this is?
        description
          "Global Severity.";
      }

    }
  } // end notification address-added-notification

  notification mgmt-address-removed-notification {
    description
     "A address-removed-notification is sent when an interface
         is removed.";

    container mgmt-address-removed-notification {
      description
        "Management Address Removed Notification.";

      leaf index {
        type int32;
        description
          "Index.";
      }

      leaf name {
        type string;
        description
          "Interface Name.";
      }
```


Copyright© 2022 Ciena® Corporation

```
      leaf address {
        type inet:ip-address;
        description
          "Interface Address.";
      }

      leaf prefix-length {
        type int32;
        description
          "Address Prefix Length.";
      }

      leaf global-mac-address {
        type yang:mac-address;
        description
          "Global Mac Address.";
      }

      leaf global-severity {
        type string; // need to check what severity this is?
        description
          "Global Severity.";
      }
    }
  } // end notification mgmt-address-removed-notification

  notification gateway-address-change-notification {
    description
     "A gateway-address-change-notification is sent
        when the address associated with the gateway is changed.";

    container gateway-address-change-notification {
      description
        "Gateway Address Change Notification.";

       leaf address {
        type inet:ip-address;
        description
          "Interface Address.";
      }

      leaf prefix-length {
        type int32;
        description
          "Address Prefix Length.";
      }

      leaf global-mac-address {
        type yang:mac-address;
        description
          "Global Mac Address.";
      }

      leaf global-severity {
        type string; // need to check what severity this is?
        description
          "Global Severity.";
      }
    }
```


Copyright© 2022 Ciena® Corporation

```
} // end notification gateway-address-change-notification

notification gateway-address-add-notification {
  description
   "A gateway-address-add-notification is sent
      when a new gateway is added.";

  container gateway-address-add-notification {
    description
      "Gateway Address Add Notification.";

    leaf address {
      type inet:ip-address;
      description
        "Gateway Address.";
    }

    leaf prefix-length {
      type int32;
      description
        "Address Prefix Length.";
    }

    leaf global-mac-address {
      type yang:mac-address;
      description
        "Global Mac Address.";
    }

    leaf global-severity {
      type string; // need to check what severity this is?
      description
        "Global Severity.";
    }
  }
} // end notification gateway-address-add-notification

notification gateway-address-removed-notification {
  description
   "A gateway-address-removed-notification is sent
      when a gateway is removed.";

  container gateway-address-removed-notification  {
    description
      "Gateway Address Removed Notification.";

    leaf address {
      type inet:ip-address;
      description
        "Gateway Address.";
    }

    leaf prefix-length {
      type int32;
      description
        "Address Prefix Length.";
    }

    leaf global-mac-address {
      type yang:mac-address;
```


Copyright© 2022 Ciena® Corporation

```
        description
          "Global Mac Address.";
      }

      leaf global-severity {
        type string; // need to check what severity this is?
        description
          "Global Severity.";
      }
    }
  } // end notification gateway-address-removed-notification


  /*
  notification <rpc template> {
    description
     "";

    container <rpc template> {
      description
        "";
    }
  }
  */

} // end ciena template model
```


Copyright© 2022 Ciena® Corporation

ciena-mef-access-flow.yang

```
module ciena-mef-access-flow {
   namespace "urn:ciena:params:xml:ns:yang:ciena-pn::ciena-mef-access-flow";
   prefix "acl";

   import ciena-mef-classifier {
      prefix "classifier";
   }

   import ciena-mef-logical-port {
      prefix "mef-logical-port";
   }

   import ciena-mef-fp {
      prefix "mef-fp";
   }

   organization
      "Ciena Corporation";

   contact
       "Web URL: http://www.ciena.com/
       E-mail:  yang@ciena.com
       Postal:  7035 Ridge Road
                Hanover, Maryland 21076
                U.S.A.
       Phone:   +1 800-921-1144
       Fax:     +1 410-694-5750";

   description
      "This YANG module defines Access Flows and Access Profiles.";

   revision 2017-06-20 {
      description "Initial revision";
   }

/*
 * typedefs
 */
   typedef access-flow-ref {
      type leafref {
         path "/acl:access-flows/acl:access-flow/acl:name";
      }
      description
         "This type is used by the data models that needs to reference
          configured Access flows.";
   }

   typedef access-profile-ref {
      type leafref {
         path "/acl:access-profiles/acl:access-profile/acl:name";
      }
      description
         "This type is used by the data models that needs to reference
          configured Access Profile.";
   }

   typedef acl-action-ref {
      type leafref {
```


Copyright© 2022 Ciena® Corporation

```
         path "/acl:acl-actions/acl:acl-action/acl:name";
      }
      description
         "This type is used by the data models that needs to reference
          configured ACL actions.";
   }

   grouping  acl-action-destination {
      choice destination {
         case logical-port-list {
            leaf-list logical-port {
               type mef-logical-port:logical-port-ref;
            }
            description
                "one or more logical-ports which act as destination for the
acl-action (e.g. mirror/redirect)";
         }

         case flow-point-list {
            leaf-list flow-point {
               type mef-fp:fp-ref;
            }
            description
              "one or more flow-points which act as destination for the acl-
action (e.g. mirror/redirect)";
         }
      }
   }


   /*
    * Configuration.
    */
   container acl-actions {

      list acl-action {
         key "name";

         leaf name {
            type string;

            description
               "A unique string that is either system assigned or assigned
                by the user but does not change over its life.";
         }

         leaf description {
            type string;
            description
               "A more detailed description that an operator can use
                to describe the action.";
         }

         leaf action {
            type enumeration {
               enum redirect {
                  description
                    "Redirect the classified flow to one or more ports/flow-
points etc.";
               }
```


Copyright© 2022 Ciena® Corporation

```
                enum ingress-mirror {
                   description
                     "Ingress-Mirror the classified flow to one or more ports/
flow-points etc.";
                }
             }
             description
                "Specifies an action for this acl action";
          }

          uses acl:acl-action-destination {
             when "action = 'redirect' or action = 'ingress-mirror'";
          }
       }
    }

    container access-flows {

       list access-flow {
          key "name";

          leaf name {
             type string;

             description
                "A unique string that is either system assigned or assigned
                 by the user but does not change over its life.";
          }

          leaf description {
             type string;
             description
                "A more detailed description that an operator can use
                 to describe the flow.";
          }

          leaf classifier {
             type classifier:classifier-ref;
             description
                "A reference to a list of classifier entries.";
          }

          leaf classifier-precedence {
             type uint32;
             description
                "A precedence value for the access flow. Lower values take
                 precedence over higher values";
          }

          choice parent-interface {

             case none {
                leaf none {
                   type empty;
                }
                description
                   "When the Access-Flow is an ACL rule in an Access-Profile,
use none.
                    In this case, the access-profile is referenced by a parent-
interface instead.";
```


Copyright© 2022 Ciena® Corporation

```
            }

            case logical-port {
               leaf-list parent-port {
                  type mef-logical-port:logical-port-ref;
               }
               description
                  "This is used when the Access-Flow is an ACL rule that is
not part of an Access-Profile.
                   In this case, the parent-interface defines the port that
the Access-Flow applies to";
            }

            case flow-point {
               leaf-list parent-fp {
                  type mef-fp:fp-ref;
               }
               description
                  "This is used when the Access-Flow is an ACL rule that is
not part of an Access-Profile.
                   In this case, the parent-interface defines the flow-point
that the Access-Flow applies to";
            }
         }

         leaf filter-action {

            type enumeration {
               enum deny {
                  description
                     "Drop traffic for traffic that matches this specific
access-flow-rule.
                   This may be augmented with specific actions in the action
list for this traffic flow";
               }
               enum allow {
                  description
                  "Do not drop traffic for traffic that matches this specific
access-flow-rule.
                   This may be augmented with specific actions in the action
list for this traffic flow";
               }
            }

            default deny;

            description
               "Specifies the filter-action for this Access-Flow/ACL-Rule";
         }

         leaf-list augment-action {
               type acl:acl-action-ref;
               description
                   "A list of Actions that may augment the filter-action of
the Access-Flow/ACL-Rule.";
         }

         leaf stats-collection {
             type enumeration {
                enum on;
```


Copyright© 2022 Ciena® Corporation

```
                enum off;
              }
              description
                "Determines whether stats collection will be turned on or not
for an access-flow";
          }
       }
   }

   container access-profiles {

      list access-profile {
         key "name";

         leaf name {
            type string;

            description
               "A unique string that is either system assigned or assigned
                by the user but does not change over its life.";
         }

         leaf description {
            type string;
            description
               "A more detailed description that an operator can use
                to describe the profile.";
         }

         leaf-list access-flow-rule {
               type acl:access-flow-ref;
               description
                "A list of ACL rules, Access-Flows that are part of an access-
profile.
                 Each ACL-rule/Access-Flow specifies classifiers, precedence
and actions independently";
         }

         leaf default-filter-action {

            type enumeration {
               enum deny {
                  description
                    "Drop traffic for all other traffic that does not match
a specific access-flow-rule but
                       classifies to this Access-Profile";
               }
               enum allow {
                  description
                    "Do not drop traffic for all other traffic that does not
match a specific access-flow-rule but
                       classifies to this Access-Profile";
               }
            }

            default deny;

            description
              "Specifies the default filter-action for this Access-Profile";
         }
```


Copyright© 2022 Ciena® Corporation

```
         leaf stats-collection {
             type enumeration {
                enum on;
                enum off;
             }
             description
               "Determines whether stats collection will be turned on or not
instance of an access-profile";
         }

         description
            "A list of all access-profile configuration entries.";
      }
   }

   /*
    * State.
    */
   container access-flows-state {
      config false;

      description
         "Access Flow operational data for all Access Flows.";

      list access-flow {
         key "name";
         description
            "The operational data for this Access-Flow.";

         leaf name {
            type string;
            description
               "A string that identifies the Access-Flow.";
         }

         leaf hitBytes {
            type uint64;
            description
               "byte count of frames that hit the Access-Flow";
         }

         leaf hitFrames {
            type uint64;
            description
               "frame count of frames that hit the Access-Flow";
         }
      }
   }

   container acl-actions-state {
      config false;

      description
         "ACL action operational data for all ACL Augment Actions.";

      list access-action {
         key "name";
         description
            "The operational data for this ACL Augment Action.";
```


Copyright© 2022 Ciena® Corporation

```
         leaf name {
            type string;
            description
               "A string that identifies the Access Augment Action.";
         }

         leaf number-of-attached-rules {
            type uint16;
            description
            "Number of rules the augment action instance is associated with.";
         }
      }

      leaf defined-acl-actions {
         type uint16;
         description
            "Number of defined augment action instances on the device.";
      }

      leaf remaining-acl-actions-available {
         type uint16;
         description
            "Number of available augment action instances on the device.";
      }
   }
}
```


Copyright© 2022 Ciena® Corporation

ciena-mef-classifier-deviation.yang

```
module ciena-mef-classifier-deviation {

    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-mef-classifier-
dev";
    prefix "ciena-mef-classifier-dev";

    import ciena-mef-classifier { prefix ciena-mef-classifier; }

    revision 2017-05-16 {
        description "Added the correct deviation path.";
    }

    revision 2015-07-14 {
        description "Initial revision.";
    }

    //
    //  Config data that is not supported in SAOS 6.14
    //

    deviation /ciena-mef-classifier:classifiers/ciena-mef-
classifier:classifier/ciena-mef-classifier:filter-entry/ciena-mef-
classifier:filter-parameters/ciena-mef-classifier:mpls-label-stack/ciena-
mef-classifier:mpls-labels {
        deviate not-supported;
    }

    //
    //  Config data that is not supported in SAOS 6.15
    //
    deviation /ciena-mef-classifier:classifiers/ciena-mef-
classifier:classifier/ciena-mef-classifier:filter-operation {
        deviate not-supported;
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-mef-classifier.yang

```
module ciena-mef-classifier {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn::ciena-mef-classifier";
    prefix "classifier";

    import ietf-inet-types {
        prefix "inet";
    }

    import ietf-yang-types {
        prefix "yt";
    }

    revision 2017-05-16 {
        description
            "Various updates to classifers since first revision";
    }

    revision 2014-12-19 {
        description
            "First revision of MEF based classifier";
    }

    /*
     * Typedefs
     */
    typedef classifier-ref {
        type leafref {
            path "/classifier:classifiers/classifier:classifier/
classifier:name";
        }
        description
          "This type is used by data models that need to reference
           configured classifier list.";
    }

    /*
     * Feature
     */
    feature filter-parameters-type {
        description
            "Allows for configuration of filter parameters inline.";
    }

    identity filter-param-type {
        description
            "This is the base identity of the filter-param-type.";
    }

    identity vtag-stack {
        base filter-param-type;
    }

    identity mpls-label {
        base filter-param-type;
    }

    identity dscp {
        base filter-param-type;
```


Copyright© 2022 Ciena® Corporation

```
}

identity source-ip {
    base filter-param-type;
}

identity destination-ip {
    base filter-param-type;
}

identity l4-destination-port {
    base filter-param-type;
}

identity l4-source-port {
    base filter-param-type;
}

identity ip-protocol {
    base filter-param-type;
}

identity base-etype {
    base filter-param-type;
}

identity any {
    base filter-param-type;
}

feature vtag-stack-feature {
    description
        "Allows for configuration of vtag stack type inline.";
}

identity vtag-stack-type {
    description
        "This is the base identity for vtag stack.";
}

identity untagged {
    base vtag-stack-type;
}

identity l2cp {
    base vtag-stack-type;
}

identity tagged {
    base vtag-stack-type;
}

identity ip-fragment {
    base filter-param-type;
}

identity l4-application {
    base filter-param-type;
}
```


Copyright© 2022 Ciena® Corporation

```
    identity tcp-flags {
        base filter-param-type;
    }

    identity l4-dst-protocol {
        base filter-param-type;
    }

    identity source-mac {
        base filter-param-type;
    }

    identity destination-mac {
        base filter-param-type;
    }

    grouping classifier-group {
        list classifier {
            ordered-by user;    // User sets the order of this list.
            key "name";

            leaf name {
                type string;
                description
                    "A unique name for the classifier.";
            }

            leaf filter-operation {
                type enumeration {
                    enum match-all;
                    enum match-any;
                }
            }

            list filter-entry {
                key "filter-parameter";

                ordered-by user;

                leaf filter-parameter {
                    type identityref {
                        base filter-param-type;
                    }
                    description
                        "Indicates which filter parameter is used by this
filter entry";
                }

                leaf logical-not {
                    type boolean;
                    default "false";
                    description
                        "Opposite of what is specified in the
                         filter-parameters. If the filter-parameter
                         specifies a tpid as tpid-8100, then anything
                         other than tpid-8100 is considered an acceptable
                         packet.";
                }

                choice filter-parameters {
```


Copyright© 2022 Ciena® Corporation

```
                    case vtag-stack {
                        choice vtag-stack-type {
                            case untagged {
                                leaf untagged-exclude-priority-tagged {
                                    type boolean;
                                }
                            }

                            case l2cp {
                                leaf l2cp-exclude-priority-tagged {
                                    type boolean;
                                }
                            }

                            case tagged {
                                list vtags {
                                    key "tag";

                                    leaf tag {
                                        type uint8;
                                        description
                                            "'1' represents outer most tag,
'2' next outer most, etc";
                                    }

                                    leaf tpid {
                                        type enumeration {
                                            enum tpid-8100;
                                            enum tpid-88a8;
                                            enum tpid-9100;
                                        }

                                    }

                                    leaf pcp {
                                        type uint8 {
                                            range "0..7";
                                        }
                                    }

                                    leaf pcp-mask {
                                        type uint8 {
                                            range "1..7";
                                        }
                                        description
                                          "Allow PCP values to be optionally
coupled with a mask in a single classifier";
                                    }

                                    leaf dei {
                                        type enumeration {
                                            enum discard-eligible;
                                            enum not-discard-eligible;
                                        }
                                        description
                                           "Discard Eligibility Indication";
                                    }

                                    leaf vlan-id {
                                        type uint16 {
```


Copyright© 2022 Ciena® Corporation

```
                                            range "1..4094";
                                        }
                                    }

                                    leaf vlan-id-max {
                                       type uint16 {
                                          range "1..4094";
                                       }
                                       description
                                          "The maximum value of VLAN ID for
ranged VLAN-ID values. Mutually exclusive to vlan-id-mask";
                                    }

                                    leaf vlan-id-mask {
                                       type uint16 {
                                          range "1..4095";
                                       }
                                       description
                                     "Allow VLAN ID values to be optionally
coupled with a mask. Mutually exclusive to vlan-id-max";
                                    }
                                }
                            }
                        }
                    }

                    case mpls-label-stack {
                        list mpls-labels {
                      key "label"; // Need a key. It cannot be a choice param.

                            leaf label {
                                type uint32;
                            }
                            choice labels {
                                case any {
                                    leaf label-any {
                                        type empty;
                                    }
                                    description
                                        "Accept any label.";
                                }

                                case value {
                                    leaf mpls-label {
                                        type uint32;
                                    }
                                }
                            }

                            choice tc {
                                case any {
                                    leaf tc-any {
                                        type empty;
                                    }
                                    description
                                        "Accept any tc value.";
                                }

                                case value {
                                    leaf tc-value {
```


Copyright© 2022 Ciena® Corporation

```
                                        type uint8;
                                    }
                                }
                            }
                        }
                    }

                    case dscp {
                        leaf dscp-min {
                            type inet:dscp;
                            description
                                "The minimum value of DSCP.";
                        }
                        leaf dscp-max {
                            type inet:dscp;
                            description
                                "The maximum value of DSCP for ranged DSCP
values in a single classifier. Mutually exclusive to dscp-mask";
                        }
                        leaf dscp-mask {
                            type inet:dscp;
                            description
                                "Allow DSCP values to be optionally coupled
with a mask in a single classifier. Mutually exclusive to dscp-max";
                        }
                    }

                    case source-ip {
                        leaf source-address {
                            type inet:ip-prefix;
                        }
                        description
                           "Classification on IP source-address (v4/v6) and
masking.";
                    }

                    case destination-ip {
                        leaf destination-address {
                            type inet:ip-prefix;
                        }
                        description
                          "Classification on IP destination-address (v4/v6)
and masking.";
                    }

                    case l4-source-port {
                        leaf source-min {
                            type inet:port-number;
                        }
                        leaf source-max {
                            type inet:port-number;
                        }
                    }

                    case l4-destination-port {
                        leaf destination-min {
                            type inet:port-number;
                        }
                        leaf destination-max {
                            type inet:port-number;
```


Copyright© 2022 Ciena® Corporation

```
        }
    }

    case ip-protocol {
        leaf min-prot {
            type uint16;
        }
        leaf max-prot {
            type uint16;
        }
    }

    case base-etype {
        leaf base-ethertype {
            type uint16;
        }
    }

    case any {
        leaf any {
            type empty;
            description
           "Accept any classification. Wide-Open classifier";
        }
    }

    case ip-fragment {
        leaf ip-fragment {
            type boolean;
            description
                "IP-fragment bit true/false";
        }
    }

    case l4-application {
        leaf l4-application {
           type enumeration {
              enum twamp;
           }
        }
    }

    case l4-dst-protocol {
        leaf l4-dst-protocol {
          type enumeration {
            enum bootp-client {
              description "bootp-client.";
            }
            enum bootp-server {
              description "bootp-server.";
            }
            enum bgp {
              description "bgp.";
            }
            enum dhcpv6-client {
              description "dhcpv6 client.";
            }
            enum dhcpv6-server {
              description "dhcpv6 server.";
            }
```


Copyright© 2022 Ciena® Corporation

```
                enum dhcp-client {
                  description "dhcp client.";
                }
                enum dhcp-server {
                  description "dhcp server.";
                }
                enum dns {
                  description "dns.";
                }
                enum ftp {
                  description "ftp.";
                }
                enum http {
                  description "http.";
                }
                enum ldp {
                  description "ldp.";
                }
                enum ntp {
                  description "ntp.";
                }
                enum olsr {
                  description "olsr.";
                }
                enum rip {
                  description "rip.";
                }
                enum rpc {
                  description "rpc.";
                }
                enum snmp {
                  description "snmp.";
                }
                enum snmptrap {
                  description "snmp trap.";
                }
                enum ssh {
                  description "ssh.";
                }
                enum syslog {
                  description "syslog.";
                }
                enum tacacs {
                  description "tacacs.";
                }
                enum telnet {
                  description "telnet.";
                }
                enum tftp {
                  description "tftp.";
                }
                enum twampctl {
                  description "twampctl.";
                }
              }
              description
                "L4 protocol.";
            }
        }
```


Copyright© 2022 Ciena® Corporation

```
                    case tcp-flags {
                        leaf syn {
                          type boolean;
                        }
                        leaf ack {
                          type boolean;
                        }
                        leaf rst {
                          type boolean;
                        }
                        leaf urg {
                          type boolean;
                        }
                        leaf psh {
                          type boolean;
                        }
                        leaf fin {
                          type boolean;
                        }
                        description
                          "TCP flags.";
                    }

                    case source-mac {
                        leaf source-mac {
                            type yt:mac-address;
                        }
                        leaf source-mac-mask {
                            type yt:mac-address;
                        }
                    }

                    case destination-mac {
                        leaf destination-mac {
                            type yt:mac-address;
                        }
                        leaf destination-mac-mask {
                            type yt:mac-address;
                        }
                    }
                }
            }
        }
    }

    container classifiers {
       uses classifier-group;
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-mef-cos-to-frame-map-deviation.yang

```
module ciena-mef-cos-to-frame-map-deviation {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-mef-cos-to-frame-
map-dev";
    prefix "ctf-dev";

    import ciena-mef-cos-to-frame-map {
        prefix ctf;
    }

    revision 2017-07-24 {
        description "Remove restriction of characters in cos-to-frame-map
name";
    }

    revision 2016-09-20 {
        description "Restrict invalid characters in cos-to-frame-map name";
    }

    revision 2015-07-16 {
        description "Initial revision";
    }

    deviation /ctf:cos-to-frame-maps/ctf:cos-to-frame-map/ctf:description {
        deviate not-supported;
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-mef-cos-to-frame-map.yang

```
module ciena-mef-cos-to-frame-map {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-mef-cos-to-frame-
map";
    prefix "ctf";

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
                 Hanover, Maryland 21076
                 U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
        "This YANG module defines the CoS to Frame Map. A combination of
         cos and color yields a pcp and dei or ip-dscp or mpls-tc";

    revision 2015-07-16 {
        description "Initial revision";
    }

    /*
     * Typedefs
     */
    typedef cos-to-frame-ref {
        type leafref {
            path "/ctf:cos-to-frame-maps/ctf:cos-to-frame-map/ctf:name";
        }
        description
          "This type is used by data models that need to reference
           configured CoS to frame maps.";
    }

    container cos-to-frame-maps {

        list cos-to-frame-map {
            description
                "A list of profiles that can contain several map entries.";

            key "name";

            leaf name {
                type string;

                description
                   "A string used to uniquely identify a profile entry within
                     the list of maps (profiles).";
            }

            leaf description {
                type string;
                description
                    "A more detailed description of the map.";
            }
```

```
            list map-entry {
                key "cos color";

                leaf cos {
                    type uint8 {
                        range "0..63";
                    }
                    description
                        "Class of Service bits.";
                }

                leaf color {
                    type enumeration {
                        enum green;
                        enum yellow;
                    }
                }

                leaf pcp {
                    type uint8 {
                        range "0..7";
                    }
                    description
                      "The frame's VLAN tag priority bits will be set to this
                       value if the frame's assigned cos and color matches this
                           map instance. ";
                }

                leaf dei {
                    type enumeration {
                        enum enabled;
                        enum disabled;
                    }
                    description
                        "Drop Eligibility Indication";
                }

                leaf ip-dscp {
                    type uint8 {
                        range "0..63";
                    }
                    description
                        "IP DiffServ Code Point value. ";
                }

                leaf mpls-tc {
                    type uint8 {
                        range "0..7";
                    }
                    description
                        "MPLS Traffic Class (TC) bits.";
                }
            }
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-mef-egress-qos-deviation.yang

```
module ciena-mef-egress-qos-deviation {

    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-mef-egress-qos-
dev";
    prefix "ciena-mef-egress-qos-dev";

    import ciena-mef-egress-qos {
        prefix mef-egress-qos;
    }

    revision "2017-07-24" {
      description
       "Remove deviation added to restrict characters in interface name";
    }

    revision "2016-09-20" {
      description
       "Deviation added to restrict invalid characters in interface name";
    }

    revision "2016-02-09" {
        description
            "Initial version";
    }

    deviation /mef-egress-qos:egress-qos/mef-egress-qos:cos-queue-map/mef-
egress-qos:cos-count {
        deviate not-supported;
    }

    deviation /mef-egress-qos:egress-qos/mef-egress-qos:cos-queue-map/mef-
egress-qos:map-entry/mef-egress-qos:green-wred-curve {
        deviate not-supported;
    }

    deviation /mef-egress-qos:egress-qos/mef-egress-qos:cos-queue-map/mef-
egress-qos:map-entry/mef-egress-qos:yellow-wred-curve {
        deviate not-supported;
    }

    deviation /mef-egress-qos:egress-qos/mef-egress-qos:queue-group-profile/
mef-egress-qos:queue/mef-egress-qos:cir-percent {
        deviate not-supported;
    }

    deviation /mef-egress-qos:egress-qos/mef-egress-qos:queue-group-profile/
mef-egress-qos:queue/mef-egress-qos:eir-percent {
        deviate not-supported;
    }

    deviation /mef-egress-qos:egress-qos/mef-egress-qos:scheduler-profile/
mef-egress-qos:cir-percent {
        deviate not-supported;
    }

    deviation /mef-egress-qos:egress-qos/mef-egress-qos:scheduler-profile/
mef-egress-qos:eir-percent {
        deviate not-supported;
```


Copyright© 2022 Ciena® Corporation

```
    }

    deviation /mef-egress-qos:egress-qos/mef-egress-qos:scheduler-profile/
mef-egress-qos:cir-policy {
        deviate not-supported;
    }

    deviation /mef-egress-qos:egress-qos/mef-egress-qos:scheduler-profile/
mef-egress-qos:eir-policy {
        deviate not-supported;
    }

    deviation /mef-egress-qos:egress-qos/mef-egress-qos:queue-group/mef-
egress-qos:instance-id {
        deviate not-supported;
    }

   deviation /mef-egress-qos:egress-qos/mef-egress-qos:scheduler/mef-egress-
qos:instance-id {
        deviate not-supported;
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-mef-egress-qos.yang

```
module ciena-mef-egress-qos {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn::ciena-mef-egress-qos";
    prefix "mef-egress-qos";

    import ciena-mef-logical-port {
          prefix "mef-logical-port";
    }

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
         E-mail:  yang@ciena.com
         Postal:  7035 Ridge Road
                  Hanover, Maryland 21076
                  U.S.A.
         Phone:   +1 800-921-1144
         Fax:     +1 410-694-5750";

    description
        "This YANG module defines Ciena's configuration of
         queues and schedulers for Egress QoS.";

    revision "2016-02-09" {
        description
            "Initial version";
    }

    typedef cos-queue-map-ref {
        type leafref {
          path "/mef-egress-qos:egress-qos/mef-egress-qos:cos-queue-map/mef-
egress-qos:name";
        }

        description
            "This type is used by the data models that needs to reference
             configured CoS queue maps.";
    }

    typedef congestion-avoidance-profile-ref {
        type leafref {
            path "/mef-egress-qos:egress-qos/mef-egress-qos:congestion-
avoidance-profile/mef-egress-qos:name";
        }

        description
            "This type is used by the data models that needs to reference
             configured congestion avoidance profiles.";
    }

    typedef queue-group-profile-ref {
        type leafref {
            path "/mef-egress-qos:egress-qos/mef-egress-qos:queue-group-
profile/mef-egress-qos:name";
        }

        description
```


Copyright© 2022 Ciena® Corporation

```
            "This type is used by the data models that needs to reference
             configured queue group profiles.";
    }

    typedef queue-group-indirection-ref {
        type leafref {
            path "/mef-egress-qos:egress-qos/mef-egress-qos:queue-group-
indirection/mef-egress-qos:name";
        }

        description
            "This type is used by the data models that needs to reference
             queue-groups indirectly (e.g. where the root port changes
dynamically).";
    }

    typedef scheduler-profile-ref {
        type leafref {
          path "/mef-egress-qos:egress-qos/mef-egress-qos:scheduler-profile/
mef-egress-qos:name";
        }

        description
            "This type is used by the data models that needs to reference
             configured scheduler-profiles.";
    }

    typedef scheduler-ref {
        type leafref {
            path "/mef-egress-qos:egress-qos/mef-egress-qos:scheduler/mef-
egress-qos:name";
        }

        description
            "This type is used by the data models that needs to reference
             configured scheduler.";
    }

    grouping eqos-parameters-group {
        description
          "Egress QoS parameters used for queue group profiles and scheduler
profiles.";

        leaf cir {
            type uint32;

            description
               "Committed information rate for aggregate shaping in kilobits
per second.";
        }

        leaf cir-percent {
             type uint32;

             description
            "Committed information rate for aggregate shaping on a percentage
basis <0..100>.";
        }

        leaf cbs {
```


Copyright© 2022 Ciena® Corporation

```
            type uint32;

            description
                "Committed burst size in kilobytes.";
        }

        leaf eir {
            type uint32;

            description
               "Maximum rate for aggregate shaping in kilobits per second.";
        }

        leaf eir-percent {
            type uint32;

            description
                "Maximum rate for aggregate shaping on a percentage basis
<0..100>.";
        }

        leaf ebs {
            type uint32;

            description
                "Maximum burst size in kilobytes.";
        }
    }

    container egress-qos {

        list cos-queue-map {
            key "name";

            leaf name {
                type string;

                description
                    "Name for the CoS queue map.";
            }

            leaf cos-count {
                type uint32;

                description
                     "Number of CoS values supported (e.g. 8 or 64)";
            }

            list map-entry {
                key "cos";

                leaf cos {
                    type uint32 {
                        range "0..63";
                    }
                    description
                         "Internal CoS to use for mapping to a Queue, green-
wred-curve and yellow-wred curve.";
                }
```


Copyright© 2022 Ciena® Corporation

```
                leaf queue {
                    type uint32;

                    description
                         "Queue mapped to by internal CoS";
                }

                leaf green-wred-curve {
                    type uint32;

                    description
                         "WRED curve relative to a Queue to use for traffic
whose internal color is Green";
                }

                leaf yellow-wred-curve {
                    type uint32;

                    description
                         "WRED curve relative to a Queue to use for traffic
whose internal color is Yellow";
                }
            }
        }

        list congestion-avoidance-profile {
            key "name";

            leaf name {
                type string;

                description
                    "Name for the profile.";
            }

            leaf type {
                type enumeration {
                    enum wred;
                    enum red;
                }
            }
        }

        list queue-group-profile {
            description
                "Defines the characteristics of an instance of a group of
queues.";

            key "name";

            leaf name {
                type string;

                description
                    "Name for the profile.";
            }

            leaf queue-count {
                type uint32;
```


Copyright© 2022 Ciena® Corporation

```
                description
                 "Number of queues in the associated queue-group instances.";
            }

            leaf cos-queue-map {
                type cos-queue-map-ref;

                description
                    "cos-queue map to use for mapping CoS to a queue.";
            }

            list queue {
                key "queue-number";

                leaf queue-number {
                    type uint32;
                    mandatory true;

                    description
                        "The queue-number for this queue. The priority and
weight for this queue are determined via
                         the mapping of queue-number to tap-point number
(typically 1:1) from the associated scheduler-profile
                         of the parent scheduler-instance.";
                }

                leaf congestion-avoidance-profile {
                    type congestion-avoidance-profile-ref;

                    description
                   "Optional congestion avoidance algorithm for this queue.";
                }

                uses eqos-parameters-group;

            }
        }

        list scheduler-profile {
            key "name";

            leaf name {
                type string;

                description
                    "Name for the profile.";
            }

            leaf tap-point-count {
                type uint32;

                description
                    "The number of tap-points in corresponding scheduler-
instances that queues
                     or child schedulers can uniquely use to attach to the
scheduler.";
            }

            leaf scheduling-algorithm {
                type enumeration {
```


Copyright© 2022 Ciena® Corporation

```
                    enum sp {
                        description "Strict priority";
                    }
                    enum rr {
                        description "Round-robin";
                    }
                    enum wrr {
                        description "Weighted round-robin";
                    }
                    enum wfq {
                        description "Weighted fair queueing";
                    }
                    enum wdrr {
                        description "Weighted deficit round-robin";
                    }
                }

                description
                  "The scheduling algorithm used to service the tap-points of
                     the scheduler instances";
            }

            uses eqos-parameters-group;

            leaf cir-policy {
                type enumeration {
                    enum auto-adjust-disabled {
                        description
                        "cir-policy of auto-adjust-disabled determines that
cir (not cir-percent) is used for the scheduler instance associated
                             with this scheduler-profile and also the child
queues/scheduler-instances use cir.";
                    }
                    enum cir-as-percent {
                        description
                       "cir-policy of cir-as-percent determines that cir in
percentage values are used for the scheduler instance associated
                             with this scheduler-profile and also the child
queues/scheduler-instances use cir-percent";
                    }
                    enum child-cir-as-percent {
                        description
                        "cir-policy of child-cir-as-percent determines that
cir (not cir-percent) is used for the scheduler instance associated
                         with this scheduler-profile, but cir in percentage
values are used for the child queues/scheduler-instances";
                    }
                    enum child-cir-sum {
                        description
                       "cir-policy of child-cir-sum determines that cir for
the scheduler instance associated
                             with this scheduler-profile is calculated based
on the sum of cir of the child queues/scheduler-instances";
                    }
                }
            }

            leaf eir-policy {
                type enumeration {
                    enum auto-adjust-disabled {
```


Copyright© 2022 Ciena® Corporation

```
                        description
                        "eir-policy of auto-adjust-disabled determines that
eir (not eir-percent) is used for the scheduler instance associated
                           with this scheduler-profile and also the child
queues/scheduler-instances use eir.";
                    }
                    enum eir-as-percent {
                        description
                       "eir-policy of eir-as-percent determines that eir in
percentage values are used for the scheduler instance associated
                        with this scheduler-profile and also the child
queues/scheduler-instances use eir-percent";
                    }
                    enum child-eir-as-percent {
                        description
                        "eir-policy of child-eir-as-percent determines that
eir (not eir-percent) is used for the scheduler instance associated
                         with this scheduler-profile, but eir in percentage
values are used for the child queues/scheduler-instances";
                    }
                }
            }

            list tap-point {
                description
                    "Defines priority and weight for each tap-point of the
associated scheduler instances.
                     Either Queues or Schedulers map to the tap points and
their priority and weight is obtained
                     from this tap-point configuration.";

                key "number";

                leaf number {
                    type uint32;

                    description
                    "tap-point identifier within a scheduler-instance";
                }

                leaf priority {
                    type uint32;

                    description
                   "Priority of this tap-point relative to other tap-points
in a scheduler-instance. Used when scheduling algorithm is strict-priority";
                }

                leaf weight {
                    type uint32;

                    description
                  "Weight of this tap-point relative to other tap-points in
a scheduler-instance. Used when scheduling algorithm is
                    weighted-round-robin, weighted-fair-queuing, weighted-
deficit-round-robin";
                }
            }
        }
```


Copyright© 2022 Ciena® Corporation

```
        list queue-group {
            description
             "The queue-group is an instance of a group of queues as described
by the associated queue-group-profile. The queue-group
                 instance has a parent scheduler instance which is part of a
scheduling hierarchy rooted to a port.
               Additionally, the queue-group instance may be mapped to both
directly (e.g. from flow-points) and indirectly via the
                queue-group-indirection (e.g. via MPLS PWs/MPLS Tunnels)";

            key "name";

            leaf name {
                type string;

                description
                    "Name for the queue group instance.";
            }

            leaf queue-group-profile {
                type queue-group-profile-ref;
                mandatory true;

                description
                  "The queue-group-profile describing the attributes of this
queue-group instance.";
            }

            leaf instance-id
            {
                type uint32;

                description
                    "Identifier for the scheduler-instance.";
            }

            leaf queue-group-indirection {
                type queue-group-indirection-ref;

                description
                 "The queue-group-indirection that this queue-group instance
is assigned to. Multiple different queue-groups
                      on different ports can be assigned to the same queue-
group-indirection";
            }

            leaf parent-scheduler {
                type scheduler-ref;
                mandatory true;

                description
                    "The parent-scheduler instance for this queue group.";
            }
        }

        list scheduler {
            description
               "The scheduler is an instance of a scheduler as described by
the associated scheduler-profile. The scheduler
```


Copyright© 2022 Ciena® Corporation

```
                instance itself has either a parent scheduler instance which
is part of a scheduling hierarchy rooted to a port or
                 the root port itself.";

            key "name";

            leaf name {
                type string;

                description
                    "Name for the scheduler instance.";
            }

            leaf scheduler-profile {
                type scheduler-profile-ref;
                mandatory true;

                description
                   "The scheduler-profile describing the attributes of this
scheduler instance.";
            }

            leaf instance-id
            {
                type uint32;

                description
                    "Identifier for the scheduler-instance.";
            }

            leaf parent-scheduler {
                type scheduler-ref;

                description
                   "Optional reference to parent-scheduler instance if this
scheduler instance is not the root scheduler instance
                    in the egress QoS hierarchy of a port.";
            }

            leaf parent-tap-point {
                type uint32;

                description
                    "The tap point that this scheduler instance in a parent
scheduler-instance. Determines the priority and weight
                    for this scheduler instance relative to its parent from
the scheduler-profile.";
            }

            leaf parent-port {
                type mef-logical-port:logical-port-ref;

                description
                    "Optional reference to a logical-port if this scheduler
instance is the root scheduler instance
                     in the egress QoS hierarchy of a port";
            }
        }

        list queue-group-indirection {
```


Copyright© 2022 Ciena® Corporation

```
            description
                "This is an indirected identification of a queue group that
would be used by entities
                 which do not resolve directly to the root of the egress-qos
hierarchy (a port).
                An example of this is an MPLS PW which maps to a queue-group-
indirection for service-based queuing and scheduling.
               Dependent on the state of the underlying transport (e.g. MPLS
Tunnels to next-hops to port), the actual queue-group instance
                 being used may change dynamically, however the queue-group-
indirection allows";

            key "name";

            leaf name {
                type string;

                description
                    "Name for the queue group indirection.";
            }

            leaf indirection-id
            {
                type uint32;

                description
                    "Identifier for the queue group indirection.";
            }
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-mef-fd-deviation.yang

```
module ciena-mef-fd-deviation {

    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-mef-fd-dev";
    prefix "ciena-mef-fd-dev";

    import ciena-mef-fd {
        prefix mef-fd;
    }

    revision "2016-01-18" {
        description
            "Initial version";
    }

    deviation /mef-fd:fds/mef-fd:fd/mef-fd:l2cp-profile {
        deviate not-supported;
    }

    deviation /mef-fd:fds/mef-fd:fd/mef-fd:flood-containment-profile {
        deviate not-supported;
    }

    deviation /mef-fd:fds/mef-fd:fd/mef-fd:description {
        deviate replace {
            type string {
                length "0..128";
            }
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-mef-fd.yang

```
module ciena-mef-fd {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-mef-fd";
    prefix "mef-fd";

    import ciena-mef-l2cp-profile {
        prefix "mef-l2cp";
    }

    import ciena-mef-flood-containment-profile {
        prefix "mef-fc";
    }

    import ciena-mef-egress-qos {
        prefix "mef-egress-qos";
    }

    import ciena-mef-pfg-profile {
        prefix "mef-pfg";
    }

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
         E-mail:  yang@ciena.com
         Postal:  7035 Ridge Road
                  Hanover, Maryland 21076
                  U.S.A.
         Phone:   +1 800-921-1144
         Fax:     +1 410-694-5750";

    description
        "This YANG module defines Ciena's configuration of
         the Forwarding Domain (Bridge Domain, Virtual Switch,
         VPLS representation for the NETCONF protocol).";

    revision "2017-08-31" {
        description
            "Added reference to Private Forwarding Group Profile for a FD";
        reference
            "RFC 6020: YANG - A Data Modeling Language for
             the Network Configuration Protocol (NETCONF).
             No specific reference; standard not available.";
    }

    revision "2016-09-25" {
        description
            "Added fd-state for operational data";
      reference
          "RFC 6020: YANG - A Data Modeling Language for
              the Network Configuration Protocol (NETCONF).
              No specific reference; standard not available.";
    }

    revision "2016-01-25" {
        description
            "Initial version";
```


Copyright© 2022 Ciena® Corporation

```
}

/*
 * Features
 */

/*
 * Configuration data nodes
 */
container fds {

    list fd {
        key "name";

        description
            "The list of configured forwarding domains on the device.";

        leaf name {
            type string;

            description
                "An administratively assigned string, which may be used
                 to identify the forwarding domain.";
        }

        leaf description {
            type string;
            description
               "This is string used to describe the Forwarding Domain.";
        }

        leaf mode {
            type enumeration {
                enum vlan;
                enum vpls;
                enum vpws;
            }
        }

        leaf vlan-id {
            type uint16 {
                range "1..4094";
            }
            mandatory true;
            when "../mode='vlan'";
        }

        leaf mac-learning {
            type enumeration {
                enum enabled;
                enum disabled;
            }
        }

        leaf l2cp-profile {
            type mef-l2cp:l2cp-profile-ref;
            description
             "Reference to a Layer 2 Control Protocol Tunneling Profile.";
        }
```


Copyright© 2022 Ciena® Corporation

```
            leaf flood-containment-profile {
                type mef-fc:flood-containment-profile-ref;
                description
                    "Reference to a Flood Containment Profile definition.";
            }

        leaf pfg-profile {
                type mef-pfg:pfg-profile-ref;
                description
                    "Reference to a Private Forwarding Group Profile.";
            }

            leaf cos-queue-map {
                type mef-egress-qos:cos-queue-map-ref;
                description
               "Reference to queue-map for hierarchical shaping/scheduling.";
            }

            leaf queue-group-indirection {
                type mef-egress-qos:queue-group-indirection-ref;
                description
                    "Reference to queue-group for hierarchical shaping/
scheduling.";
            }
        }
    }

    container fds-state {
            // Currently there is no global FD oper data, so this container
            // is for augemenation purposes
            // Global FD oper data may be augmented here

        config false;

        description
            "Global FD operational data.";

        list fd {
            // Currently there is no per FD oper data, so this container
            // is for augemenation purposes
            // Per FD oper data may be augmented here

            key "name";

            description
                "The operational data for this FD.";

            leaf name {
                type string;

                description
                    "A string that identifies the forwarding domain.";
            }
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-mef-flood-containment-profile-deviation.yang

```
module ciena-mef-flood-containment-profile-deviation {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-mef-flood-
containment-profile-dev";
    prefix "mef-fc-dev";

    import ciena-mef-flood-containment-profile {
        prefix mef-fc;
    }

    revision 2018-01-09 {
        description "Removed the restriction for a profile to have one
containment for 6.x systems";
    }

    revision 2016-05-18 {
        description "Restrict a profile to one containment for 6.x systems";
    }

    revision 2015-05-13 {
        description "Initial revision";
    }

    deviation /mef-fc:flood-containment-profiles/mef-fc:flood-containment-
profile/mef-fc:description {
        deviate not-supported;
    }

}
```


Copyright© 2022 Ciena® Corporation

ciena-mef-flood-containment-profile.yang

```
module ciena-mef-flood-containment-profile {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-mef-flood-
containment-profile";
    prefix "mef-fc";

   organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
         E-mail:  yang@ciena.com
         Postal:  7035 Ridge Road
                  Hanover, Maryland 21076
                  U.S.A.
         Phone:   +1 800-921-1144
         Fax:     +1 410-694-5750";

    description
        "This YANG module defines Ciena's configuration of
         the Flood Containment Profile.";

    revision 2016-01-13 {
        description "Initial revision";
    }

    typedef flood-containment-profile-ref {
        type leafref {
          path "/mef-fc:flood-containment-profiles/mef-fc:flood-containment-
profile/mef-fc:name";
        }
        description
            "This type is used by the data models that need to reference
             configured Flood Containment Profiles.";
    }

    container flood-containment-profiles {

        list flood-containment-profile {
            key "name";

            leaf name {
                type string {
                    length "1..1024";
                }
                description
                    "An administratively assigned string, which may be used
                     to identify the profile.";
            }

            leaf description {
                type string;
                description
                "This is a user-defined string used to describe the profile.";
            }

       list containment {
                key "frame-type";
```


Copyright© 2022 Ciena® Corporation

```
                leaf frame-type {
                    type bits {
                        bit unknown-unicast;
                        bit unknown-l2-multicast;
                        bit unknown-ip-multicast;
                        bit known-multicast;
                        bit broadcast;
                    }
                    description
                      "The frame types to be contained with the given rate.";
                }

                leaf rate {
                    type uint64;
                    description
                        "Data rate in kbps.";
                }
        }
         }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-mef-fp-deviation.yang

```
module ciena-mef-fp-deviation {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-mef-fp-dev";
    prefix "ciena-mef-fp-dev";

    import ciena-mef-fp {
        prefix "mef-fp";
    }

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
                 Hanover, Maryland 21076
                 U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
        "This YANG module defines Ciena's deviations to ciena-mef-fp.";

    revision 2019-04-05 {
        description
          "Fix xpath in deviation statement by including path of choice";
    }

    revision "2016-01-28" {
        description
            "Initial Version.";
        reference
          "UNI and EVC Definition of Managed Objects (MEF 40), January 2013";
    }

    deviation /mef-fp:fps/mef-fp:fp/mef-fp:description {
        deviate not-supported;
    }

    deviation /mef-fp:fps/mef-fp:fp/mef-fp:type/mef-fp:mpls-pw {
        deviate not-supported;
    }

    deviation /mef-fp:fps/mef-fp:fp/mef-fp:type/mef-fp:other {
        deviate not-supported;
    }

    deviation /mef-fp:fps/mef-fp:fp/mef-fp:mtu-size {
        deviate not-supported;
    }

    deviation /mef-fp:fps/mef-fp:fp/mef-fp:admin-state {
        deviate not-supported;
    }

   deviation /mef-fp:fps/mef-fp:fp/mef-fp:ingress-l2-transform/mef-fp:frame-
type/mef-fp:stack/mef-fp:vlan-stack {
        deviate not-supported;
```


Copyright© 2022 Ciena® Corporation

```
    }

   deviation /mef-fp:fps/mef-fp:fp/mef-fp:ingress-l2-transform/mef-fp:frame-
type/mef-fp:untagged/mef-fp:untagged-tpid {
        deviate not-supported;
    }

   deviation /mef-fp:fps/mef-fp:fp/mef-fp:ingress-l2-transform/mef-fp:frame-
type/mef-fp:untagged/mef-fp:untagged-pcp {
        deviate not-supported;
    }

   deviation /mef-fp:fps/mef-fp:fp/mef-fp:ingress-l2-transform/mef-fp:frame-
type/mef-fp:untagged/mef-fp:untagged-dei {
        deviate not-supported;
    }

    deviation /mef-fp:fps/mef-fp:fp/mef-fp:egress-l2-transform/mef-fp:frame-
type/mef-fp:stack/mef-fp:vlan-stack/mef-fp:action/mef-fp:push/mef-fp:push-
tpid {
        deviate not-supported;
    }

    deviation /mef-fp:fps/mef-fp:fp/mef-fp:egress-l2-transform/mef-fp:frame-
type/mef-fp:stack/mef-fp:vlan-stack/mef-fp:action/mef-fp:push/mef-fp:push-
pcp {
        deviate not-supported;
    }

    deviation /mef-fp:fps/mef-fp:fp/mef-fp:egress-l2-transform/mef-fp:frame-
type/mef-fp:stack/mef-fp:vlan-stack/mef-fp:action/mef-fp:push/mef-fp:push-
dei {
        deviate not-supported;
    }

    deviation /mef-fp:fps/mef-fp:fp/mef-fp:egress-l2-transform/mef-fp:frame-
type/mef-fp:stack/mef-fp:vlan-stack/mef-fp:action/mef-fp:push/mef-fp:push-
vid {
        deviate not-supported;
    }

    deviation /mef-fp:fps/mef-fp:fp/mef-fp:egress-l2-transform/mef-fp:frame-
type/mef-fp:stack/mef-fp:vlan-stack/mef-fp:action/mef-fp:stamp/mef-fp:stamp-
tpid {
        deviate not-supported;
    }

    deviation /mef-fp:fps/mef-fp:fp/mef-fp:egress-l2-transform/mef-fp:frame-
type/mef-fp:stack/mef-fp:vlan-stack/mef-fp:action/mef-fp:stamp/mef-fp:stamp-
pcp {
        deviate not-supported;
    }

    deviation /mef-fp:fps/mef-fp:fp/mef-fp:egress-l2-transform/mef-fp:frame-
type/mef-fp:stack/mef-fp:vlan-stack/mef-fp:action/mef-fp:stamp/mef-fp:stamp-
vid/mef-fp:no-op {
        deviate not-supported;
    }
```


Copyright© 2022 Ciena® Corporation

```
    deviation /mef-fp:fps/mef-fp:fp/mef-fp:egress-l2-transform/mef-fp:frame-
type/mef-fp:stack/mef-fp:vlan-stack/mef-fp:action/mef-fp:stamp/mef-fp:stamp-
vid/mef-fp:vid-value {
        deviate not-supported;
    }

    deviation /mef-fp:fps/mef-fp:fp/mef-fp:egress-l2-transform/mef-fp:frame-
type/mef-fp:untagged/mef-fp:untagged-tpid {
        deviate not-supported;
    }

    deviation /mef-fp:fps/mef-fp:fp/mef-fp:egress-l2-transform/mef-fp:frame-
type/mef-fp:untagged/mef-fp:untagged-pcp {
        deviate not-supported;
    }

    deviation /mef-fp:fps/mef-fp:fp/mef-fp:egress-l2-transform/mef-fp:frame-
type/mef-fp:untagged/mef-fp:untagged-dei {
        deviate not-supported;
    }

    deviation /mef-fp:fps/mef-fp:fp/mef-fp:egress-l2-transform/mef-fp:frame-
type/mef-fp:untagged/mef-fp:untagged-vid {
        deviate not-supported;
    }

    deviation /mef-fp:fps/mef-fp:fp/mef-fp:ingress-l3-transform {
        deviate not-supported;
    }

    deviation /mef-fp:fps/mef-fp:fp/mef-fp:egress-l3-transform {
        deviate not-supported;
    }

    deviation /mef-fp:fps/mef-fp:fp/mef-fp:frame-to-cos {
        deviate not-supported;
    }

    deviation /mef-fp:fps/mef-fp:fp/mef-fp:flood-containment-profile {
        deviate not-supported;
    }

    deviation /mef-fp:fps/mef-fp:fp/mef-fp:mac-learning {
        deviate not-supported;
    }

    deviation /mef-fp:fps/mef-fp:fp/mef-fp:classifier-list-precedence {
        deviate not-supported;
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-mef-fp.yang

```
module ciena-mef-fp {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-mef-fp";
    prefix "mef-fp";

    import ciena-mef-fd {
        prefix "mef-fd";
    }

    import ciena-mef-logical-port {
        prefix "mef-logical-port";
    }

    import ciena-mef-classifier {
        prefix "classifier";
    }

    import ciena-mef-cos-to-frame-map {
        prefix "ctf";
    }

    import ciena-mef-frame-to-cos-map {
        prefix "ftc";
    }

    import ciena-mef-flood-containment-profile {
        prefix "mef-fc";
    }

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
                 Hanover, Maryland 21076
                 U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
        "This YANG module defines Ciena's management data definition for the
     management of fp.";

    revision "2016-01-28" {
        description
            "Initial Version.";
        reference
           "UNI and EVC Definition of Managed Objects (MEF 40), January 2013";
    }

/*
 * typedefs
 */

  typedef fp-ref {
     type leafref {
        path "/mef-fp:fps/mef-fp:fp/mef-fp:name";
```


Copyright© 2022 Ciena® Corporation

```
      }
      description
         "This type is used by the data models that need to reference
          configured flow points";
  }

  typedef vlan-id {
      type uint16 {
        range "1..4094";
      }
      description
          "Represents a IEEE 802.1Q VLAN-ID.";
  }

/*
 * features
 */

/*
 * groupings
 */
    grouping transform {

        choice frame-type {
            case stack {
                list vlan-stack {
                    key "tag";

                    leaf tag {
                        type uint8;
          description
                            "'1' represents outer most tag, '2' next outer
most, etc";
                    }

                    choice action {
                        case push {
                            leaf push-tpid {
                                type enumeration {
                                    enum tpid-8100 {
                                        value 33024;
                                    }
                                    enum  tpid-88a8 {
                                        value 34984;
                                    }
                                    enum tpid-9100 {
                                        value 37120;
                                    }
                                }
                                mandatory true;
                            }

                            leaf push-pcp {
                                type enumeration {
                                    enum pcp-0 {
                                        value 0;
                                    }
                                    enum pcp-1 {
                                        value 1;
                                    }
```


Copyright© 2022 Ciena® Corporation

```
                    enum pcp-2 {
                        value 2;
                    }
                    enum pcp-3 {
                        value 3;
                    }
                    enum pcp-4 {
                        value 4;
                    }
                    enum pcp-5 {
                        value 5;
                    }
                    enum pcp-6 {
                        value 6;
                    }
                    enum pcp-7 {
                        value 7;
                    }
                    enum map;
                }
            }

            leaf push-dei {
                type enumeration {
                    enum enabled;
                    enum disabled;
                }
            }

            leaf push-vid {
                type vlan-id;
                mandatory true;
            }
        }

        case pop {
            leaf pop-type {
                type empty;
            }
        }

        case stamp {
            leaf stamp-tpid {
                type enumeration {
                    enum no-op;
                    enum tpid-8100 {
                        value 33024;
                    }
                    enum  tpid-88a8 {
                        value 34984;
                    }
                    enum tpid-9100 {
                        value 37120;
                    }
                }
            }

            leaf stamp-pcp {
                type enumeration {
                    enum pcp-0 {
```


Copyright© 2022 Ciena® Corporation

```
                                value 0;
                            }
                            enum pcp-1 {
                                value 1;
                            }
                            enum pcp-2 {
                                value 2;
                            }
                            enum pcp-3 {
                                value 3;
                            }
                            enum pcp-4 {
                                value 4;
                            }
                            enum pcp-5 {
                                value 5;
                            }
                            enum pcp-6 {
                                value 6;
                            }
                            enum pcp-7 {
                                value 7;
                            }
                            enum no-op;
                            enum map;
                        }
                    }

                    leaf stamp-dei {
                        type enumeration {
                            enum no-op;
                            enum enabled;
                            enum disabled;
                        }
                    }

                    choice stamp-vid {
                        case no-op {
                            leaf no-op {
                                type empty;
                            }
                        }
                        case vid-value {
                            leaf stamp-vid-value {
                                type vlan-id;
                            }
                        }
                    }
                }
            }
        }

        case untagged {

            leaf untagged-tpid {
                type enumeration {
                    enum tpid-8100 {
                        value 33024;
                    }
```


Copyright© 2022 Ciena® Corporation

```
                        enum  tpid-88a8 {
                            value 34984;
                        }
                        enum tpid-9100 {
                            value 37120;
                        }
                    }
                }

                leaf untagged-pcp {
                    type enumeration {
                        enum pcp-0 {
                            value 0;
                        }
                        enum pcp-1 {
                            value 1;
                        }
                        enum pcp-2 {
                            value 2;
                        }
                        enum pcp-3 {
                            value 3;
                        }
                        enum pcp-4 {
                            value 4;
                        }
                        enum pcp-5 {
                            value 5;
                        }
                        enum pcp-6 {
                            value 6;
                        }
                        enum pcp-7 {
                            value 7;
                        }
                        enum map;
                    }
                }

                leaf untagged-dei {
                    type enumeration {
                        enum enabled;
                        enum disabled;
                    }
                }

                leaf untagged-vid {
                    type vlan-id;
                }
            }
        }
    }


/*
 * Configuration.
 */
    container fps {

        list fp {
```


Copyright© 2022 Ciena® Corporation

```
key "name";
description
    "A list of all mef-fp configuration entries.";

leaf name {
    type string;
    description
        "This object indicates the flow point identifier.
        The identifier is a text string that is used to identify
         a flow point.

        Unique string values are chosen to uniquely identify the
         flow point.

         Octet values of 0x00 through 0x1f are illegal.

         MEF 26.1 restricts the maximum size identifiers to 45
         octets.";
   reference
       "[MEF 6.1] 6.1; [MEF 7.2] 6.2.1.3";
}

leaf description {
    type string;
    description
        "A editable string used to describe this entry.";
}

leaf fd-name {
    type leafref {
        path "/mef-fd:fds/mef-fd:fd/mef-fd:name";
    }
}

leaf logical-port {
    type mef-logical-port:logical-port-ref;
}

choice type {
    case q-in-q {
        leaf svlan {
            type uint32;
        }
    }
    case mpls-pw {
        leaf mpls-pw {
                type empty;
        }
    }
    case uni {
        leaf uni {
            type empty;
        }
    }
    case other {
        leaf other {
                type empty;
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

```
leaf mtu-size {
    type uint32;
    units "octets";
    description
       "This object indicates the configured EVC maximum service
        frame format size. It must be less than or equal to the
      max-mtu-size. Vendors may choose to go beyond this limit.";
    reference
        "[MEF 6.1] 6.1; [MEF 7.2] 6.2.1.3";
    default "2000";
}


leaf admin-state {
    type enumeration {
        enum enabled;
        enum disabled;
    }
}

list ingress-l2-transform {
    key "ingress-name";

    leaf ingress-name {
        type string;
        description
            "Need a key for this list. It cannot be
             a choice of several objects but objects
             that will always be specified.";
    }

    uses transform;
}

list egress-l2-transform {
    key "egress-name";

    leaf egress-name {
        type string;
        description
            "Need a key for this list. It cannot be
             a choice of several objects but objects
             that will always be specified.";
    }

    uses transform;
}

choice ingress-l3-transform {
    case map {
        leaf ingress-l3-mapped {
            type empty;
        }
    }
    case remark-dscp {
        leaf ingress-remark-dscp-value {
            type uint8 {
                range "0..63";
            }
```


Copyright© 2022 Ciena® Corporation

```
            }
        }
    }

    choice egress-l3-transform {
        case map {
            leaf egress-l3-mapped {
                type empty;
            }
        }
        case remark-dscp {
            leaf egress-remark-dscp-value {
                type uint8 {
                    range "0..63";
                }
            }
        }
    }

    choice frame-to-cos {
        case map {
            leaf map-policy {
                type enumeration {
                    enum outer-tag;
                    enum inner-tag;
                    enum mpls-tc;
                    enum dscp;
                }
            }

            leaf frame-to-cos-map {
                type ftc:frame-to-cos-ref;
            }
        }

        case fixed {
            leaf cos {
                type uint8 {
                    range "0..63";
                }
            }

            leaf color {
                type enumeration {
                    enum green;
                    enum yellow;
                    enum red;
                }
            }
        }
    }

    leaf cos-to-frame-map {
        type ctf:cos-to-frame-ref;
    }

    leaf flood-containment-profile {
        type mef-fc:flood-containment-profile-ref;
        description
            "Reference to a Flood Containment Profile definition.";
```


Copyright© 2022 Ciena® Corporation

```
            }

            leaf-list classifier-list {
                type classifier:classifier-ref;
            }

            leaf classifier-list-precedence {
                type uint32;
            }

            leaf mac-learning {
                type enumeration {
                    enum enabled;
                    enum disabled;
                }
            }
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-mef-frame-to-cos-map-deviation.yang

```
module ciena-mef-frame-to-cos-map-deviation {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-mef-frame-to-cos-
map-dev";
    prefix "ftc-dev";

    import ciena-mef-frame-to-cos-map {
        prefix ftc;
    }

    revision 2017-08-09 {
       description "Remove Restriction of invalid characters in frame-to-cos-
map name";
    }

    revision 2016-09-20 {
        description "Restrict invalid characters in frame-to-cos-map name";
    }

    revision 2015-07-16 {
        description "Initial revision";
    }

    deviation /ftc:frame-to-cos-maps/ftc:frame-to-cos-map/ftc:description {
        deviate not-supported;
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-mef-frame-to-cos-map.yang

```
module ciena-mef-frame-to-cos-map {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-mef-frame-to-cos-
map";
    prefix "ftc";

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
                 Hanover, Maryland 21076
                 U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
        "This YANG module defines the Frame to Cos Map.";

    revision 2015-07-16 {
        description "Initial revision";
    }

    /*
     * Typedefs
     */
    typedef frame-to-cos-ref {
        type leafref {
            path "/ftc:frame-to-cos-maps/ftc:frame-to-cos-map/ftc:name";
        }
        description
          "This type is used by data models that need to reference
           configured frame to CoS maps.";
    }

    container frame-to-cos-maps {

        list frame-to-cos-map {
            description
                "A list of profiles that can contain several map entries.";

            key "name";

            leaf name {
                type string;
                description
                    "A string used to uniquely identify a list of profiles.";
            }

            leaf description {
                type string;
                description
                    "A more detailed description of the map.";
            }

            list map-entry {
                key "name";
```


Copyright© 2022 Ciena® Corporation

```
leaf name {
    type string;

    description
       "A string used to uniquely identify a particular map
        entry within the profile.";
}

choice frame-type {
    case vlan-tag {
        leaf pcp {
            type uint8 {
                range "0..7";
            }
        }
        leaf dei {
            type enumeration {
                enum enabled;
                enum disabled;
            }
        }
        description
        "For layer 2 frames, the VLAN tag's priority bits and
       dei value are used as key fields to derive an assigned
            cos and color for the frame";
    }

    case ip {
        leaf ip-dscp {
            type uint8 {
                range "0..63";
            }
        }
        description
            "The frame a IP frame.";
    }

    case mpls {
        leaf mpls-tc {
            type uint8 {
                range "0..7";
            }
        }
    }
}

leaf cos {
    type uint8 {
        range "0..63";
    }
    description
        "Class of Service bits.";
}

leaf color {
    type enumeration {
        enum green;
        enum yellow;
        enum red;
```


Copyright© 2022 Ciena® Corporation

```
                    }
                    description
                        "The assigned color for the frame. A color of green
                         indicates the frame is conformant with CIR,
                        a yellow indicates it is not conformant with CIR but
                         is conformant with EIR and red means it is not
                       conformant with EIR or CIR and thus will be dropped.";
                }
            }
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-mef-l2cp-profile-deviation.yang

```
module ciena-mef-l2cp-profile-deviation {
   namespace "urn:ciena:params:xml:ns:yang:ciena-pn::ciena-mef-l2cp-profile-
dev";
    prefix "mef-l2cp-dev";

    import ciena-mef-l2cp-profile {
        prefix mef-l2cp;
    }

    revision 2015-05-13 {
        description "Initial revision";
    }

    deviation /mef-l2cp:l2cp-profiles/mef-l2cp:l2cp-profile/mef-
l2cp:description {
        deviate not-supported;
    }

    deviation /mef-l2cp:l2cp-profiles/mef-l2cp:l2cp-profile/mef-l2cp:fixed-
cos {
        deviate not-supported;
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-mef-l2cp-profile.yang

```
module ciena-mef-l2cp-profile {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn::ciena-mef-l2cp-
profile";
    prefix "mef-l2cp";

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
         E-mail:  yang@ciena.com
         Postal:  7035 Ridge Road
                  Hanover, Maryland 21076
                  U.S.A.
         Phone:   +1 800-921-1144
         Fax:     +1 410-694-5750";

    description
        "This YANG module defines Ciena's configuration of
         the L2 Control Protocol Profile.";

    revision 2015-05-13 {
        description "Initial revision";
    }

    typedef l2cp-profile-ref {
        type leafref {
         path "/mef-l2cp:l2cp-profiles/mef-l2cp:l2cp-profile/mef-l2cp:name";
        }
        description
            "This type is used by the data models that need to reference
             configured L2 Control Protocol Profiles.";
    }

    typedef DispositionType {
        type enumeration {
            enum discard;
            enum peer;
            enum forward;
        }
    }

    container l2cp-profiles {

        list l2cp-profile {
            key "name";

            leaf name {
                type string {
                }
                description
                    "An administratively assigned string, which may be used
                     to identify the profile.";
            }

            leaf description {
                type string;
                description
```


Copyright© 2022 Ciena® Corporation

```
                "This is a user-defined string used to describe the profile.";
            }

            leaf untagged-cos-policy {
                type enumeration {
                    enum fixed {
                        description
                         "This policy assigns fixed resolved and frame CoS to
                            forwarded untagged L2 control frames.";
                    }
                    enum ignore {
                        description
                        "This policy doesn't assign explicit CoS to forwarded
                          untagged L2 control frames. Instead it allows the
control
                       frames to pick resolved and frame CoS in datapath just
                            like untagged data frames.";
                    }
                }
                description
                  "This denotes CoS policy for forwarded untagged L2 control
frames.";
            }

            leaf fixed-cos {
                type uint8 {
                    range "0..63";
                }
                description
                 "This denotes fixed resolved CoS and frame CoS for forwarded
                    untagged L2 control frames.";
            }

            list protocol-disposition {

                key "protocol";

                leaf protocol {
                    type enumeration {
                        enum xstp {
                            description "STP/RSTP/MSTP protocols. MAC DA 01-
80-C2-00-00-00.";
                        }
                        enum pause {
                          description "PAUSE protocol. Ethertype 0x8808. MAC
DA 01-80-C2-00-00-01.";
                        }
                        enum lacp {
                           description "LACP protocol. Ethertype 0x8809/01.
MAC DA 01-80-C2-00-00-02.";
                        }
                        enum lamp {
                           description "LAMP protocol. Ethertype 0x8809/02.
MAC DA 01-80-C2-00-00-02.";
                        }
                        enum link-oam {
                           description "LINK OAM protocol. Ethertype 0x8809/
03. MAC DA 01-80-C2-00-00-02.";
                        }
                        enum port-auth {
```


Copyright© 2022 Ciena® Corporation

```
                        description "Port Authentication protocol. Ethertype
0x888E. MAC DA 01-80-C2-00-00-03.";
                        }
                        enum e-lmi {
                          description "E-LMI protocol. Ethertype 0x88EE. MAC
DA 01-80-C2-00-00-07.";
                        }
                        enum lldp {
                           description "LLDP protocol. Ethertype 0x88CC. MAC
DA 01-80-C2-00-00-0E.";
                        }
                        enum ptp-peer-delay {
                            description "PTP Peer-Delay protocol. Ethertype
0x88F7. MAC DA 01-80-C2-00-00-0E.";
                        }
                        enum esmc {
                           description "ESMC protocol. Ethertype 0x8809/0A.
MAC DA 01-80-C2-00-00-02.";
                        }
                        enum garp-block {
                            description "GARP/MRP block range. MAC DA 01-80-
C2-00-00-20 to 01-80-C2-00-00-2F.";
                        }
                        enum bridge-rsvd-0b0f {
                           description "Bridge reserved range. MAC DA 01-80-
C2-00-00-0B/0F.";
                        }
                        enum bridge-rsvd-0c0d {
                           description "Bridge reserved range. MAC DA 01-80-
C2-00-00-0C/0D.";
                        }
                        enum bridge-block {
                           description "Bridge block range. MAC DA 01-80-C2-
00-00-00 to 01-80-C2-00-00-0A and 01-80-C2-00-00-0E.";
                        }

                    }
                }

                leaf untagged-disposition {
                    type DispositionType;
                    description
                       "Disposition assigned to untagged form of L2 control
protocol PDUs.";
                }

                leaf tagged-disposition {
                    type DispositionType;
                    description
                        "Disposition assigned to tagged form of L2 control
protocol PDUs.";
                }
            }
        }

    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-mef-logical-port-deviation.yang

```
module ciena-mef-logical-port-deviation {

   namespace "urn:ciena:params:xml:ns:yang:ciena-pn::ciena-mef-logical-port-
dev";
    prefix "mef-logical-port-dev";

    import ciena-mef-logical-port {
        prefix mef-logical-port;
    }

    revision "2015-07-15" {
        description
            "Initial version";
    }

    deviation /mef-logical-port:logical-ports/mef-logical-port:logical-port/
mef-logical-port:mtu {
        deviate replace {
            type uint32 {
                range "512..12288";
            }
            default 1526;
        }
    }

    deviation /mef-logical-port:logical-ports/mef-logical-port:logical-port/
mef-logical-port:description {
        deviate not-supported;
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-mef-logical-port.yang

```
module ciena-mef-logical-port {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn::ciena-mef-logical-
port";
    prefix "mef-logical-port";

    import ietf-interfaces {
        prefix "if";
    }

    import ietf-yang-types {
        prefix "yt";
    }

    import ciena-mef-frame-to-cos-map {
        prefix "ftc";
    }

    import ciena-mef-cos-to-frame-map {
        prefix "ctf";
    }

    import ciena-mef-meter-profile {
        prefix "meter";
    }

    import ciena-mef-flood-containment-profile {
        prefix "mef-fc";
    }

    // import l2cp-profile {
    //     prefix "l2cp";
    // }

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
                 Hanover, Maryland 21076
                     U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
        "This YANG module defines Ciena's management data definition for the
         management of a Logical Port.";

    revision "2017-09-18" {
        description
          "Added operational data for logical ports.";
    }

    revision "2014-10-02" {
        description
          "Initial Version.";
    }
```


Copyright© 2022 Ciena® Corporation

```
/*
 * typedefs
 */
    typedef logical-port-ref {
        type leafref {
            path "/mef-logical-port:logical-ports/mef-logical-port:logical-
port/mef-logical-port:name";
        }
        description
            "This type is used by the data models that needs to reference
             configured logical ports.";
    }

   typedef lp-admin-state
   {
      type enumeration
      {
         enum disable;
         enum enable;
      }
   }

   typedef lp-oper-state
   {
      type enumeration
      {
         enum up;
         enum down;
      }
   }

/*
 * features
 */

/*
 * Configuration model.
 */
    container logical-ports {

        list logical-port {
            key "name";

            leaf name {
                type string;
                description
                    "This object indicates the identifier and is a
                     text string that is used to identify a logical port.
                     Unique string values are chosen to uniquely identify
                     the port.

                     Octet values of 0x00 through 0x1f are illegal.

                     MEF 26.1 restricts the maximum size identifiers to
                     45 octets.";
                 reference
                   "[MEF 7.2] 6.2.1.4";
            }
```


Copyright© 2022 Ciena® Corporation

```
leaf admin-state {
   type lp-admin-state;
   description "Enable or disable this logical-port";
}

leaf binding {
type if:interface-ref;
}

leaf mtu {
    type uint32 {
        range "64..9216";
    }
    default 1526;
    description
        "The size in bytes of the maximum transmission unit.";
}

leaf meter-profile {
    type meter:meter-ref;
    description
        "A reference to a Meter Profile.";
}

choice frame-to-cos-policy {
    case map {
        leaf policy {
            type enumeration {
                enum outer-tag;
                enum inner-tag;
                enum dscp;
                enum outer-mpls-tc;
            }
        }

        leaf frame-to-cos-map {
            type ftc:frame-to-cos-ref;
        }
    }
    case fixed {
        leaf cos {
            type uint8 {
                range "0..63";
            }
        }

        leaf color {
            type enumeration {
                enum green;
                enum yellow;
                enum red;
            }
        }
    }
}

leaf cos-to-frame-map {
    type ctf:cos-to-frame-ref;
}
```


Copyright© 2022 Ciena® Corporation

```
            leaf flood-containment-profile {
                type mef-fc:flood-containment-profile-ref;
            }

            leaf description {
                type string;
                description
                      "List of logical ports.";
            }

            leaf-list outer-tpid {
                 type enumeration {
                     enum tpid-8100;
                     enum tpid-88a8;
                     enum tpid-9100;
                 }
                 description
                     "A list of valid outer-vlan-tag TPIDs for the port.";
            }

            leaf-list inner-tpid {
                 type enumeration {
                     enum tpid-8100;
                     enum tpid-88a8;
                     enum tpid-9100;
                 }
                 description
                     "A list of valid inner-vlan-tag TPIDs for the port.";
            }

            leaf egress-qos {
                type enumeration {
                    enum off;
                    enum on;
                }
            }
        }
    }

    container logical-port-oper-status {

       config false;

       list logical-port-status {

          key "name";

          description
             "List of all logical ports operational data.";

          leaf name {
             type string;

             description
                 "A string that identifies the logical port.";
          }

          leaf index {
             type uint32;
          }
```


Copyright© 2022 Ciena® Corporation

```
            leaf mac-address {
               type yt:mac-address;
            }

            leaf oper-state {
               type lp-oper-state;

               description "Operational state of this logical port";
            }
         }
      }
}
```


Copyright© 2022 Ciena® Corporation

ciena-mef-mac-management.yang

```
module ciena-mef-mac-management {
   namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-mef-mac-
management";
   prefix "mef-mac-management";

   import ciena-mef-logical-port {
      prefix "mef-logical-port";
   }

   import ciena-mef-fp {
      prefix "mef-fp";
   }

   import ciena-mef-fd {
      prefix "mef-fd";
   }

   import ietf-yang-types {
      prefix "yt";
   }

   import ciena-types {
      prefix ciena;
   }

   organization
      "Ciena Corporation";

   contact
      "Web URL: http://www.ciena.com/
       E-mail:  yang@ciena.com
       Postal:  7035 Ridge Road
                Hanover, Maryland 21076
                U.S.A.
       Phone:   +1 800-921-1144
       Fax:     +1 410-694-5750";

   description
      "This YANG module defines Ciena's configuration of
      the L2 Forwarding Databases (FDB), MAC-Tables, MAC Address Management";

   revision "2017-07-28" {
      description
         "Initial version";
   }

   /*
    * typedefs
    */
   typedef fd-ref {
      type leafref {
         path "/mef-fd:fds/mef-fd:fd/mef-fd:name";
      }
     description
        "This type is used by the data models that need to reference
         configured fd";
   }
```


Copyright© 2022 Ciena® Corporation

```
   typedef fdb-ref {
      type leafref {
         path "/mef-mac-management:fdbs/mef-mac-management:fdb/mef-mac-
management:name";
       }
     description
        "This type is used by the data models that need to reference
         configured fdb";
   }

   /*
    * Type definition
    */
   identity mac-refresh-algorithm {
      description
         "Algorithm to refresh dynamic-mac entries.";
   }

   identity  sa-hit-refresh {
       base mac-refresh-algorithm;
       description
          "With this algorithm, dynamic-mac entries are refreshed based on a
MAC-SA match.";
   }

   identity sa-da-hit-refresh {
      base mac-refresh-algorithm;
      description
         "With this algorithm, dynamic-mac entries are refreshed based on a
MAC-DA or MAC-SA match.";
   }

   /*
    * Features
    */

   /*
    * Configuration data nodes
    */
   container mac-management {

      leaf mac-aging {
         type ciena:admin-state;
         default enabled;
         description
            "Configuration Control of MAC-Aging.";
      }

      leaf mac-aging-timeout {
         type uint32;
         units seconds;
         default 300;
         description
            "The number of seconds of inactivity to determine when a given
dynamic mac-entry in an
             FDB ages out and is removed from the FDB.";
      }

      leaf dynamic-mac-refresh-algorithm
      {
```


Copyright© 2022 Ciena® Corporation

```
         type identityref {
            base mac-refresh-algorithm;
         }
         description
            "Algorithm selection to refresh dynamic-mac entries.";
      }

      choice  mac-maximum-table-size {

         case size {
            leaf size {
               type uint32;
            }
            description
               "The maximum number of mac-entries allowed for a given FDB.";
         }

         case maximum {
            leaf maximum {
               type empty;
            }
            description
              "The maximum number of mac-entries is determined by underlying
forwarding-plane.";
         }
      }
   }

   container fdbs {

      list fdb {
         key "name";

         description
            "The list of configured l2 forwarding databases.";

         leaf name {
            type string;
            description
               "An administratively assigned string, which may be used to
identify the l2 forwarding database.
                Typically this may be the name of the forwarding-domain when
there is an FDB per forwarding-domain (IVL case).";
         }

         leaf description {
            type string;
            description
              "This is string used to describe the l2 Forwarding Database.";
         }

         leaf mac-learning {
            type ciena:admin-state;
            description
              "Configuration Control of MAC-Learning-Enable per-FDB. Separately
there may be configuration
                control of MAC-Learning-Enable at a more granular level (e.g.
per FD, per-Flow-Point etc).";
         }
      }
```


Copyright© 2022 Ciena® Corporation

```
   }

   container static-macs {

      list mac-entry {
         key "mac-address fdb";

         description
            "The list of configured static mac entries. The destination-
interface indirectly determines the
             fdb that the static mac-entry belongs to.";

         leaf mac-address {
            type yt:mac-address;
            description
               "The mac-address for this static-mac entry.";
         }

         leaf mac-address-mask {
            type yt:mac-address;
            description
               "An optional mac-address mask for this static-mac entry.";
         }

         leaf fdb {
            type mef-mac-management:fdb-ref;
            description
              "Reference to fdb name, based on the static-mac to be added.";
         }

         choice destination-interface {
            case flow-point {
               leaf flow-point {
                  type mef-fp:fp-ref;
               }
               description
                 "In this case flow-point defines the destination-flow-point
that the static-mac is associated to.";
            }

            case logical-port {
               leaf logical-port {
                  type mef-logical-port:logical-port-ref;
               }
               description
                 "In this case logical-port defines the destination-logical-
port that the static-mac is associated to.";
            }
         }

         leaf filter-action {
            type enumeration {
               enum allow;
               enum drop;
            }
            description
              "allow/drop destination MAC addresses which match this static
mac entry.";
            }
      }
```


Copyright© 2022 Ciena® Corporation

```
   }

   container mac-access-controls {

      list mac-access-control {
         key "logical-port";

         description
            "The list of mac-access-controls configured.";

         leaf logical-port {
            type mef-logical-port:logical-port-ref;
            description
               "The list of mac-access-controls configured.";
         }

         choice mac-learn-limit {
            case on {
               leaf maximum-dynamic-macs {
                  type uint64;
                  description
                  "The count of dynamic mac entries beyond which mac-learning
will not be allowed to occur for this scope";
               }

               leaf forward-unlearned {
                  type boolean;
               }
            }

            case off {
               leaf mac-learn-limit-off {
                  type empty;
               }
            }
         }
      }
   }

   /*
    * Oper State data nodes
    */
   container fdbs-state {
      config false;

      description
        "FDB operational data for all L2 Forwarding-Databases (MAC-Tables).";

      list fdb {
         key "name";
         description
            "The operational data for this L2 Forwarding Database.";

         leaf name {
            type string;
            description
               "A string that identifies the L2 Forwarding Database.";
         }

         leaf mac-learn-count {
```


Copyright© 2022 Ciena® Corporation

```
            type uint64;
            description
               "Current count of mac entries in the fdb";
         }

         leaf mac-max-learn-count {
            type uint64;
            description
               "Maximum count of mac entries allowed in the fdb";
         }

         list mac-entry {
            key "mac-address";
            description
               "The list of mac entries in the fdb.";

            leaf mac-address {
               type yt:mac-address;
               description
                  "The mac-address for this mac entry.";
            }

            leaf mac-address-mask {
               type yt:mac-address;
               description
                  "An optional mac-address mask for this mac entry.";
            }

            choice destination-interface {

               case flow-point {
                  leaf flow-point {
                     type mef-fp:fp-ref;
                  }
                  description
                     "In this case flow-point defines the destination-flow-
point that the mac-entry is associated to.";
                  }

               case logical-port {
                  leaf logical-port {
                     type mef-logical-port:logical-port-ref;
                  }
                  description
                  "In this case logical-port defines the destination-logical-
port that the mac-entry is associated to.";
                  }
            }

            leaf mac-learn-type {
               type enumeration {
                  enum dynamic;
                  enum static;
               }
               description
                  "Dynamic learned mac or static mac.";
            }
         }
      }
   }
```


Copyright© 2022 Ciena® Corporation

```
   container mac-access-controls-state {

      list mac-access-control {
         key "logical-port";

         description
            "The list of mac-access-controls configured.";

         leaf logical-port {
            type mef-logical-port:logical-port-ref;
         }

         leaf mac-dynamic-access-count {
            type uint64;
            description
               "The current count of dynamic mac entries for this scope";
         }

         leaf high-water-mark {
            type uint64;
            description
               "High water mark of dynamically learned MAC entries";
         }

         leaf is-threshold-reached {
            type boolean;
            description
              "Mac learn limit threshold reached or not, true means yes and
false means no";
         }
      }
   }

   /*
    * RPC.
    */
   rpc dynamic-mac-flush {
      description
         "Flush dynamic MAC-entries from FDB(s) at a given scope.";

      input {
         choice flush-scope {

            case forwarding-domain {
               leaf forwarding-domain {
                  type fd-ref;
               }
               description
                 "In this case flush all dynamic mac-entries learned against
the specified forwarding-domain.";
            }

            case flow-point {
               leaf flow-point {
                  type mef-fp:fp-ref;
               }
               description
                 "In this case flush all dynamic mac-entries learned against
the specified flow-point.";
```


Copyright© 2022 Ciena® Corporation

```
            }

            case logical-port {
               leaf logical-port {
                  type mef-logical-port:logical-port-ref;
               }
               description
                 "In this case flush all dynamic mac-entries learned against
the specified logical-port.";
            }

            case all {
               leaf all {
                  type empty;
               }
               description
                  "In this case flush all dynamic mac-entries learned on the
device as a whole.";
            }
         }
      }
      output {
         leaf status {
            type string;
            description
               "Status of MAC flush operation.";
         }
      }
   }

   rpc service-access-control-state-clear {
      description
         "Clear the service access control state data.";

      input {
         leaf logical-port {
            type mef-logical-port:logical-port-ref;
            description
               "The logical port name to clear the state data.";
         }
      }
      output {
         leaf status {
            type string;
            description
               "Status of the clear state operation.";
         }
      }
   }
}
```


Copyright© 2022 Ciena® Corporation

ciena-mef-meter-profile-deviation.yang

```
module ciena-mef-meter-profile-deviation {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn::ciena-mef-meter-
profile-dev";
    prefix "meter-dev";

    import ciena-mef-meter-profile {
        prefix meter;
    }

    revision 2015-07-16 {
        description "Initial revision";
    }


    deviation /meter:meter-profiles/meter:meter-profile/meter:description {
        deviate not-supported;
    }

   deviation /meter:meter-profiles/meter:meter-profile/meter:coupling-flag {
        deviate not-supported;
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-mef-meter-profile.yang

```
module ciena-mef-meter-profile {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn::ciena-mef-meter-
profile";
    prefix "meter";

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
                 Hanover, Maryland 21076
                 U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
        "This YANG module defines the Meter Profile configuration
         requirements.";

    revision 2015-07-16 {
        description "Initial revision";
    }

    /*
     * Typedefs
     */
    typedef meter-ref {
        type leafref {
            path "/meter:meter-profiles/meter:meter-profile/meter:name";
        }
        description
          "This type is used by data models that need to reference
           configured meter profiles.";
    }

    container meter-profiles {
        list meter-profile {
            key "name";

            leaf name {
                type string;
                description
                    "A unique identifier for the profile that is either
                     set as part of configuration or set by the system.";
            }

            leaf description {
                type string;
                description
                    "A more detailed description of the profile.";
            }

            leaf cir {
                type uint32;
                description
                    "Committed Information Rate (CIR). CIR is a Bandwidth
```


Copyright© 2022 Ciena® Corporation

```
                    Profile parameter. It defines the average rate in
                    Kbits/s of Service Frames up to which the network
                    delivers Service Frames and meets the performance
                    objectives defined by the CoS Service Attribute.";
            }
            leaf eir {
                type uint32;
                description
                  "Excess Information Rate (EIR). EIR is a Bandwidth Profile
                     parameter. It defines the average rate in Kbits/s of
                     Service Frames up to which the network may deliver
                    Service Frames but without any performance objectives.";
            }
            leaf cbs {
                type uint32;
                description
                    "Commited Burst Size (CBS). CBS is a Bandwidth Profile
                     parameter. It limits the maximum number of bytes
                    available for a burst of Service Frames sent at the UNI
                     speed to remain CIR-conformant.";
            }
            leaf ebs {
                type uint32;
                description
                    "Excess Burst Size (EBS). EBS is a Bandwidth Profile
                     parameter. It limits the maximum number of bytes
                    available for a burst of Service Frames sent at the UNI
                     speed to remain EIR-conformant.";
            }
            leaf color-aware {
                type boolean;
                description
                    "The Bandwidth Profile algorithm is said to be in color
                     aware mode when each Service Frame already has a level
                     of compliance (i.e., a color) associated with it and
                     that color is taken into account in determining the
                     level of compliance  by the Bandwidth Profile
                   algorithm. The Bandwidth Profile algorithm is said to be
                     in color blind mode when the color (if any) already
                     associated with each Service Frame is ignored by the
                     Bandwidth Profile Algorithm.";
            }
            leaf coupling-flag {
                type boolean;
                description
                    "CF is a Bandwidth Profile parameter. The Coupling Flag
                    allows the choice between two modes of operation of the
                     rate enforcement algorithm. It takes a value of 0 or 1
                     only.";
                reference
                    "MEF 10.2, Section 7.11.1, Standard Bandwidth Profile
                     and Parameters.";
            }
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-mef-pfg-profile.yang

```
module ciena-mef-pfg-profile {
   namespace "urn:ciena:params:xml:ns:yang:ciena-pn::ciena-mef-pfg-profile";
    prefix "mef-pfg";

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
         E-mail:  yang@ciena.com
         Postal:  7035 Ridge Road
                  Hanover, Maryland 21076
                  U.S.A.
         Phone:   +1 800-921-1144
         Fax:     +1 410-694-5750";

    description
        "This YANG module defines Ciena's configuration of
         the Private Forwarding Group Profile.";

    revision 2016-05-19 {
        description "Initial revision";
    }

    typedef pfg-profile-ref {
        type leafref {
            path "/mef-pfg:pfg-profiles/mef-pfg:pfg-profile/mef-pfg:name";
        }
        description
            "This type is used by the data models that need to reference
             configured Private Forwarding Group Profiles.";
    }

    typedef PfgType {
        type enumeration {
            enum pfg-groups {
                description
               "This pfg-type allows for up to 4 PFG groups with configurable
forwarding policies.
                 Interfaces (e.g. flow points) in the domain (e.g forwarding-
domain) will specify which
                  PFG-group A,B,C or D that they belong to. This is the only
PfgType which allows
                     configuration of forwarding group policies.";
            }
            enum spoke-and-mesh {
                description
                "This pfg-type supports spoke and mesh with strict forwarding
policy rules
                       spoke => can forward to mesh and can forward to spoke.
                     mesh => can forward to spoke but cannot forward to mesh.
                  Interfaces (e.g. flow points) in the domain (e.g forwarding-
domain) will specify which
                     PFG-group Spoke or Mesh that they belong to.";
            }
            enum leaf-and-root {
                description
```

```
                "This pfg-type supports leaf and root with strict forwarding
policy rules
                     leaf => can forward to root but cannot forward to leaf.
                      root => can forward to leaf and can forward to root.
                 Interfaces (e.g. flow points) in the domain (e.g forwarding-
domain) will specify which
                   PFG-group Leaf or Root that they belong to.";
            }
            enum spokemesh-and-leafroot {
                description
                    "This pfg-type supports leaf, root, spoke and mesh with
strict forwarding policy rules
                     leaf => can forward to root, can forward to mesh, can
forward to spoke, but cannot forward to leaf.
                     root => can forward to root, can forward to mesh, can
forward to spoke, can forward to leaf.
                     spoke => can forward to root, can forward to mesh, can
forward to mesh, can forward to leaf.
                     mesh => can forward to root, can forward to spoke, can
forward to leaf, but cannot forward to mesh.
                 Interfaces (e.g. flow points) in the domain (e.g forwarding-
domain) will specify which
                  PFG-group Spoke, Mesh, Leaf or Root that they belong to.";
            }
        }
    }

    typedef group-policy {
        type bits {
            bit forward-to-a {
                description
                    "can forward to PFG-group A";
                }
            bit forward-to-b {
                description
                    "can forward to PFG-group B";
                }
            bit forward-to-c {
                description
                    "can forward to PFG-group C";
                }
            bit forward-to-d {
                description
                    "can forward to PFG-group D";
                }
        }
        description
            "When PfgType is pfg-groups the forwrding policies between the
groups can be configured.
             Each PFG Group A,B,C or D can define the set of PFG-groups that
it can forward to (including itself)
             If a forward-to-x bit is not set for a PFG-group, the PFG-group
cannot forward to any interface (e.g. flow-point)
              in that PFG-group. Each interface defines which PFG-group it
belongs to.";
    }

    container pfg-profiles {

        list pfg-profile {
```


Copyright© 2022 Ciena® Corporation

```
            key "name";

            leaf name {
                type string;
                description
                    "An administratively assigned string, which may be used
                     to identify the profile.";
            }

            leaf description {
                type string;
                description
                "This is a user-defined string used to describe the profile.";
            }

            leaf pfg-type {
                type PfgType;
                description
                    "The type of PFG-profile whch has configurable policies
or strict forwarding policies.";
            }

            leaf pfg-group-count {
                type uint32;
                description
                   "When the pfg-type is pfg-groups, this defines the number
of PFG groups that are valid in this profile.
                    2 => PFG-Groups A and B
                    3 => PFG-Groups A, B and C
                    4 => PFG-Groups A,B,C and D.";
            }

            leaf group-A-policy {
                type group-policy;
                description
                    "The set of forwarding rules for Group A.";
            }

            leaf group-B-policy {
                type group-policy;
                description
                    "The set of forwarding rules for Group B.";
            }

            leaf group-C-policy {
                type group-policy;
                description
                    "The set of forwarding rules for Group C.";
            }

            leaf group-D-policy {
                type group-policy;
                description
                    "The set of forwarding rules for Group D.";
            }
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-mef-qos-flow-deviation.yang

```
module ciena-mef-qos-flow-deviation {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn::ciena-mef-qos-flow-
dev";
    prefix "flow-dev";

    import ciena-mef-qos-flow {
        prefix flow;
    }

    revision 2015-07-14 {
        description "Initial revision";
    }

    deviation /flow:qos-flows/flow:qos-flow/flow:description {
        deviate not-supported;
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-mef-qos-flow.yang

```
module ciena-mef-qos-flow {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn::ciena-mef-qos-flow";
    prefix "flow";

    import ciena-mef-classifier {
        prefix "classifier";
    }

    import ciena-mef-meter-profile {
        prefix "meter";
    }

    import ciena-mef-frame-to-cos-map {
        prefix "ftc";
    }

    import ciena-mef-cos-to-frame-map {
        prefix "ctf";
    }

    import ciena-mef-logical-port {
        prefix "mef-logical-port";
    }

    import ciena-mef-fp {
        prefix "mef-fp";
    }

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
                 Hanover, Maryland 21076
                 U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
        "This YANG module defines the QoS Flow.";


    revision 2016-02-03 {
        description "Initial revision";
    }

/*
 * typedefs
 */
    typedef qos-flow-ref {
        type leafref {
            path "/flow:qos-flows/flow:qos-flow/flow:name";
        }
        description
            "This type is used by the data models that needs to reference
             configured QoS flows.";
```

```
}

container qos-flows {

    list qos-flow {
        key "name";

        leaf name {
            type string;

            description
              "A unique string that is either system assigned or assigned
                 by the user but does not change over its life.";
        }

        leaf description {
            type string;
            description
                "A more detailed description that an operator can use
                 to describe the flow.";
        }

        leaf classifier {
            type classifier:classifier-ref;
            description
                "A reference to a list of classifier entries.";
        }

        leaf classifier-precedence {
            type uint32;
            description
                "A precedence value for the qos flow. Lower values take
                 precedence over higher values";
        }

        leaf meter-profile {
            type meter:meter-ref;
            description
                "A reference to a Meter Profile.";
        }

        choice parent-interface {
            case port {
                leaf-list parent-port {
                    type mef-logical-port:logical-port-ref;
                }
            }

            case fp {
                leaf-list parent-fp {
                    type mef-fp:fp-ref;
                }
            }
        }

        leaf cos-to-frame-map {
            type ctf:cos-to-frame-ref;
        }

        choice frame-to-cos-map {
```


Copyright© 2022 Ciena® Corporation

```
            case map {
                leaf map-policy {
                    type enumeration {
                        enum outer-vlan-tag {
                            description
                                "Use outer VLAN to map to CoS
                                 and color.";
                        }
                        enum inner-vlan-tag {
                            description
                                "Use inner VLAN to map to CoS
                                 and color.";
                        }
                        enum dscp {
                            description
                                "Use DSCP to map to CoS and
                                 color.";
                        }
                        enum  outer-mpls-tc {
                            description
                                "Use MPLS TC bits to map to CoS
                                 and color.";
                        }
                    }
                }

                leaf map-entry {
                    type ftc:frame-to-cos-ref;
                }
            }

            case fixed {
                leaf fixed-cos {
                    type uint16 {
                        range "0..63";
                    }
                }

                leaf fixed-color {
                    type enumeration {
                        enum green;
                        enum yellow;
                        enum red;
                    }
                }
            }
        }

        choice ingress-l2-transform {
            case stamp {
                choice dei-policy {
                    case map {
                        leaf dei-map {
                            type empty;
                        }
                        description
                            "Map from the incoming frame.";
                    }
                    case fixed {
                        leaf dei-value {
```


Copyright© 2022 Ciena® Corporation

```
                            type boolean;
                        }
                        description
                            "Fixed DEI value.";
                    }
                }

                choice pcp-policy {
                    case map {
                        leaf pcp-map {
                            type empty;
                        }
                        description
                            "Map from the incoming frame.";
                    }
                    case fixed {
                        leaf pcp-value {
                            type uint8 {
                                range "0..7";
                            }
                        }
                        description
                            "Fixed PCP value.";
                    }
                }
            }
        }

        choice ingress-l3-transform {
            case stamp {
                choice dscp-policy {
                    case map {
                        leaf dscp-map {
                            type ftc:frame-to-cos-ref;
                        }
                        description
                            "Reference to a Frame to CoS Map.";
                    }

                    case fixed {
                        leaf dscp-value {
                            type uint8 {
                                range "0..63";
                            }
                        }
                        description
                            "Fixed DSCP value.";
                    }
                }
            }
        }
    }
}

/*
 * Oper Data
 */
container qos-flow-state {

    config false;
```


Copyright© 2022 Ciena® Corporation

```
        description
            "Global Qos Flow operational data.";

        list qos-flow {

            key "name";

            description
                "The operational data for this Qos Flow.";

            leaf name {
                type string;

                description
                    "A string that identifies the Qos Flow.";
            }
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-netch-yuma-api.yang

```
module ciena-netch-yuma-api {

    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-netch-yuma-api";

    prefix "netchyumaapi";

    import yumaworks-ycontrol { prefix yctl; }

    organization "Ciena";

    description
     "This yang model defines Ciena defined netch-yuma-api service over
Ycontrol";

    revision 2019-04-05 {
        description
          "Fix broken augment statement for adding inside a choice";
    }

    revision 2016-05-02 {
        description
          "Add load-bundle-request";
    }

    revision 2015-07-06 {
        description
          "Initial version";
    }

   augment "/yctl:ycontrol/yctl:message-payload/yctl:payload/yctl:payload" {
        container netch-yuma-api {
            choice message-type {
                mandatory true;

                leaf register-request {
                    type empty;
                    description
                     "Message type: subsys-request;
                      Purpose: register the NETCH-YUMA-API subsystem
                      Expected Response Message: ok or error";
                }

                container edit-request {
                    description
                     "Message type: subsys-request;
                      Purpose: Ask the main server to accept an edit request
                      Expected Response Message: ok or error";

                    leaf log-level {
                        type string;
                        description
                         "Message type: subsys-request;
                          Purpose: Set the server log level
                          Expected Response Message: ok or error";
                    }

                    leaf allowed-user {
                        type string;
```


Copyright© 2022 Ciena® Corporation

```
                        description
                         "Message type: subsys-request;
                          Purpose: Add user to yumapro server user database
                          Expected Response Message: ok or error";
                    }

                    leaf load-bundle {
                        type string;
                        description
                         "Message type: subsys-request;
                          Purpose: Ask the yumapro server to load a bundle
                          Expected Response Message: ok or error";
                    }
                }
            }
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-ntp.yang

```
module ciena-ntp {
  namespace "http://www.ciena.com/ns/yang/ciena-ntp";
  prefix "ciena-ntp";

  import ietf-inet-types {
    prefix "inet";
  }

  import ietf-system {
    prefix sys;
  }

  import ciena-types {
    prefix ciena;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module defines the nodes for the NTP client for the
     NETCONF protocol.

     Copyright (c) 2016 Ciena Corporation.  All rights reserved.

     All information contained herein is, and remains the property of
     Ciena Corporation. Dissemination of this information or
     reproduction of this material is strictly forbidden unless prior
     written permission is obtained from Ciena Corporation.";

  revision "2018-01-08" {
    description
      "Initial version";
    reference
      "RFC 6020: YANG - A Data Modeling Language for the Network
       Configuration Protocol (NETCONF).  No specific reference;
       standard not available.";
  }

  /* Items of Note:
   *
   *
   * NEW :
   * 1. Added typedef 'float' with type decimal64
   *      - not sure what to call this type. since yang modeling does not
   *        have 'float' or double types.
   */

  /*
```


Copyright© 2022 Ciena® Corporation

```
 * Typedefs
 *
 */

typedef float {
  type decimal64 {
    fraction-digits 9; // range is 1 to 18. needed to use 9 to
                       // represent parts-per-billion.
  }
  description
    "";
}

/*
 * Configuration data nodes
 */
augment /sys:system {
  description
    "System group configuration augmentations.";
  container ntp {
    description
      "Configuration parameters for NTP.";

    leaf admin-state {
      type ciena:admin-state;
      description
        "Setting this node to 'enabled' enables the NTP client on
         Ciena devices.";
    }

    leaf mode {
      type enumeration {
        enum polling {
          value 1;
          description
            "";
        }
        enum broadcast {
          value 2;
          description
            "";
        }
        enum multicast {
          value 3;
          description
            "";
        }
      }
      description
        "Setting this node to 'broadcast' causes the client to not
         poll the NTP server at all. Instead, it waits for configured
         NTP servers to broadcast their current time. When the client
         receives the first message, it retrieves a reliable time from
         that server. If it is set to polling then device polls the
         NTP server.";
    }

    leaf polling-interval-min {
      type int32 {
        range "16..4096";
```


Copyright© 2022 Ciena® Corporation

```
    }
    units "seconds";
    description
      "The polling frequency (interval in seconds) at which NTP
       server is polled if the NTP client is configured to be in
       polling mode and is not currently synchronized to an NTP
       server.  The value of this node must be a power of 2.  The
       NTP client accepts the values 2^4 (16 seconds) to 2^12 (4096
       seconds).  The change in the NTP mode is only effective if
       the NTP client is enabled.";
  }

  leaf polling-interval-max {
    type int32 {
      range "16..4096";
    }
    units "seconds";
    description
      "The polling frequency (interval in seconds) at which NTP
       server is polled if the NTP client is configured to be in
       polling mode and is currently synchronized to an NTP server.
       The value of this node must be a power of 2.  The NTP client
       accepts the following values 2^4 (16 seconds) to 2^12 (4096
       seconds).  The change in the NTP mode is only effective if
       the NTP client is enabled.";
  }

  leaf sync-notification {
    type enumeration {
      enum on {
        value 1;
        description
          "";
      }
      enum off {
        value 2;
        description
          "";
      }
    }
    description
      "Set to on to turn on sync change notifications.";
  }

  container authentication {
    description
      "Configuration of authentication.";

    leaf auth-admin-state {
      type ciena:admin-state;
      description
        "Set to enabled to use authentication for NTP client.";
    }

    list auth-entry {

      key "auth-key-id";
      description
        "An entry in the auth-entry list.";
```


Copyright© 2022 Ciena® Corporation

```
        leaf auth-key-id {
          type uint32 {
            range "1..65534";
          }
          description
            "This node specifies the key ID.
             It is also used as key in the table.";
        }

        leaf auth-key {
          type string {
            length "0..40";
          }
          description
            "Used to specify the key to use with this key identifier.
             This node may be populated with the unencrypted string
             while creating this entry in this list.  When read, this
             node returns an empty string, for security reasons.
             Refer to the auth-key-type leaf for the Authentication
             Type used.";
        }

        leaf auth-key-enc {
          type string {
            length "0..259";
          }
          description
            "Used to specify the encrypted key to use with this key
             identifier.  This node may be populated with the
             pre-encrypted string while creating a new entry in this
             list.  When read, this node returns the encrypted string
             for the entry in the list.  Refer to the auth-key-type
             leaf for the Authentication Type used.";
        }

        leaf auth-key-type {
          type enumeration {
            enum md5 {
              value 1;
              description
                "";
            }
            enum sha1 {
              value 2;
              description
                "";
            }
            enum none {
              value 3;
              description
                "";
            }
          }
          description
            "Used to specify the type of authentication key used.";
        }

      } // end auth-entry

    } // end authentication container
```


Copyright© 2022 Ciena® Corporation

```
container associations {
  description
    "Configuration of peering and client servers.";

  container peer-server {
    description
      "Peer associations.";

    leaf peering-admin-state {
      type ciena:admin-state;
      description
        "Controls whether NTP client peers may retrieve time from
         the device.  By default, the device will allow time
         synchronization with NTP client peers while the device
         itself has synchronized time from an NTP server.";
    }

  }

  // The client connects to a remote NTP server to obtain the
  // time.  NTP Servers set by user, and not by DHCP.
  // NTP Servers set by DHCP are available by the oper. state.
  container remote-ntp-server {
    description
      "Remote NTP Servers.";

    list server-entry {
      key "address";
      description
        "";

      leaf admin-state {
        type ciena:admin-state;
        description
          "This node is used to specify the user administrative
           state of the server.";
      }

      leaf address {
        type inet:host;
        description
          "The IP address or host name of the NTP server.  If this
           node contains a hostname; then the operational state
           node resolved-address will contain the resolved
           address.";
      }

      leaf auth-key-id {
        type uint32 {
          range "0..4294967295";
        }
        description
          "This node is used to specify the auth key ID that the
           server uses.  A 0 key ID implies no auth key";
      }

    }

  } // end remote-ntp-server
```


Copyright© 2022 Ciena® Corporation

```
        container multicast-server {
          description
            "Configuration for Multicast.";
          leaf-list ipv6-address {
            type inet:ipv6-address;
            description
              "The address of the multicast server.";
          }
        }

      } // end associations container

    } // end ntp container
  }
  /*
   * Operational state data nodes
   */
  container ntp-state {
    config false;
    description
      "";

    leaf delay {
      type float;
      description
        "When NTP client is synchronized, this indicates the round trip
         time in micro-seconds taken by the reply packet to return in
         response, to a query sent by the NTP server.";
    }

    leaf offset {
      type float;
      description
        "When NTP client is synchronized, this contains the difference
         between the reference clock and the system clock in
         micro-seconds.";
    }

    leaf jitter {
      type float;
      description
        "When NTP is synchronized, it indicates the magnitude of jitter
         between several time queries in micro-seconds.";
    }

    leaf drift {
      type float;
      description
        "The Drift in micro-seconds of the local clock.  This value is
         only valid when NTP client is enabled and has a valid peer.";
    }

    leaf fast-offset {
      type float;
      description
        "The Fastest Offset seen in micro-seconds of the local clock.
         This value is only valid when NTP client is enabled and has
         a valid peer.";
    }
```


Copyright© 2022 Ciena® Corporation

```
leaf slow-offset {
  type float;
  description
    "The Slowest Offset seen in micro-seconds of the local clock.
     This value is only valid when NTP is enabled and has a valid
     peer.";
}

leaf driftPPB {
  type float;
  units "parts per billion";
  description
    "The frequency error in the local clock as expressed in parts
     per billion.  This value is only valid when NTP client is
     enabled and has a valid peer.";
}

leaf synchronized {
  type boolean;
  description
    "This node indicates the synchronized state of the NTP client.";
}

leaf dhcp-ntp-option-state {
  type enumeration {
    enum enable {
      value 1;
      description
        "";
    }
    enum disable {
      value 2;
      description
        "";
    }
  }
  description
    "This node is used to get the DHCP NTP Option State";
}

leaf scope {
  type enumeration {
    enum user {
      value 1;
      description
        "";
    }
    enum dhcp {
      value 2;
      description
        "";
    }
    enum both {
      value 3;
      description
        "";
    }
  }
  description
```


Copyright© 2022 Ciena® Corporation

```
        "This node is used to specify the from where the configuration
         was set.  The scope indicates scope if the entry is configured
         by DHCP, user, or both.";
    }

    container associations {
      description
        "Operational state of user set remote NTP servers.";

      container remote-ntp-server {
        description
          "Contains NTP Servers.";
        list server-entry {
          key "address";
          description
            "";

          leaf admin-state {
            type ciena:admin-state;
            description
              "This node is used to specify the user administrative
               state of the server.";
          }

          leaf address {
            type inet:host;
            description
              "The IP address or host name of the NTP server.  If this
               node contains a hostname; then the operational state node
               resolved-address will contain the resolved address.";
          }

          leaf resolved-address {
            type inet:ipv4-address;
            description
              "This will be the resolved ip address of remote-ntp-server
               address leaf if a hostname is used.";
          }

          leaf auth-key-id {
            type uint32 {
              range "0..4294967295";
            }
            description
              "This node is used to specify the auth key ID that the
               server uses.  A 0 key ID implies no auth key";
          }

          leaf scope {
            type enumeration {
              enum user {
                value 1;
                description
                  "";
              }
              enum dhcp {
                value 2;
                description
                  "";
              }
```


Copyright© 2022 Ciena® Corporation

```
      enum both {
        value 3;
        description
          "";
      }
    }
    description
      "This node is used to specify the from where the
       configuration was set.  The scope indicates scope if the
       entry is configured by DHCP, user, or both.";
  }

  leaf oper-state {
    type enumeration {
      enum enable {
        value 1;
        description
          "";
      }
      enum disable {
        value 2;
        description
          "";
      }
    }
    description
      "This node gets the operational state of the remote NTP
       server";
  }

  leaf server-state {
    type enumeration {
      enum reach {
        value 1;
        description
          "";
      }
      enum unReach {
        value 2;
        description
          "";
      }
      enum error {
        value 3;
        description
          "";
      }
    }
    config false;
    description
      "This node is used to specify the reach state of the
       remote server.";
  }

  leaf server-condition {
    type enumeration {
      enum reject {
        value 1;
        description
          "";
```


Copyright© 2022 Ciena® Corporation

```
        }
        enum insane {
          value 2;
          description
            "";
        }
        enum correct {
          value 3;
          description
            "";
        }
        enum standby {
          value 4;
          description
            "";
        }
        enum candidate {
          value 5;
          description
            "";
        }
        enum selected {
          value 6;
          description
            "";
        }
        enum syspeer {
          value 7;
          description
            "";
        }
        enum ppspeer {
          value 8;
          description
            "";
        }
        enum reaching {
          value 255;
          description
            "";
        }
        enum error {
          value 256;
          description
            "";
        }
      }
      description
        "This node reflects the server condition of the remote
         server.";
    }

    leaf auth-state {
      type enumeration {
        enum ok {
          value 1;
          description
            "";
        }
        enum bad {
```


Copyright© 2022 Ciena® Corporation

```
              value 2;
              description
                "";
            }
            enum none {
              value 3;
              description
                "";
            }
            enum error {
              value 4;
              description
                "";
            }
          }

          description
            "This node indicates the auth state of the remote NTP
             server.";
        }

        leaf offset {
          type int32;
          description
            "When remote NTP server is synchronized, this contains the
             difference between the reference clock and the system
             clock in micro-seconds.";
        }

      } // end dhcp set remote ntp servers list

    } // end dhcp-set-remote-ntp-servers

  } // end associations container

  container broadcast-server {
    description
      "Operational State for the Broadcast Server.";

    leaf address {
      type inet:ipv4-address;
      description
        "Specifies the NTP broadcast address.";
    }

    leaf key-id {
      type uint32 {
        range "0..65534";
      }
      description
        "This node is used to specify the cleartext key ID that the
         server uses.  A 0 key ID implies no key.";
    }

    leaf admin-state {
      type ciena:admin-state;

      description
        "This node is used to specify the user administrative state of
         the broadcast server.";
```


Copyright© 2022 Ciena® Corporation

```
}

leaf reach-state {
  type enumeration {
    enum reach {
      value 1;
      description
        "";
    }
    enum unReach {
      value 2;
      description
        "";
    }
    enum error {
      value 3;
      description
        "";
    }
  }
  description
    "This node is used to specify the reach state of the broadcast
     server.";
}

leaf status {
  type enumeration {
    enum reject {
      value 1;
      description
        "";
    }
    enum insane {
      value 2;
      description
        "";
    }
    enum correct {
      value 3;
      description
        "";
    }
    enum standby {
      value 4;
      description
        "";
    }
    enum candidate {
      value 5;
      description
        "";
    }
    enum selected {
      value 6;
      description
        "";
    }
    enum syspeer {
      value 7;
      description
```


Copyright© 2022 Ciena® Corporation

```
            "";
        }
        enum ppspeer {
          value 8;
          description
            "";
        }
        enum reaching {
          value 255;
          description
            "";
        }
        enum error {
          value 256;
          description
            "";
        }
      }
      description
        "This node reflects the current connection status of the
         broadcast server.";
    }

    leaf oper-state {
      type enumeration {
        enum enabled {
          value 1;
          description
            "";
        }
        enum disabled {
          value 2;
          description
            "";
        }
        enum notConfigured
        {
          value 3;
          description
            "";
        }
      }
      description
        "This node is used to specify the operational state of the
         broadcast server";
    }

  } // end broadcast container

  /*
   * If there is no system support for getting NTP statistics;
   * a wrapper function will need to be created around 'ntpdc iostats'
   * using iostats for sent, received, dropped, and send failed
   * packets, to get the data for this container.
   * https://www.eecis.udel.edu/~mills/ntp/html/ntpdc.html
   * - ref for control message commands, like iostats.
   *
   */

  container statistics {
```


Copyright© 2022 Ciena® Corporation

```
      description
        "Packet statistics of NTP.";

      leaf packets-sent {
        type uint32;
        description
          "Indicates the total number of packets sent.";
      }

      leaf sent-fail {
        type uint32;
        description
          "Indicates the number of times packet sending failed.";
      }

      leaf received {
        type uint32;
        description
          "Indicates the total number of packets received.";
      }

      leaf dropped {
        type uint32;
        description
          "Indicates the number of packets dropped.";
      }

    }

  } // end ntp-state container

  // sync-change-notification [on/off]
  notification sync-status-change-notification {
    description
      "This notification is sent whenever there is a change in the
       synchronization state for the NTP client. The sync-notification
       leaf should be enabled to receive this notification.";

    container sync-status-change-notification {
      description
        "";
      leaf sync-state {
        type enumeration {
          enum synchronized {
            value 1;
            description
              "";
          }
          enum not-synchronized {
            value 2;
            description
              "";
          }
        }
        description
          "This node indicates the synchronized state of the NTP
           client.";
      }
    }
```


Copyright© 2022 Ciena® Corporation

```
  } // end sync-change-notification

  /* RPC Operations */
  rpc drift-clear {
    description
      "An RPC operation to clear the drift, stats, etc. and should
       should return the status of the operation.";
    input {
      leaf request {
        type string;
        description
          "";
      }
    }
    output {
      leaf status {
        type string;
        description
          "";
      }
    }
  }

} // end module ciena-refactor-ntp
```


Copyright© 2022 Ciena® Corporation

ciena-ocsp.yang

```
module ciena-ocsp {
  namespace "http://www.ciena.com/ns/yang/ciena-ocsp";

  prefix "ciena-ocsp";

  import ciena-types {
    prefix ciena;
  }

  import ietf-inet-types {
    prefix inet;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This module contains the YANG definition of OCSP.

     Copyright (c) 2017 Ciena Corporation.  All rights reserved.

     All information contained herein is, and remains
     the property of Ciena Corporation. Dissemination of this
     information or reproduction of this material is strictly
     forbidden unless prior written permission is obtained from
     Ciena Corporation.";

  revision "2017-08-16" {
    description
      "Initial version";
    reference
      "";
  }

  grouping ocsp {
    description
      "Holds the leaf nodes used for configuration of
       OCSP.";
    leaf admin-state {
      type ciena:admin-state;
      description
        "This leaf specifies the administrative
         state of OCSP.";
    }
    leaf default-responder {
      type inet:uri;
      description
        "This leaf specifies the OCSP default-responder";
    }
```

```
    leaf nonce {
      type enumeration {
        enum on {
          description
            "OCSP nonce state on";
        }
        enum off {
          description
            "OCSP nonce state off";
        }
      }
      description
        "This leaf specifies the OCSP nonce state";
    }
    leaf responder-preference {
      type enumeration {
        enum aia {
          description
            "Responder preference aia.";
        }
        enum default-responder {
          description
            "Responder preference default-responder";
        }
      }
      description
        "This leaf specifies the OCSP responder-preference";
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-perfmon.yang

```
module ciena-perfmon {
  namespace "http://www.ciena.com/ns/yang/ciena-perfmon";
  prefix "ciena-perfmon";

  import ietf-yang-types {
    prefix yang;
  }

  import ietf-inet-types {
    prefix inet;
  }

  import ciena-types {
    prefix ciena;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module defines Ciena's configuration of
     Performance Manager

     Copyright (c) 2016 Ciena Corporation.  All rights reserved.

     All information contained herein is, and remains
     the property of Ciena Corporation. Dissemination of this
     information or reproduction of this material is strictly
     forbidden unless prior written permission is obtained from
     Ciena Corporation.";

  revision "2020-01-30" {
    description
      "Added new PM profile TdmE3PortStats.";
    reference
      "";
  }

revision "2019-11-14" {
    description
      "Added login-id to sftp/ftp server type only.";
    reference
      "";
  }

  revision "2017-11-01" {
    description
      "Added port to the server-config grouping.";
    reference
```

```
    "";
}

revision "2017-08-07" {
  description
    "Removed dependency on ciena-xftp.";
  reference
    "";
}

revision "2016-10-27" {
  description
    "Initial version";
  reference
    "";
}

/*
 * Features
 */

feature perfmon-storage {
  description
    "This feature enables the use of changing the default perfmon
     storage types which is exclusive to certain platforms.";
}

/*
 * Types
 */

typedef pm-duration {
  type enumeration {
    enum 1m {
      description
        "1 minute.";
    }

    enum 5m {
      description
        "5 minutes.";
    }
    enum 10m {
      description
        "10 minutes.";
    }
    enum 15m {
      description
        "15 minutes.";
    }
    enum 30m {
      description
        "30 minutes.";
    }
    enum 60m {
      description
        "60 minutes.";
    }
    enum 3h {
      description
```


Copyright© 2022 Ciena® Corporation

```
          "3 hours.";
      }
      enum 12h {
        description
          "12 hours.";
      }
      enum 24h {
        description
          "24 hours.";
      }
    }
    description
      "PM duration times.";
  }

  typedef pm-bulkstats-instance-statistics {
    type enumeration {
      enum cpu {
        description
          "CPU load statistics.";
      }
      enum egress-port-queue-group {
        description
          "Egress port queue group statistics.";
      }
      enum fans {
        description
          "Fan RPM statistics.";
      }
      enum memory {
        description
          "Memory usage statistics.";
      }
      enum mpls-lsp {
        description
          "MPLS LSP statistics.";
      }
      enum mpls-vc {
        description
          "MPLS VC statistics.";
      }
      enum port {
        description
          "Port statistics.";
      }
      enum power {
        description
          "Power usage statistics.";
      }
      enum temperature {
        description
          "Temperature statistics.";
      }
      enum qos-flow {
        description
          "QoS flow statistics.";
      }
      enum virtual-circuit {
        description
          "Virtual circuit statistics.";
```


Copyright© 2022 Ciena® Corporation

```
      }
    }
    description
      "Bulkstats collection statistics.";
  }

  typedef pm-interface-type {
    type enumeration {
      enum tdm-port {
        description
          "TDM port interface type.";
      }
      enum tdm-vc {
        description
          "TDM VC interface type.";
      }
      enum access-flow {
        description
          "Access flow interface type.";
      }
      enum queue-group-profile {
        description
          "Queue group profile interface type.";
      }
      enum mpls-transit {
        description
          "MPLS transit interface type.";
      }
      enum ptp {
        description
          "PTP interface type.";
      }
      enum mpls-vc {
        description
          "MPLS VC interface type.";
      }
      enum qos-flow {
        description
          "QoS flow interface type.";
      }
      enum odu {
        description
          "ODU interface type.";
      }
      enum sub-port {
        description
          "Sub-port interface type.";
      }
      enum otu {
        description
          "OTU interface type.";
      }
      enum tp-static-transit-corout {
        description
          "TP static transit corout interface type.";
      }
      enum port {
        description
          "Port interface type.";
      }
```


Copyright© 2022 Ciena® Corporation

```
      enum tp-static-transit-unidir {
        description
          "TP static transit unidir interface type.";
      }
      enum port-queue-group {
        description
          "Port queue group interface type.";
      }
    }
    description
      "PM interface types.";
  }

  typedef pm-alignment-type {
    type enumeration {
      enum local-clock {
        description
          "Local clock based bin time alignment.";
      }
      enum start-time {
        description
          "Start time based bin time alignment.";
      }
    }
    description
      "Bin time alignment types.";
  }

  typedef pm-instance-type {
    type enumeration {
      enum on-demand {
        description
          "On demand PM instance type.";
      }
      enum proactive {
        description
          "Proactive PM instance type.";
      }
    }
    description
      "Type of PM instance.";
  }

  typedef pm-profile-type {
    type enumeration {
      enum tdm-ds3-port-stats {
        description
          "TDM DS3 port stats PM profile type.";
      }
      enum tdm-e3-port-stats {
        description
          "TDM E3 port stats PM profile type.";
      }
      enum tdm-ocn-port-stats {
        description
          "TDM DS3 port stats PM profile type.";
      }
      enum tdm-t1-port-stats {
        description
          "TDM DS3 port stats PM profile type.";
```


Copyright© 2022 Ciena® Corporation

```
      }
      enum queue-group {
        description
          "Queue group stats PM profile type.";
      }
      enum extended-tx-rx {
        description
          "Extended Tx Rx PM profile type.";
      }
    }
    description
      "PM profile types.";
  }

  typedef pm-storage-type {
    type enumeration {
      enum on-board {
        description
          "On board storage type.";
      }
      enum usb-flash {
        description
          "USB flash storage type.";
      }
    }
    description
      "PM storage types.";
  }

  typedef server-types {
    type enumeration {
      enum none {
        description
          "The none server type.";
      }
      enum tftp-server {
        description
          "The system's TFTP server.";
      }
      enum ftp-server {
        description
          "The system's FTP server.";
      }
      enum sftp-server {
        description
          "The system's SFTP server.";
      }
      enum xftp-server {
        description
          "The system's default file transfer server.";
      }
      enum http-server {
        description
          "The http server type.";
      }
    }
    description
      "Type used to define the file transfer servers.";
  }
```


Copyright© 2022 Ciena® Corporation

```
  /*
   * groupings
   */

  grouping server-config {
    leaf type {
      type server-types;
      description
        "The server type for file transfer.";
    }

    leaf address {
      type string {
        length 0..253;
      }
      description
        "The server IP address or hostname to be used for the selected
         server type.";
    }

    leaf port {
      type inet:port-number;
      description
        "The server port number to be used.";
    }

    leaf login-id {
      when "../type != 'tftp-server'" {
         description
          "Login ID does not apply to TFTP.";
      }
      type string {
        length 0..32;
      }
      description
        "Login ID for the specified server.";
    }

    uses ciena:security-password;

    description
      "XFTP server configuration options.";
  }

  /*
   * Configuration data nodes
   */
  container pm {
    description
      "Performance Monitoring related data.";

    // Configuration
    container configuration {
      description
        "Performance Monitoring configuration state data.";

      leaf admin-state {
        type ciena:admin-state;
        description
          "This leaf specifies the global administrative
```


Copyright© 2022 Ciena® Corporation

```
      state of performance monitoring.";
  }

  // PM instance
  container pm-instance {
    description
      "PM instance configuration data.";

    leaf persistence-interval {
      type uint32 {
        range "60..86400";
      }
      units seconds;
      description
        "The set interval in seconds for saving history bins.";
    }

    leaf persistence-storage {
      if-feature perfmon-storage;
      type pm-storage-type;
      description
        "Location for pm persistent storage.";
    }

    // Auto-gen instances
    list auto-instance {
      key instance-name;
      description
        "Auto created PM instance list.";

      leaf instance-name {
        type string {
          length "1..31";
        }
        description
          "PM instance name.";
      }

      leaf admin-state {
        type ciena:admin-state;
        description
          "PM instance admin state.";
      }

    }

    // User created instances
    list user-instance {
      key instance-name;
      description
        "User created PM instance list.";

      leaf instance-name {
        type string {
          length "1..31";
        }
        description
          "PM instance name.";
      }
```


Copyright© 2022 Ciena® Corporation

```
leaf interface-type {
  type pm-interface-type;
  description
    "Interface type of the PM instance.";
}

// Interface indexes
choice interface {
  mandatory true;

  case single-index {
    leaf single-index1 {
      type uint32;
      description
        "Interface with a single index.";
    }
  }

  case double-index {
    leaf double-index1 {
      type uint32;
      description
        "First index of the double index.";
    }
    leaf double-index2 {
      type uint32;
      description
        "Second index of the double index.";
    }
  }

  case triple-index {
    leaf triple-index1 {
      type uint32;
      description
        "First index of the triple index.";
    }
    leaf triple-index2 {
      type uint32;
      description
        "Second index of the triple index.";
    }
    leaf triple-index3 {
      type uint32;
      description
        "Third index of the triple index.";
    }
  }

  description
    "Interface name of the PM instance.  Some interfaces
     require one or more indexes for correlation.";
}

leaf profile-type {
  type pm-profile-type;
  mandatory true;
  description
    "PM profile type.";
}
```


Copyright© 2022 Ciena® Corporation

```
      leaf admin-state {
        type ciena:admin-state;
        description
          "PM instance admin state.";
      }

      leaf alignment {
        type pm-alignment-type;
        description
          "PM bin time alignment.";
      }

      leaf bin-count {
        type uint16 {
          range "0..96";
        }
        description
          "Number of history bins.";
      }

      leaf bin-duration {
        type pm-duration;
        description
          "Duration in ";
      }

      leaf start-date-and-time {
        type yang:date-and-time;
        description
          "Date and time to start collecting stats.";
      }

      leaf end-date-and-time {
        type yang:date-and-time;
        description
          "Date and time to end collecting stats.";
      }

      leaf instance-type {
        type pm-instance-type;
        description
          "PM instance type.";
      }
    }
  }
  // Bulkstats
  container bulkstats {
    description
      "Performance Monitor bulkstats management.";

    leaf admin-state {
      type ciena:admin-state;
      description
        "This leaf specifies the administrative state
         of bulkstats.";
    }

    // Collector
    container collector {
```


Copyright© 2022 Ciena® Corporation

```
    description
      "Bulkstats collector attributes.";

    // Bulkstats collector
    list bulkstats-collector {
      key name;
      description
        "Bulkstats collection list.";

      leaf name {
        type string {
          length "1..31";
        }
        description
          "Bulkstats collector name.";
      }

      leaf admin-state {
        type ciena:admin-state;
        description
          "Administrative state of a bulkstats collector.";
      }

      leaf instance-id {
        type uint32;
        description
          "Bulkstats collector instance ID.";
      }

      leaf push-interval {
        type uint32;
        description
          "This leaf specifies the push interval
           in terms of seconds.";
      }

      leaf-list bulkstats-instance {
        type string {
          length "1..31";
        }
        description
          "Bulkstats collector attached bulkstats instances.";
      }

      uses server-config;

      leaf remote-file-prefix {
        type string {
          length "0..127";
        }
        description
          "Remote filename prefix.";
      }

    } // Bulkstats collector
  } // Collector

  // Instance
  container instance {
    description
```


Copyright© 2022 Ciena® Corporation

```
            "Bulkstats instance attributes.";

          // Bulkstats instance
          list bulkstats-instance {
            key name;
            description
              "Bulkstats instance list.";

            leaf name {
              type string {
                length "1..31";
              }
              description
                "Bulkstats instance name.";
            }

            leaf admin-state {
              type ciena:admin-state;
              description
                "The administrative state of a bulkstats instance.";
            }

            leaf-list statistics {
              type pm-bulkstats-instance-statistics;
              description
                "Types of statistics saved to this bulkstats
                 instance";
            }

            leaf interval {
              type pm-duration;

              description
                "The bulkstats collection interval.";
            }
            leaf duration {
              type uint32;
              description
                "The bulkstats collection duration in seconds";
            }
            leaf instance-id {
              type uint32;
              description
                "The bulkstats instance logical ID.";
            }
          }

        } // Instance
      } // Bulkstats
    } // Configuration
  } // pm
} // Ciena PM
```


Copyright© 2022 Ciena® Corporation

ciena-pfg-profile-deviation-sds.yang

```
module ciena-pfg-profile-deviation-sds {
   namespace "http://www.ciena.com/ns/yang/ciena-pfg-profile-deviation-sds";
    prefix "pfg-dev";

    import ciena-mef-pfg-profile {
        prefix "mef-pfg";
    }

    revision 2017-06-02 {
        description "Initial revision";
    }

   deviation /mef-pfg:pfg-profiles/mef-pfg:pfg-profile/mef-pfg:description {
        deviate not-supported;
    }

    deviation /mef-pfg:pfg-profiles/mef-pfg:pfg-profile/mef-pfg:pfg-type {
        deviate replace {
            type enumeration {
                enum port-based {
                    description
                       "This represents the profiles created for port based
forwarding.";
                }
                enum l2-based {
                    description
                        "This represents the profiles created for L2 based
forwarding.";
                }
            }
        }
        deviate add {
            default port-based;
        }
    }

    deviation /mef-pfg:pfg-profiles/mef-pfg:pfg-profile/mef-pfg:pfg-group-
count {
        deviate not-supported;
    }

    deviation /mef-pfg:pfg-profiles/mef-pfg:pfg-profile/mef-pfg:group-C-
policy {
        deviate not-supported;
    }

    deviation /mef-pfg:pfg-profiles/mef-pfg:pfg-profile/mef-pfg:group-D-
policy {
        deviate not-supported;
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-pfg-profile-sds.yang

```
module ciena-pfg-profile-sds {
    namespace "http://www.ciena.com/ns/yang/ciena-pfg-profile-sds";
    prefix "ciena-pfg";

    import ciena-mef-pfg-profile {
        prefix "mef-pfg";
    }

    import ciena-mef-logical-port {
        prefix "mef-logical-port";
    }

    import ciena-types {
        prefix "ciena-types";
    }
    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
        Hanover, Maryland 21076
        U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
        "This is an extension to mef-pfg YANG model, contains
         6x specific configuration parameters.";

    revision "2017-06-02" {
        description
            "Initial version";
    }

    //
    // Definitions related to PFG
    //
    identity frwd-grps {
        description
            "This represents the group to which traffic to
             be forwarded.";
    }

    identity frwd-grp-A {
        base frwd-grps;
        description
            "Forward to group A.";
    }

    identity frwd-grp-B {
        base frwd-grps;
        description
            "Forward to group B.";
    }

    typedef flood-types {
```


Copyright© 2022 Ciena® Corporation

```
        type bits {
            bit bcast {
                description
                    "Broadcast traffic flood type.";
            }
            bit unknown-mcast {
                description
                    "Multicast traffic flood type.";
            }
            bit unknown-ucast {
                description
                    "Unknown unicast traffic flood type.";
            }
        }
        description
          "Flooded traffic types that can be forwarded. More than one traffic
             type can be specified.";
    }

    //
    // Configuration related to PFG
    //
    augment "/mef-pfg:pfg-profiles" {
        container pfg-global-config {
            description
                "Contains global l2 based PFG configuration parameters.";

            leaf pfg-global-admin-state {
                type ciena-types:admin-state;
                default enabled;

                description
                    "Enables or disables PFG on system globally.";
            }

            leaf pfg-global-port-admin-state {
                type ciena-types:admin-state;
                default disabled;

                description
                "Enables or disables PFG port-forwarding Administrative state
                     on the system globally.";
            }
        }
    }

    augment "/mef-pfg:pfg-profiles/mef-pfg:pfg-profile" {

        leaf pfg-profile-admin-state {
            type ciena-types:admin-state;
            description
               "This represents PFG admin state for L2-based PFG profiles.";
        }

        leaf flood-type {
            type flood-types;
            description
                "This represents allowed type of flooded traffic to be
forwarded.";
        }
```


Copyright© 2022 Ciena® Corporation

```
        leaf-list ingress-logical-ports {
            type mef-logical-port:logical-port-ref;
            description
              "This is a list of ingress port(s) for port-based PFG profile.";
        }

        leaf-list egress-logical-ports {
            type mef-logical-port:logical-port-ref;
            description
              "This is a list of egress port(s) for port-based PFG profile.";
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-pkix-types.yang

```
module ciena-pkix-types {
  namespace "http://www.ciena.com/ns/yang/ciena-pkix-types";
  prefix pkix-types;

  import ietf-inet-types {
    prefix inet;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module defines the data types and groupings for Public
     Key Infrastructure (PKI) Security Certificates CA (Certificate
     Authority) and CRL (Certificate Revocation List) for the NETCONF
     protocol.

     Copyright (c) 2016-2017 Ciena Corporation.  All rights reserved.

     All information contained herein is, and remains the property of
     Ciena Corporation. Dissemination of this information or
     reproduction of this material is strictly forbidden unless prior
     written permission is obtained from Ciena Corporation.";

  revision 2017-10-27 {
    description
      "Initial version";
    reference
      "RFC 6020: YANG - A Data Modeling Language for
       the Network Configuration Protocol (NETCONF).
       No specific reference; standard not available.";
  }

  /*
   * Identities
   */

  identity key-algorithm {
    description
      "Base identity from which all key-algorithms are derived.";
  }

  identity rsa1024 {
    base key-algorithm;
    description
      "The RSA algorithm using a 1024-bit key.";
    reference
      "RFC3447: Public-Key Cryptography Standards (PKCS) #1:
                RSA Cryptography Specifications Version 2.1.";
```

```
  }

  identity rsa2048 {
    base key-algorithm;
    description
      "The RSA algorithm using a 2048-bit key.";
    reference
      "RFC3447: Public-Key Cryptography Standards (PKCS) #1:
                RSA Cryptography Specifications Version 2.1.";
  }

  identity rsa3072 {
    base key-algorithm;
    description
      "The RSA algorithm using a 3072-bit key.";
    reference
      "RFC3447: Public-Key Cryptography Standards (PKCS) #1:
                RSA Cryptography Specifications Version 2.1.";
  }

//identity rsa4096 {
//  base key-algorithm;
//  description
//    "The RSA algorithm using a 4096-bit key.";
//  reference
//    "RFC3447: Public-Key Cryptography Standards (PKCS) #1:
//              RSA Cryptography Specifications Version 2.1.";
//}

//identity rsa7680 {
//  base key-algorithm;
//  description
//    "The RSA algorithm using a 7680-bit key.";
//  reference
//    "RFC3447: Public-Key Cryptography Standards (PKCS) #1:
//              RSA Cryptography Specifications Version 2.1.";
//}

//identity rsa15360 {
//  base key-algorithm;
//  description
//    "The RSA algorithm using a 15360-bit key.";
//  reference
//    "RFC3447: Public-Key Cryptography Standards (PKCS) #1:
//              RSA Cryptography Specifications Version 2.1.";
//}

//identity secp192r1 {
//  base key-algorithm;
//  description
//    "The secp192r1 algorithm.";
//  reference
//    "RFC5480:
//       Elliptic Curve Cryptography Subject Public Key Information.";
//}

  identity secp256r1 {
    base key-algorithm;
    description
      "The secp256r1 algorithm.  Also referred to in some contexts such
```


Copyright© 2022 Ciena® Corporation

```
      as RFC 4754 as ecdsa256.";
    reference
      "RFC5480:
        Elliptic Curve Cryptography Subject Public Key Information.";
  }

  identity secp384r1 {
    base key-algorithm;
    description
      "The secp384r1 algorithm.  Also referred to in some contexts such
       as RF 4754 as ecdsa384.";
    reference
      "RFC5480:
        Elliptic Curve Cryptography Subject Public Key Information.";
  }

  identity secp521r1 {
    base key-algorithm;
    description
      "The secp521r1 algorithm.  Also referred to in some contexts such
       as RFC 4754 as ecdsa521.";
    reference
      "RFC5480:
        Elliptic Curve Cryptography Subject Public Key Information.";
  }

  // key-algorithm identities used by SAOS but not included in the IETF
  // model.

  identity dsa1024 {
    base key-algorithm;
    description
      "The dsa1024 algorithm.";
    reference
      "";
  }

  identity dsa2048 {
    base key-algorithm;
    description
      "The dsa2048 algorithm.";
    reference
      "";
  }

  identity dsa3072 {
    base key-algorithm;
    description
      "The dsa3072 algorithm.";
    reference
      "";
  }

  typedef min-tls-version {
     type enumeration {
        enum tls1_0 {
           description
              "TLSv1.0 would be minimum version supported";
        }
        enum tls1_1 {
```


Copyright© 2022 Ciena® Corporation

```
            description
               "TLSv1.1 would be minimum version supported";
         }
         enum tls1_2 {
            description
               "TLSv1.2 would be minimum version supported";
         }
      }
      description
         "This enum contains the minimum TLS versions that, once set
         within a protocol, prevents the use of any TLS version
         older than the version set.";
  }

  /*
   * Groupings
   */

  grouping pkix-certificate-fields {
    leaf subject-common-name {
      type string;
      description
        "The Subject Common Name of this X.509 certificate.";
    }

    leaf issuer-common-name {
      type string;
      description
        "The certificate Issuer Common Name.";
    }

    leaf algorithm-identifier {
      type identityref {
        base "key-algorithm";
      }
      description
        "Identifies which algorithm was used to generate the key.
         Also referred to within SAOS as key-type.";
    }

    leaf valid-not-after {
      type string;
      description
        "The end of the certificate's validity period.";
    }

    description
      "Common X.509 certificate fields";
  }

  grouping pkix-file-transfer-parameters {
    leaf remote-file-uri {
      type inet:uri;
      mandatory true;
      description
        "URI (Universal Resource Identifier) of the remote file.";
    }

    leaf login-id {
      type string {
```


Copyright© 2022 Ciena® Corporation

```
        length 0..32;
      }
      description
        "Login ID for the specified server.  Not needed for TFTP.";
    }

    leaf password {
      type string;
      description
        "The string used for the clear text password.  Not needed for
         TFTP.";
    }

    leaf sftp-passphrase {
      type string;
      description
        "Passphrase for SFTP (Secure File Transfer Protocol)
         transfers.";
    }

    leaf force {
      type boolean;
      description
        "Use force to overwrite the local file with the same filename.
         Applies only to install operations.";
    }

    description
      "Common file transfer parameters";
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-pkix.yang

```
module ciena-pkix {
  namespace "http://www.ciena.com/ns/yang/ciena-pkix";
  prefix pkix;

  import ietf-yang-types {
    prefix yang;
  }

  import ciena-pkix-types {
    prefix pkix-types;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module defines the nodes for Public Key Infrastructure
     (PKI) Security Certificates CA (Certificate Authority) and CRL
     (Certificate Revocation List) for the NETCONF protocol.

     Copyright (c) 2016-2017 Ciena Corporation.  All rights reserved.

     All information contained herein is, and remains the property of
     Ciena Corporation. Dissemination of this information or
     reproduction of this material is strictly forbidden unless prior
     written permission is obtained from Ciena Corporation.";

  revision 2017-10-18 {
    description
      "Initial version";
    reference
      "RFC 6020: YANG - A Data Modeling Language for
       the Network Configuration Protocol (NETCONF).
       No specific reference; standard not available.";
  }

  /*
   * No configuration data nodes
   *
   * All configuration is done with RPCs.
   *
   */

  /*
   * Operational state data nodes
   */

  container pkix-state {
    config false;
```

```
    description
      "CA certificate and CRL state.";

    container ca-certificates {
      description
        "CA certificates.";

      list ca-certificate {
        key certificate-hash;
        description
          "The CA certificate list.";

        leaf certificate-hash {
          type yang:hex-string;
          description
            "The CA certificate hash.";
        }

        uses pkix-types:pkix-certificate-fields;
      }
    }

    container crls {
      description
        "Certificate Revocation Lists.";

      list crl {
        key crl-hash;

        leaf crl-hash {
          type yang:hex-string;
          description
            "The CRL hash.";
        }

        leaf issuer-common-name {
          type string;
          description
            "The CRL issuer common name.";
        }

        leaf last-update {
          type string;
          description
            "The CRL last update.";
        }

        leaf revoked-cert-count {
          type int32;
          description
            "The CRL revoked certificate count.";
        }
        description
          "The CRL Certificate List.";
      }
    }
  }

  /*
   * RPC Operations
```


Copyright© 2022 Ciena® Corporation

```
 */

rpc pkix-ca-install {
  description
    "Install a CA certificate.";

  input {
    uses pkix-types:pkix-file-transfer-parameters;
  }
}

rpc pkix-ca-uninstall {
  description
    "Uninstall a CA certificate.";

  input {
    leaf ca-cert-hash {
      type yang:hex-string;
      description
        "The hash identifying a particular CA certificate.";
    }
  }
}

rpc pkix-certificates-csr-generate {
  description
    "Generate a certificate signing request.";

  input {
    leaf cert-name {
      type string;
      description
        "The certificate name.";
    }

    leaf algorithm-identifier {
      type identityref {
        base "pkix-types:key-algorithm";
      }
      description
        "Identifies which algorithm is to be used to generate the
         key.  Also referred to within SAOS as key-type.";
    }

    uses pkix-types:pkix-file-transfer-parameters;
  }
}

rpc pkix-certificates-install {
  description
    "Install a certificate.";

  input {
    leaf cert-name {
      type string;
      description
        "The certificate name.";
    }

    uses pkix-types:pkix-file-transfer-parameters;
```


Copyright© 2022 Ciena® Corporation

```
    }
  }

  rpc pkix-certificates-uninstall {
    description
      "Install a certificate.";

    input {
      leaf cert-name {
        type string;
        description
          "The certificate name.";
      }
    }
  }

  rpc pkix-crl-install {
    description
      "Install a CRL.";

    input {
      leaf algorithm-identifier {
        type identityref {
          base "pkix-types:key-algorithm";
        }
        description
          "Identifies which algorithm is to be used to generate the
           key.  Also referred to within SAOS as key-type.";
      }

      uses pkix-types:pkix-file-transfer-parameters;
    }
  }

  rpc pkix-crl-uninstall {
    description
      "Uninstall a CRL.";

    input {
      leaf crl-hash {
        type yang:hex-string;
        description
          "The hash identifying a particular CRL.";
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-qos-flow.yang

```
module ciena-qos-flow {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-qos-flow";
    prefix "ciena-flow";

    import ciena-mef-qos-flow {
        prefix flow;
    }

    import ciena-mef-logical-port {
        prefix "mef-logical-port";
    }

    import ietf-inet-types {
        prefix "inet";
    }

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
                 Hanover, Maryland 21076
                 U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
        "This YANG module defines Ciena's management data definition for the
         management of a QoS flow.";

    revision "2019-05-24" {
        description
            "Added new leafs extended-resource and resource-type to support
increase in standard-profiles scale.";
    }

    revision "2018-07-17" {
        description
            "Added new globalExtended mode under profile-mode.";
        reference
            "RFC 6020: YANG - A Data Modeling Language for
            the Network Configuration Protocol (NETCONF).
            No specific reference; standard not available.";
    }

    revision "2017-02-13" {
        description
           "Making the parent-profile leaf available to global profiles and
            adding the parent mode to global profiles.";
    }

    revision "2016-05-03" {
        description
            "Removal of unused leaf qos-flow-type";
    }
```


Copyright© 2022 Ciena® Corporation

```
    revision "2016-02-03" {
        description
            "Initial version";
    }

    augment "/flow:qos-flows" {

        container ciena-qos-flows {

            leaf profile-mode {
                type enumeration {
                    enum global {
                        description
                            "Global profile mode";
                    }
                    enum standard {
                        description
                            "Standard profile mode";
                    }
                    enum globalExtended {
                        description
                            "Global extended profile mode";
                    }
                }
                description
                    "Mode of qos profile.";
            }

            leaf extended-resource {
                type enumeration {
                    enum enabled {
                        description
                         "Globally enable the extended-resource for standard-
profiles.";
                    }
                    enum disabled {
                         description
                             "Globally disable the extended-resource for
standard-profiles.";
                    }
                }
                description
                  "This leaf allows to globally control the extended-resource
for standard-profiles";
            }

            list logical-ports {
                key "logical-port";

                leaf logical-port {
                    type mef-logical-port:logical-port-ref;
                }

                leaf admin-state {
                    type boolean;
                }

                leaf classifier-mode {
                    type enumeration {
                        enum narrow;
```


Copyright© 2022 Ciena® Corporation

```
                        enum wide;
                    }
                }

                leaf mode {
                    type enumeration {
                        enum advanced;
                        enum standard-dot1dpri;
                        enum standard-ip-prec;
                        enum standard-dscp;
                        enum standard-vlan;
                        enum standard-vlan-dot1dpri;
                        enum standard-vlan-ip-prec;
                        enum standard-vlan-dscp;
                        enum hierarchical-port;
                        enum hierarchical-vlan;
                    }
                }

                leaf arp-standard-profile {
                    type flow:qos-flow-ref;
                }

                leaf nonconform-standard-profile {
                    type flow:qos-flow-ref;
                }
            }
        }
    }

    augment "/flow:qos-flows/flow:qos-flow" {

        container ciena-qos-flow {

            leaf remark-cos-policy {
                type enumeration {
                    enum remark-green;
                    enum remark-yellow;
                    enum remark-all;
                    enum remark-green-cos-only;
                    enum remark-yellow-cos-only;
                    enum remark-all-cos-only;
                }
            }

            leaf green-remark-cos {
                type uint16 {
                    range "0..63";
                }
            }

            leaf yellow-remark-cos {
                type uint16 {
                    range "0..63";
                }
            }

            leaf parent-profile {
                type flow:qos-flow-ref;
                description
```


Copyright© 2022 Ciena® Corporation

```
                "A reference to a parent qos flow.";
            }

            choice profile-type {
                case standard {

                    leaf drop {
                        type boolean;
                    }

                    leaf is-default {
                        type boolean;
                    }

                    leaf child-mode {
                        type enumeration {
                            enum dot1dpri;
                            enum dscp;
                            enum ip-prec;
                            enum vlan;
                            enum vlan-cos;
                        }
                    }
                }

                case global {

                    leaf admin-state {
                        type boolean;
                    }

                    leaf source-ip-subnet {
                        type inet:ip-address;
                    }

                    leaf destination-ip-subnet {
                        type inet:ip-address;
                    }

                    leaf parent-mode {
                        type enumeration {
                            enum enabled;
                        }
                        description
                            "Parent mode can only be set on creation and once
                              enabled cannot be disabled.";
                    }
                }
            }

            leaf resource-type {
                type enumeration {
                    enum default {
                        description
                            "When configured default, standard-profile will
consume existing resources";
                    }
                    enum extended {
                         description
```


Copyright© 2022 Ciena® Corporation

```
                              "When configured extended, standard-profile will
consume system resource type
                                  service-meter and service-counter";
                        }
                    }
                    description
                        "This leaf specifies the resource-type for a standard-
profile. Resource-type can only be
                         set on creation and once configured while creation, It
cannot be updated.";
                }
            }
        }
}
```


Copyright© 2022 Ciena® Corporation

ciena-radius.yang

```
module ciena-radius {
  namespace "http://www.ciena.com/ns/yang/ciena-radius";
  prefix "ciena-radius";

  import ietf-inet-types {
    prefix inet;
  }

  import ietf-yang-types {
    prefix yang;
  }

  import ciena-types {
    prefix ciena;
  }

  import ietf-system {
    prefix sys;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
   "This YANG module defines Ciena's global configuration for radius settings.

    Copyright (c) 2016 Ciena Corporation.  All rights
    reserved.

    All information contained herein is, and remains
    the property of Ciena Corporation. Dissemination of this
    information or reproduction of this material is strictly
    forbidden unless prior written permission is obtained from
    Ciena Corporation.";

  revision 2017-05-25 {
    description
      "Initial version";
    reference
      "RFC 6020: YANG - A Data Modeling Language for
       the Network Configuration Protocol (NETCONF).
       No specific reference; standard not available.";
  }

  /*
   * typedefs
   */
  typedef radius-search-method {
    type enumeration {
```

```
      enum priority {
        description
        "Set the search method to priority.";
      }
      enum cached {
        description
        "Set the search method to cached.";
      }
      enum load-balanced {
        description
        "Set the search method to load balanced.";
      }
    }
    description
      "RADIUS search type.";
  }

  typedef greylist-time {
    type uint16 {
      range "60 .. 14400";
    }
    units "seconds";
    description
      "A server that attempts to send a request to a server and the request
times out
       after the specified number of retries becomes greylisted. The maximum
greylist
       timeout for a RADIUS server is between 1 minute and 4 hours.";
  }

  typedef greylist-remaining {
    type uint16 {
      range "0 .. 14400";
    }
    units "seconds";
    description
     "If a server is greylisted, this represents the time remaining until the
server
       is whitelisted.  A value of 0 (zero) indicates the server is
whitelisted.";
  }

  /*
   * Identities
   */

  /*
   * Features
   */

  /*
   * Groupings
   */

  grouping server-config-attributes {
    description
      "Common RADIUS global attributes.";

    leaf timeout {
      type uint8 {
```


Copyright© 2022 Ciena® Corporation

```
        range "1 .. 30";
      }
      units "seconds";
      description
        "This is the time in seconds between re-transmissions to the
         RADIUS server.";
    }

    leaf retries {
      type uint8 {
        range "0 .. 3";
      }
      description
        "Indicates the number of times the RADIUS server should be tried
         before giving up on the server.";
    }

    // Key and secret information
    uses ciena:security-key;

    leaf search-method {
      type radius-search-method;
      description
        "The search type of the RADIUS client.";
    }
  }

  grouping server-config {
    description
      "Common RADIUS server configuration properties.";

    leaf address {
      type inet:host;
      description
        "The address or host name of the RADIUS server.";
    }

    leaf admin-state {
      type ciena:admin-state;
      description
        "The administrative state of the RADIUS server.";
    }

    leaf priority {
      type uint8 {
        range "1 .. 8";
      }
      description
        "Specifies the priority of the RADIUS servers configured on the
device.";
    }

    leaf port {
      type inet:port-number;

      description
        "The destination UDP port number to which RADIUS messages should
         be sent. The RADIUS server is not used for authentication is this
         port number is 0.";
    }
```


Copyright© 2022 Ciena® Corporation

```
}

grouping server-state {
  description
    "Common RADIUS server state properties.";

  leaf address {
    type inet:host;
    description
      "The address or host name of the RADIUS server.";
  }

  leaf operational-state {
    type ciena:oper-state;
    description
      "The operational state of the radius client.";
  }

  leaf resolved-address {
    type inet:ip-address;
    description
      "Specifies the resolved IP address if a hostname is given.";
  }
}

grouping server-statistics {
  description
    "Common RADIUS server statistics.";

  leaf resolved-address {
    type inet:ip-address;
    description
      "Specifies the resolved IP address if a hostname is given.";
  }

  leaf round-trip-time {
    type yang:timeticks;
    description
      "The time interval (in hundredths of a second) between the most
       recent Access-Reply/Access-Challenge and the Access-Request that
       matched it from this RADIUS server.";
  }

  leaf requests {
    type yang:counter32;
    description
      "The number of RADIUS Access-Request packets sent to this server.
       This does not include retransmissions.";
  }

  leaf retransmissions {
    type yang:counter32;
    description
    "The number of RADIUS Access-Request packets retransmitted to the RADIUS
       server.";
  }

  leaf access-accepts {
    type yang:counter32;
    description
```


Copyright© 2022 Ciena® Corporation

```
       "The number of RADIUS Access-Accept packets (valid or invalid) received
         from this server.";
    }

    leaf access-rejects {
      type yang:counter32;
      description
       "The number of RADIUS Access-Reject packets (valid or invalid) received
         from this server.";
    }

    leaf access-challenges {
      type yang:counter32;
      description
        "The number of RADIUS Access-Challenge packets (valid or invalid)
received
         from this server.";
    }

    leaf accounting-responses {
      type yang:counter32;
      description
        "The number of RADIUS Accounting-Response packets (valid or invalid)
received
         from this server.";
    }

    leaf malformed-responses {
      type yang:counter32;
      description
        "The number of RADIUS Malformed-Response packets (valid or invalid)
received
        from this server.  Malformed packets include packets with an invalid
length.
         Bad authenticators or signature attributes or unknown types are not
included
         as malformed access responses.";
    }

    leaf bad-authenticators {
      type yang:counter32;
      description
        "The number of RADIUS Accounting-Response packets containing invalid
authenticators
         or signature attributes received from this server.";
    }

    leaf timeouts {
      type yang:counter32;
      description
       "The number of authentication timeouts to this server.  After a timeout
the client
        may retry sending to the same server, send to a different server, or
give up.
         A retry to the same server is counted as a retransmit as well as a
timeout. A send
        to a different server is counted as a request as well as a timeout.";
    }

    leaf unknown-types {
```


Copyright© 2022 Ciena® Corporation

```
      type yang:counter32;
      description
       "The number of RADIUS packets of unknown type which were received from
this server
          on the authentication port.";
    }

    leaf packets-dropped {
      type yang:counter32;
      description
       "The number of RADIUS packets which were received from this server on
the authentication
          port and dropped for some other reason.";
    }
  }

  /*
   * Configuration data nodes
   */
  augment /sys:system {
    description
      "SAOS's concept of RADIUS configuration.";

    container radius {
      description
        "Configuration of a RADIUS client.";

      leaf admin-state {
        type ciena:admin-state;
        description
          "The administrative state of the radius client.";
      }

      leaf preferred-source-address {
         type ciena:preferred-source-address;
         description
            "The setting for which address to use with radius.";
      }

      container user-login {
        description
          "RADIUS user login authentication.";
        uses server-config-attributes;

        list user-login-servers {
          key address;
          description
            "The list of RADIUS servers";

          uses server-config;
        }
      }

      container user-accounting {
        description
          "RADIUS user login accounting.";

        uses server-config-attributes;
        leaf admin-state {
          type ciena:admin-state;
```


Copyright© 2022 Ciena® Corporation

```
      description
        "Administrative state of the client on the device.";
    }

    list user-accounting-servers {
      key address;
      description
        "The list of RADIUS servers";

      uses server-config;
    }
  }

  container dot1x-authentication {
    description
      "RADIUS 802.1x authentication.";

    uses server-config-attributes;

    leaf greylist-timeout {
      type greylist-time;
      description
     "The max greylist timeout for a RADIUS dot1x authentication server.";
    }

    leaf config-reauth {
      type ciena:on-off-state;
      description
        "Enable or disable RADIUS config reauthorization.";
    }

    list dot1x-authentication-servers {
      key address;
      description
        "The list of RADIUS servers";

      uses server-config;
    }
  }

  container dot1x-accounting {
    description
      "RADIUS 802.1x accounting.";

    uses server-config-attributes;

    leaf admin-state {
      type ciena:admin-state;
      description
        "Administrative state of the client on the device.";
    }

    leaf interval {
      type uint32;
      units "seconds";
      description
        "Set RADIUS accounting interval in seconds.";
    }

    leaf greylist-timeout {
```


Copyright© 2022 Ciena® Corporation

```
          type greylist-time;
          description
           "The max greylist timeout for a RADIUS dot1x authentication server.";
        }

        list dot1x-accounting-servers {
          key address;
          description
            "The list of RADIUS servers";

          uses server-config;
        }
      }
    }
  }

  /*
   * Operational state data nodes
   */
  augment /sys:system-state {
    description
      "SAOS's concept of RADIUS states.";

    container radius {
      config false;
      description
        "RADIUS operational state data.";

      leaf operational-state {
        type ciena:oper-state;
        description
          "The operational state of the radius client.";
      }

      container user-login {
        description
          "RADIUS user login authentication.";

        list user-login-servers {
          key address;
          description
            "The list of RADIUS servers";

          uses server-state;
        }

        leaf last-server-used {
          type inet:ip-address;
          description
            "The IP address of the last server used for user login.
           This is used by the cached search method as the first server to try.
              If the IP address is all zeros, either the server configuration
              changed or no server was used. ";
        }
      }

      container user-accounting {
        description
          "RADIUS user login accounting.";
```


Copyright© 2022 Ciena® Corporation

```
        list user-accounting-servers {
          key address;
          description
            "The list of RADIUS servers";

          uses server-state;
        }

        leaf last-server-used {
          type inet:ip-address;
          description
            "The IP address of the last server used for user accounting.
           This is used by the cached search method as the first server to try.
              If the IP address is all zeros, either the server configuration
              changed or no server was used. ";
        }
      }

      container dot1x-authentication {
        description
          "RADIUS 802.1x authentication.";

        list dot1x-authentication-servers {
          key address;
          description
            "The list of RADIUS servers";

          uses server-state;
          leaf greylist-time-remaining {
            type greylist-remaining;
            description
              "Reflects the remaining time before the server is removed from
the greylist.
                 A value of 0 (zero) indicates the current server is not
greylisted.";
          }
      }

        leaf last-server-used {
          type inet:ip-address;
          description
            "The IP address of the last server used for dot1x authentication.
          This is used by the cached search method as the first server to try.
            If the IP address is all zeros, either the server configuration
            changed or no server was used. ";
        }
      }

      container dot1x-accounting {
        description
          "RADIUS 802.1x accounting.";

        list dot1x-accounting-servers {
          key address;
          description
            "The list of RADIUS servers";

          uses server-state;
          leaf greylist-time-remaining {
            type greylist-remaining;
```

```
            description
              "Reflects the remaining time before the server is removed from
the greylist.
               A value of 0 (zero) indicates the current server is not
greylisted.";
            }
        }

        leaf last-server-used {
          type inet:ip-address;
          description
            "The IP address of the last server used for dot1x accounting.
           This is used by the cached search method as the first server to try.
              If the IP address is all zeros, either the server configuration
              changed or no server was used. ";
        }
      }

      container statistics {
        description
          "RADIUS statistics.";

        list user-login {
          key resolved-address;
          description
            "RADIUS user login statistics.";
          uses server-statistics;
        }

        list user-accounting {
          key resolved-address;
          description
            "RADIUS user accounting statistics.";
          uses server-statistics;
        }

        list dot1x-authentication {
          key resolved-address;
          description
            "RADIUS 802.1x authentication statistics.";
          uses server-statistics;
        }

        list dot1x-accounting {
          key resolved-address;
          description
            "RADIUS 802.1x accounting statistics.";
          uses server-statistics;
        }
      }
    }
  }

  /*
   * RPCs
   */
  rpc clear-statistics {
    description
      "Allows setting of the system date and time";
```


Copyright© 2022 Ciena® Corporation

```
    input {
      leaf server-type {
        type enumeration {
          enum user-login {
            description
              "Specifies the user login server.";
          }
          enum user-accounting {
            description
              "Specifies the user accounting server.";
          }
          enum dot1x-authentication {
            description
              "Specifies the 802.1x authentication server.";
          }
          enum dot1x-accounting {
            description
              "Specifies the 802.1x accounting server.";
          }
        }
        mandatory true;
        description
          "The type of RADIUS server to clear the statistics.";
      }

      leaf address {
        type inet:host;
        description
          "The address of the RADIUS server statistics to clear.  If not set
           this will clear all of the statistics for the given server type.";
      }
    }

    output {
      leaf status {
        type string;
        description
          "Status of the clear-statistics operation.";
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-removable-storage.yang

```
module ciena-removable-storage {
  namespace "http://www.ciena.com/ns/yang/ciena-removable-storage";
  prefix ciena-storage;

  import ciena-types {
    prefix ciena;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module defines nodes for the management of removable
     data storage devices.

     Copyright (c) 2017 Ciena Corporation. All rights reserved.

     All information contained herein is, and remains the property of
     Ciena Corporation. Dissemination of this information or
     reproduction of this material is strictly forbidden unless prior
     written permission is obtained from Ciena Corporation.";

  revision 2017-08-02 {
    description
      "Initial revision";
    reference
      "39XX/51XX Service Delivery, Aggregation and Virtualization
       Switches, Administration and Security, SAOS 6.16
       (009-3273-007 - Standard Revision A)
       (Ciena Corporation, December 2016)

       8700 Packetwave Platform, Administration and Security,
       SAOS 8.4 (380-1874-301 - Standard Revision A)
       (Ciena Corporation, May 2016)";
  }

  /*
   * Identities
   */

  identity storage-type {
    description
      "Base identity for removable storage types.";
  }

  identity usb {
    base storage-type;
    description
      "A removable storage device connected via USB interface.";
```

```
}

/*
 * Groupings
 */

grouping physical-index {
  description
    "The physical index of a storage device.";

  leaf physical-index {
    type union {
      type ciena:slot-index;
      type int32 {
        range "1..2147483647";
      }
    }

    description
      "The index of the slot physically interfacing with the
       storage device. Platforms having a single module should
       set this leaf to a value of 'chassis'.";
  }
}

/*
 * Configuration data nodes
 */

container removable-storage {
  description
    "Configuration parameters for removable storage devices.";

  list device {
    key "physical-index";
    description
      "List of data storage devices.";

    uses physical-index;

    leaf name {
      type string;
      description
        "The administrative name/label of the device.";
    }
    leaf enabled {
      type boolean;
      default "true";
      description
        "Administratively enables or disables the device or the
         interface to which the device is connected.";
    }
  }
}

/*
 * Operational state data nodes
 */

container removable-storage-state {
```


Copyright© 2022 Ciena® Corporation

```
    config false;
    description
      "Operational state parameters of removable storage devices.";

    list device {
      key "physical-index";
      description
        "List of data storage devices.";

      uses physical-index;

      leaf type {
        type identityref {
          base storage-type;
        }
        description
          "The general type of the device.";
      }
      leaf oper-status {
        type string {
          length "1..255";
        }
        description
          "The operational status of the device.";
      }
      leaf mount-point {
        type string;
        description
          "The file path to the mount point of the device.";
      }
      leaf vendor-id {
        type string;
        description
          "The vendor identification number associated with the
           device.";
      }
      leaf product-id {
        type string;
        description
          "The product identification number associated with the
           device.";
      }
      leaf firmware-rev {
        type string;
        description
          "The vendor-specific firmware revision number for the
           device.";
      }
      leaf hardware-rev {
        type string;
        description
          "The vendor-specific hardware revision number for the
           device.";
      }
      leaf software-rev {
        type string;
        description
          "The vendor-specific software revision number for the
           device.";
      }
```


Copyright© 2022 Ciena® Corporation

```
      leaf serial-num {
        type string;
        description
          "The vendor-specific serial number associated with the
           device.";
      }
      leaf mfg-name {
        type string;
        description
          "The name of the manufacturer of the device.";
      }
      leaf model-name {
        type string;
        description
          "The vendor-specific model name associated with the
           device.";
      }

      container memory {
        description
          "Memory information for the device.";

        leaf total {
          type uint64;
          units "kilobyte";
          description
            "The total memory capacity of the device expressed in
             kilobytes.";
        }
        leaf used {
          type uint64;
          units "kilobyte";
          description
            "The amount of used memory space on the device expressed
             in kilobytes.";
        }
        leaf free {
          type uint64;
          units "kilobyte";
          description
            "The amount of free memory space on the device expressed
             in kilobytes.";
        }
      }
    }
  }

  /*
   * RPCs
   */

  rpc format {
    description
      "Format a removable storage device.";
    input {
      uses physical-index;
    }
  }

  rpc mount {
```


Copyright© 2022 Ciena® Corporation

```
    description
      "Mount a removable storage device.";
    input {
      uses physical-index;
    }
  }

  rpc unmount {
    description
      "Unmount a removable storage device.";
    input {
      uses physical-index;
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-reserved-vlan-sds.yang

```
module ciena-reserved-vlan-sds {
   namespace "http://www.ciena.com/ns/yang/ciena-reserved-vlan-sds";
   prefix "ciena-reserved-vlan-sds";

   import ieee-types {
      prefix "ieee";
   }

   organization
      "Ciena Corporation";

   contact
      "Web URL: http://www.ciena.com/
      E-mail:  yang@ciena.com
      Postal:  7035 Ridge Road
      Hanover, Maryland 21076
      U.S.A.
      Phone:   +1 800-921-1144
      Fax:     +1 410-694-5750";

   description
      "This YANG module defines Ciena's configuration for
      Resereved-vlans for the NETCONF protocol.";

   revision "2017-02-01" {
      description
         "Initial version";
   }

   typedef reserved-vlan-ref {
      type leafref {
         path "/ciena-reserved-vlan-sds:reserved-vlans/ciena-reserved-vlan-
sds:vlan-id";
      }

      description
         "This is used by the forwarding domains of type vpls, that
         need to reference configured reserved vlan.";
   }

   container reserved-vlans {

      leaf-list  vlan-id {

         description
            "This is a pool of reserved vlans which may be used by
            forwarding domain of type vpls.";

         type ieee:vlanid;
      }
   }
}
```


Copyright© 2022 Ciena® Corporation

ciena-resource-manager-sds.yang

```
module ciena-resource-manager-sds {
  namespace "http://www.ciena.com/ns/yang/ciena-resource-manager-sds";
  prefix "ciena-resource-manager-sds";

  import ciena-resource-manager {
     prefix resource-manager;
  }

  import ciena-mef-qos-flow {
     prefix qos-flow;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
    E-mail:  yang@ciena.com
    Postal:  7035 Ridge Road
             Hanover, Maryland 21076
             U.S.A.
    Phone:   +1 800-921-1144
    Fax:     +1 410-694-5750";

  description
    "This YANG module defines the nodes for customized resource
    management of resources on 6.x platforms.

    Copyright (c) 2017 Ciena Corporation.  All rights
         reserved.

         All information contained herein is, and remains
         the property of Ciena Corporation. Dissemination of this
         information or reproduction of this material is strictly
         forbidden unless prior written permission is obtained from
         Ciena Corporation.";

  revision "2017-05-30"
  {
      description
          "Initial version";
      reference
          "RFC 6020: YANG - A Data Modeling Language for
              the Network Configuration Protocol (NETCONF).
              No specific reference; standard not available.";
  }

  /*
   * Oper Data
   */

  // Resource Usage per Qos Flow
  augment "/qos-flow:qos-flow-state/qos-flow:qos-flow" {
     description
        "Resources acquired by Qos Flow.";

     uses resource-manager:resource;
  }
```

```
}
```


Copyright© 2022 Ciena® Corporation

ciena-resource-manager.yang

```
module ciena-resource-manager {
  namespace "http://www.ciena.com/ns/yang/ciena-resource-manager";
  prefix "ciena-resource-manager";

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
    E-mail:  yang@ciena.com
    Postal:  7035 Ridge Road
             Hanover, Maryland 21076
             U.S.A.
    Phone:   +1 800-921-1144
    Fax:     +1 410-694-5750";

  description
    "This YANG module defines the nodes for customized resource
    management of resources.

    Copyright (c) 2017 Ciena Corporation.  All rights
         reserved.

         All information contained herein is, and remains
         the property of Ciena Corporation. Dissemination of this
         information or reproduction of this material is strictly
         forbidden unless prior written permission is obtained from
         Ciena Corporation.";

  revision "2017-05-30"
  {
      description
          "Initial version";
      reference
          "RFC 6020: YANG - A Data Modeling Language for
              the Network Configuration Protocol (NETCONF).
              No specific reference; standard not available.";
  }

  /*
   * Groupings
   */

  grouping property {
    list property {
      key "name";

      leaf name {
        type string;
        description
          "A unique system assigned string that identifies a particular
           property for a Resource.";
      }

      leaf description {
        type string;
        description
          "A description field that describes this Resource property.";
```

```
      }

      leaf count {
        type uint64;
        description
          "Resource Count";
      }
    }
  }

  grouping resource {
    list resource {
      key "name";

      leaf name {
        type string;
        description
          "A unique system assigned string that identifies a
           Resource type for a Resource Manager feature.";
      }

      leaf description {
        type string;
        description
          "A description field that describes this Resource type.";
      }

      uses property;
    }
  }

  grouping feature {
    list feature {
      key "name";

      leaf name {
        type string;
        description
          "A unique system assigned string that identifies a
           Resource Manager feature.";
      }

      leaf description {
        type string;
        description
          "A detailed description that describes this
           Resource Manager feature.";
      }

      uses resource;
    }
  }

  /*
   * Configuration Data
   */

  container ciena-resource-manager {
    description
      "Configuration Data for Reservation of Features Resources.";
```


Copyright© 2022 Ciena® Corporation

```
    uses feature;
  }

  /*
   * Oper Data
   */

  container ciena-resource-manager-state {
    description
      "Operational Data for Features Resources.";

    config false;

    uses feature;
  }

  /*
   * RPC
   */

  rpc ciena-resource-manager-validate {
    description
      "Validate Resources configured by operator.";

    output {
      leaf status {
        type string;
        description
          "Status of Resources Validation operation.";
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-rmon.yang

```
module ciena-rmon {
  namespace "http://www.ciena.com/ns/yang/ciena-rmon";
  prefix ciena-rmon;

  import ciena-file-transfer-types {
    prefix ciena-ftt;
  }

  import ietf-yang-types {
    prefix "yang";
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module defines nodes for the management of Remote
     Network Monitoring (RMON) devices through the NETCONF protocol.

     RMON supports network monitoring features described in RFC 2819
     and can be used to enable a device to gather information related
     to groups defined in RFC 1757 and RFC 2021. A network monitoring
     device can be used to probe the network to retrieve this
     information by sampling SNMP MIB objects.

     Copyright (c) 2017 Ciena Corporation. All rights reserved.

     All information contained herein is, and remains the property of
     Ciena Corporation. Dissemination of this information or
     reproduction of this material is strictly forbidden unless prior
     written permission is obtained from Ciena Corporation.";

  revision 2017-10-27 {
    description
      "Initial revision";
    reference
      "39XX/51XX Service Delivery, Aggregation and Virtualization
       Switches, Fault and Performance Management, SAOS 6.16
       (009-3273-009 - Standard Revision A)
       (Ciena Corporation, December 2016)

       8700 Packetwave Platform, Fault and Performance Management,
       SAOS 8.4 (380-1874-520 - Standard Revision A)
       (Ciena Corporation, May 2016)";
  }

  /*
   * Type definitions
   */
```


Copyright© 2022 Ciena® Corporation

```
typedef owner-string {
  type string {
    length "1..127";
  }
  description
    "This data type is used to model an administratively assigned
     name of the owner of a resource. Implementations must accept
     values composed of well-formed NVT ASCII sequences. In
     addition, implementations should accept values composed of
     well-formed UTF-8 sequences.

     It is suggested that this name contain one or more of the
     following: IP address, management station name, network
     manager's name, location, or phone number. In some cases the
     agent itself will be the owner of an entry. In these cases,
     this string shall be set to a string starting with 'monitor'.

     SNMP access control is articulated entirely in terms of the
     contents of MIB views; access to a particular SNMP object
     instance depends only upon its presence or absence in a
     particular MIB view and never upon its value or the value of
     related object instances. Thus, objects of this type afford
     resolution of resource contention only among cooperating
     managers; they realize no access control function with respect
     to uncooperative parties.";
  reference
    "RFC 2819: Remote Network Monitoring Management Information
               Base";
}

typedef sample-type {
  type enumeration {
    enum absolute {
      description
        "The value of a selected variable at the last sample.";
    }
    enum delta {
      description
        "The difference between the value of a selected variable at
         the last sample and its current value.";
    }
  }
  description
    "The method of sampling a selected variable and calculating the
     value to be used for comparison. If the value of this type is
     'absolute', the value of the selected variable at the last
     sample will be used for direct comparison. If the value of
     this type is 'delta', the value of the selected variable at
     the last sample will be subtracted from the current value, and
     the difference will be used for comparison.";
}

/*
 * Configuration data nodes
 */

container rmon {
  description
    "Configuration parameters for the management of RMON devices.";
```


Copyright© 2022 Ciena® Corporation

```
list alarm {
  key alarm-index;
  description
    "A list of configurable parameters that set up a periodic
     checking for alarm conditions.";

  leaf alarm-index {
    type uint16 {
      range "1..65535";
    }
    description
      "An index that uniquely identifies an element in the alarm
       list. Each such element defines a diagnostic sample at a
       particular interval for an object on the device.";
  }
  leaf interval {
    type uint32;
    units seconds;
    mandatory true;
    description
      "The interval over which the data is sampled and compared
       with the rising and falling thresholds. When setting this
       leaf, care should be taken in the case of delta-value
       sampling - the interval should be set short enough that
       the sampled variable is very unlikely to increase or
       decrease by more than 2^31 - 1 during a single sampling
       interval.";
  }
  leaf data-source {
    type yang:object-identifier;
    mandatory true;
    description
      "The SNMP object identifier of the particular variable to
       be sampled. Only variables that resolve to an ASN.1
       primitive type of INTEGER (INTEGER, Integer32, Counter32,
       Counter64, Gauge, or TimeTicks) may be sampled.";
  }
  leaf sample-type {
    type sample-type;
    mandatory true;
    description
      "The method of sampling the selected variable and
       calculating the value to be compared against the
       thresholds. If the value of this leaf is 'absolute', the
       value of the selected variable will be compared directly
       with the thresholds at the end of the sampling interval.
       If the value of this leaf is 'delta', the value of the
       selected variable at the last sample will be subtracted
       from the current value, and the difference compared with
       the thresholds.";
  }
  leaf startup-alarm {
    type enumeration {
      enum rising {
        description
          "If the first sample after this list element is created
           is greater than or equal to the rising-threshold, a
           single rising alarm will be generated.";
      }
```


Copyright© 2022 Ciena® Corporation

```
      enum falling {
        description
          "If the first sample after this list element is created
           is less than or equal to the falling-threshold, a
           single falling alarm will be generated.";
      }
      enum rising-or-falling {
        description
          "If the first sample after this list element is created
           is greater than or equal to the rising-threshold, a
           single rising alarm will be generated. If that sample
           is less than or equal to the falling-threshold, a
           single falling alarm will be generated.";
      }
    }
    mandatory true;
    description
      "The alarm that may be sent when this list element is first
       created. If the first sample after this element is created
       is greater than or equal to the rising-threshold and
       startup-alarm is equal to 'rising' or 'rising-or-falling',
       then a single rising alarm will be generated. If the first
       sample after this element is created is less than or equal
       to the falling-threshold and startup-alarm is equal to
       'falling' or 'rising-or-falling', then a single falling
       alarm will be generated.";
  }
  leaf rising-threshold {
    type int32 {
      range "0..2147483647";
    }
    mandatory true;
    description
      "A threshold for the sampled statistic. When the current
       sampled value is greater than or equal to this threshold,
       and the value at the last sampling interval was less than
       this threshold, a single event will be generated. A single
       event will also be generated if the first sample after
       this element becomes set is greater than or equal to this
       threshold and the associated startup-alarm is equal to
       rising-alarm or rising-or-falling-alarm.

       After a rising event is generated, another such event will
       not be generated until the sampled value falls below this
       threshold and reaches the falling-threshold.";
  }
  leaf falling-threshold {
    type int32 {
      range "0..2147483647";
    }
    mandatory true;
    description
      "A threshold for the sampled statistic. When the current
       sampled value is less than or equal to this threshold, and
       the value at the last sampling interval was greater than
       this threshold, a single event will be generated. A single
       event will also be generated if the first sample after
       this element becomes set is less than or equal to this
       threshold and the associated startup-alarm is equal to
       falling-alarm or rising-or-falling-alarm.
```


Copyright© 2022 Ciena® Corporation

```
        After a falling event is generated, another such event
        will not be generated until the sampled value rises above
        this threshold and reaches the rising-threshold.";
    }
    leaf rising-event-index {
      type uint16;
      mandatory true;
      description
        "The index of an element from the event list that is used
         when a rising-threshold is crossed. The event identified
         by a particular value of this index is the same as
         identified by the same value of the event-index leaf. If
         there is no corresponding element in the event list, then
         no association exists. In particular, if this value is
         zero, no associated event will be generated, as zero is
         not a valid event-index.";
    }
    leaf falling-event-index {
      type uint16;
      mandatory true;
      description
        "The index of an element from the event list that is used
         when a falling-threshold is crossed. The event identified
         by a particular value of this index is the same as
         identified by the same value of the event-index leaf. If
         there is no corresponding element in the event list, then
         no association exists. In particular, if this value is
         zero, no associated event will be generated, as zero is
         not a valid event-index.";
    }
    leaf owner {
      type owner-string;
      description
        "The entity that configured this element and is therefore
         using the resources assigned to it.";
    }
  } // end list alarm

  list event {
    key event-index;
    description
      "A list of configurable parameters that describe an event to
       be generated when certain conditions are met.";

    leaf event-index {
      type uint16 {
        range "1..65535";
      }
      description
        "An index that uniquely identifies an element in the event
         list. Each such element defines one event that is to be
         generated when the appropriate conditions occur.";
    }
    leaf description {
      type string {
        length "0..127";
      }
      description
        "A comment describing this event element.";
```


Copyright© 2022 Ciena® Corporation

```
    }
    leaf type {
      type enumeration {
        enum none {
          description
            "No notification will be made about the event.";
        }
        enum log {
          description
            "An entry will be made in the log for the event.";
        }
        enum snmp-trap {
          description
            "An SNMP trap will be sent to one or more management
             stations for the event.";
        }
        enum log-and-trap {
          description
            "An entry will be made in the log and an SNMP trap will
             be sent for the event.";
        }
      }
      mandatory true;
      description
        "The type of notification that the probe will make about
         this event. In the case of log, an entry is made in the
         log table for each event. In the case of snmp-trap, an
         SNMP trap is sent to one or more management stations.";
    }
    leaf community {
      type string {
        length "0..127";
      }
      description
        "If an SNMP trap is to be sent, it will be sent to the SNMP
         community specified by this leaf.";
    }
    leaf owner {
      type owner-string;
      description
        "The entity that configured this element and is therefore
         using the resources assigned to it.";
    }
  } // end list event

  list history {
    key history-index;
    description
      "A list of configurable parameters that set up a periodic
       sampling of statistics.";

    leaf history-index {
      type uint16 {
        range "1..65535";
      }
      description
        "An index that uniquely identifies an element in the
         history list. Each such element defines a set of samples
         at a particular interval for an interface on the device.";
    }
```


Copyright© 2022 Ciena® Corporation

```
leaf data-source {
  type yang:object-identifier;
  mandatory true;
  description
    "This leaf identifies the source of the data for which
     historical data was collected and placed in a media-
     specific SNMP table on behalf of this history element.
     This source can be any interface on this device. In order
     to identify a particular interface, this leaf shall
     identify the instance of the SNMP ifIndex object, defined
     in RFC 2233 [17], for the desired interface. For example,
     if an element were to receive data from interface #1, this
     leaf would be set to ifIndex.1.

     The statistics in this group reflect all packets on the
     local network segment attached to the identified
     interface.";
}
leaf buckets-requested {
  type uint16 {
    range "1..65535";
  }
  default "50";
  description
    "The requested number of discrete time intervals over which
     data is to be saved in the part of the media-specific
     SNMP table associated with this history element.

     When this leaf is created or modified, the probe should
     set /rmon-state/history/buckets-granted as closely to this
     leaf as is possible for the particular probe
     implementation and available resources.";
}
leaf interval {
  type uint32;
  units seconds;
  default "1800";
  description
    "The interval over which the data is sampled for each
     bucket in the part of the media-specific SNMP table
     associated with this element. This interval can be set to
     any duration between 1 second and 1 hour.

     Because the SNMP counters in a bucket may overflow at
     their maximum value with no indication, a prudent manager
     will take into account the possibility of overflow in any
     of the associated counters. It is important to consider
     the minimum time in which any counter could overflow on a
     particular media type and set the interval leaf to a value
     less than this interval. This is typically most important
     for the 'octets' counter in any media-specific SNMP table.
     For example, on an Ethernet network, the SNMP
     etherHistoryOctets counter could overflow in about one
     hour at the Ethernet's maximum utilization.";
}
leaf owner {
  type owner-string;
  description
    "The entity that configured this element and is therefore
     using the resources assigned to it.";
```


Copyright© 2022 Ciena® Corporation

```
}
leaf file-logging {
  type boolean;
  default "false";
  description
    "Enables or disables file-logging. The statistics for this
     history element will be logged to a file when the value of
     this leaf is 'true'.";
}
leaf sample-type {
  type sample-type;
  default "absolute";
  description
    "Whether the value of this history element is absolute or
     a delta. If the value of this leaf is 'absolute', then
     this history element represents the value of the
     associated data-source across all sampling intervals. If
     the value of this leaf is 'delta' then this history
     element represents the value of the associated data-source
     since the last sampling.";
}
leaf statistics {
  type bits {
    bit standard {
      position 0;
      description
        "Collect basic-error, basic-rx, and basic-tx
         statistics.";
    }
    bit basic-error {
      position 1;
      description
        "Collect basic-error statistics.";
    }
    bit basic-rx {
      position 2;
      description
        "Collect basic-rx statistics.";
    }
    bit basic-tx {
      position 3;
      description
        "Collect basic-tx statistics.";
    }
    bit all-error {
      position 4;
      description
        "Collect all-error statistics.";
    }
    bit all-rx {
      position 5;
      description
        "Collect all-rx statistics.";
    }
    bit all-tx {
      position 6;
      description
        "Collect all-tx statistics.";
    }
  }
```


Copyright© 2022 Ciena® Corporation

```
      default "standard";
      description
        "The type of statistics to be collected by this history
         element. Only the statistics that are configured to be
         collected at the point the data is collected are stored.";
    }
  } // end list history

  list statistics {
    key stats-index;
    description
      "A list of configurable parameters that set up a collection
       of statistics kept for a particular Ethernet interface.";

    leaf stats-index {
      type uint16 {
        range "1..65535";
      }
      description
        "An index that uniquely identifies an element in the
         statistics list.";
    }
    leaf data-source {
      type yang:object-identifier;
      mandatory true;
      description
        "This leaf identifies the source of the data that this
         statistics element is configured to analyze. This source
         can be any Ethernet interface on this device. In order to
         identify a particular interface, this leaf shall identify
         the instance of the SNMP ifIndex object, defined in
         RFC 2233 [17], for the desired interface. For example, if
         an element were to receive data from interface #1, this
         leaf would be set to ifIndex.1.

         The statistics in this group reflect all packets on the
         local network segment attached to the identified
         interface.";
    }
    leaf owner {
      type string {
        length "1..127";
      }
      description
        "The entity that configured this entry and is therefore
         using the resources assigned to it.";
    }
  } // end list statistics

  container transfer {
    description
      "File-transfer parameters that can be configured to push a
       file containing RMON history and user-history data to a
       server at a specified interval.";

    leaf enabled {
      type boolean;
      default "false";
      description
        "Enables or disables RMON file transfer.";
```


Copyright© 2022 Ciena® Corporation

```
    }
    leaf name {
      type string {
        length "1..16";
      }
      description
        "An arbitrary name for this file persistence
         configuration.";
    }
    leaf interval {
      type uint32;
      units seconds;
      default "3600";
      description
        "The interval at which the specified file will be pushed to
         the server. The value of this interval must be between
         one hour and one year.";
    }
    leaf kept-files {
      type uint8 {
        range "1..4";
      }
      default "4";
      description
        "The number of user files to be kept. When the number of
         current files is equal to the value of this leaf, new
         files will replace the oldest.";
    }
    leaf remote-dir {
      type string {
        length "0..127";
      }
      description
        "The path of the remote directory to which RMON history
         files will be pushed.";
    }
    uses ciena-ftt:file-transfer-parameters;
  } // end container transfer

  container user-history {
    description
      "Configurable parameters for RMON user-history. Nodes in this
       container combine mechanisms seen in the alarm and history
       lists to provide user-specified history collection.";
    reference
      "RFC 4502: Remote Network Monitoring Management Information
                 Base Version 2";

    list control {
      key control-index;
      description
        "A list of configurable parameters that set up a group of
         user-defined SNMP MIB objects to be sampled periodically
         (called a bucket-group).";

      leaf control-index {
        type uint16 {
          range "1..65535";
        }
        description
```


Copyright© 2022 Ciena® Corporation

```
        "An index that uniquely identifies an element in the
         user-history control list. Each such element defines a
         set of samples at a particular interval for a specified
         set of SNMP MIB instances available from the managed
         system.";
    }
    leaf buckets-requested {
      type uint16 {
        range "1..65535";
      }
      default "50";
      description
        "The requested number of discrete time intervals over
         which data is to be saved in the part of the SNMP
         usrHistoryTable associated with this user-history
         control list element.

         When this leaf is created or modified, the probe should
         set buckets-granted as closely to this leaf as is
         possible for the particular probe implementation and
         available resources.";
    }
    leaf interval {
      type uint32;
      units seconds;
      default "1800";
      description
        "The interval in seconds over which the data is sampled
         for each bucket in the part of the SNMP usrHistoryTable
         associated with this user-history control list element.

         Because the counters in a bucket may overflow at their
         maximum value with no indication, a prudent manager will
         take into account the possibility of overflow in any of
         the associated counters. It is important to consider the
         minimum time in which any counter could overflow on a
         particular media type and set the interval leaf to a
         value less than this interval.";
    }
    leaf owner {
      type owner-string;
      description
        "The entity that configured this element and is therefore
         using the resources assigned to it.";
    }
    leaf file-logging {
      type boolean;
      default "false";
      description
        "Enables or disables file-logging. The statistics for
         this user-history element will be logged to a file when
         the value of this leaf is 'true'.";
    }
  } // end list control

  list object {
    key object-index;
    description
      "A list of configurable parameters that set up SNMP MIB
       instances to be sampled periodically.";
```


Copyright© 2022 Ciena® Corporation

```
      leaf object-index {
        type uint16 {
          range "1..65535";
        }
        description
          "An index that uniquely identifies an element in the
           user-history object list. Each such element defines an
           SNMP MIB instance to be collected periodically.";
      }
      leaf control-index {
        type leafref {
          path "/rmon/user-history/control/control-index";
        }
        mandatory true;
        description
          "An index that uniquely identifies an element in the
           user-history control list that is associated with this
           element of the object list.";
      }
      leaf data-source {
        type yang:object-identifier;
        mandatory true;
        description
          "The SNMP object identifier of the particular variable to
           be sampled. Only variables that resolve to an ASN.1
           primitive type of Integer32 (Integer32, Counter, Gauge,
           or TimeTicks) may be sampled.";
      }
      leaf sample-type {
        type sample-type;
        default "absolute";
        description
          "The method of sampling the selected variable for storage
           in the user-history object list. If the value of this
           leaf is 'absolute', the value of the selected variable
           will be copied directly into the history bucket. If the
           value of this leaf is 'delta', the value of the selected
           variable at the last sample will be subtracted from the
           current value, and the difference will be stored in the
           history bucket.";
      }
    } // end list object
  } // end container user-history
} // end container rmon

/*
 * Operational state data nodes
 */

container rmon-state {
  config false;
  description
    "Operational state parameters of RMON devices.";

  list history {
    key history-index;
    description
      "A list of operational state parameters for periodic sampling
       of statistics.";
```


Copyright© 2022 Ciena® Corporation

```
    leaf history-index {
      type uint16 {
        range "1..65535";
      }
      description
        "An index that uniquely identifies an element in the
         history list. Each such element defines a set of samples
         at a particular interval for an interface on the device.";
    }
    leaf buckets-granted {
      type uint16 {
        range "1..65535";
      }
      description
        "The number of discrete sampling intervals over which data
         shall be saved in the part of the media-specific SNMP
         table associated with this history element.

         When the associated /rmon/history/buckets-requested leaf
         is created or modified, the probe should set this leaf as
         closely to the requested value as is possible for the
         particular probe implementation and available resources.
         The probe must not lower this value except as a result of
         a modification to the associated buckets-requested leaf.

         There will be times when the actual number of buckets
         associated with this element is less than the value of
         this leaf. In this case, at the end of each sampling
         interval, a new bucket will be added to the media-specific
         SNMP table.

         When the number of buckets reaches the value of this
         leaf and a new bucket is to be added to the media-specific
         SNMP table, the oldest bucket associated with this element
         shall be deleted by the agent so that the new bucket can
         be added.

         When the value of this leaf changes to a value less than
         the current value, entries are deleted from the
         media-specific SNMP table associated with this element.
         Enough of the oldest of these entries shall be deleted by
         the agent so that their number remains less than or equal
         to the new value of this leaf.

         When the value of this leaf changes to a value greater
         than the current value, the number of associated media-
         specific SNMP entries may be allowed to grow.";
    }
  } // end list history

  container transfer {
    description
      "Operational state parameters for RMON file-transfer.
       File-transfer is responsible for pushing a file containing
       RMON history and user-history data to a server at a
       specified interval.";

    leaf last-push-name {
      type string;
```


Copyright© 2022 Ciena® Corporation

```
      description
        "The remote path and filename to which the last file push
         was attempted.";
    }
    leaf last-push-state {
      type string;
      description
        "The status of the last attempted file push.";
    }
    leaf last-push-time {
      type yang:date-and-time;
      description
        "The date and time of the last attempted file push.";
    }
    leaf max-kept-files {
      type uint8;
      default "4";
      description
        "The maximum number of user-files that can be kept.";
    }
  } // end container transfer

  container user-history {
    description
      "Operational state parameters of RMON user-history. Nodes in
       this container combine mechanisms seen in the alarm and
       history lists to provide user-specified history
       collection.";
    reference
      "RFC 4502: Remote Network Monitoring Management Information
                 Base Version 2";

    list control {
      key control-index;
      description
        "A list of operational state parameters for a group of
         user-defined SNMP MIB objects to be sampled periodically
         (called a bucket-group).";

      leaf control-index {
        type uint16 {
          range "1..65535";
        }
        description
          "An index that uniquely identifies an element in the
           user-history control list. Each such element defines a
           set of samples at a particular interval for a specified
           set of SNMP MIB instances available from the managed
           system.";
      }
      leaf objects {
        type uint16 {
          range "1..65535";
        }
        description
          "The number of SNMP MIB objects to be collected in the
           portion of the SNMP usrHistoryTable associated with this
           user-history control element.";
      }
      leaf buckets-granted {
```


Copyright© 2022 Ciena® Corporation

```
          type uint16 {
            range "1..65535";
          }
          description
            "The number of discrete sampling intervals over which
             data shall be saved in the part of the SNMP
             usrHistoryTable associated with this user-history
             control element.

             When the associated
             /rmon/user-history/control/buckets-requested leaf is
             created or modified, the probe should set this leaf as
             closely to the requested value as is possible for the
             particular probe implementation and available resources.
             The probe must not lower this value except as a result
             of a modification to the associated buckets-requested
             leaf.

             The associated buckets-requested leaf should be set
             before or at the same time as this leaf to allow the
             probe to accurately estimate the resources required for
             this user-history control element.

             There will be times when the actual number of buckets
             associated with this element is less than the value of
             this leaf. In this case, at the end of each sampling
             interval, a new bucket will be added to the SNMP
             usrHistoryTable.

             When the number of buckets reaches the value of this
             leaf and a new bucket is to be added to the SNMP
             usrHistoryTable, the oldest bucket associated with this
             user-history control element shall be deleted by the
             agent so that the new bucket can be added.

             When the value of this leaf changes to a value less
             than the current value, entries are deleted from the
             SNMP usrHistoryTable associated with this user-history
             control element. Enough of the oldest of these entries
             shall be deleted by the agent so that their number
             remains less than or equal to the new value of this
             leaf.

             When the value of this leaf changes to a value greater
             than the current value, the number of associated
             SNMP usrHistory entries may be allowed to grow.";
        }
      } // end list control
    } // end container user-history
  }

  /*
   * RPCs
   */

  rpc rmon-transfer-push {
    description
      "Manually push a file containing RMON history and user-history
       data to the server configured under the /rmon/transfer
       container.";
```


Copyright© 2022 Ciena® Corporation

```
  }
} // end module ciena-rmon
```


Copyright© 2022 Ciena® Corporation

ciena-sat-deviation.yang

```
module ciena-sat-deviation {
    namespace "http://www.ciena.com/ns/yang/ciena-sat-deviation-sds";
    prefix "sat-dev";

    import ciena-sat {
        prefix "ciena-sat";
    }

    revision 2019-06-11 {
        description "Initial revision";
    }

   deviation /ciena-sat:sat/ciena-sat:emix-sequence/ciena-sat:u-frame-size {
        deviate replace {
            type uint32 {
                range "64..9216";
            }
        }
    }

    deviation /ciena-sat:sat/ciena-sat:test-profile/ciena-sat:frame-size {
        deviate replace {
            type uint32 {
                range "64..9216";
            }
        }
    }

    deviation /ciena-sat:sat/ciena-sat:test-profile/ciena-sat:frameloss-
decrement-type {
        deviate not-supported;
    }

    deviation /ciena-sat:sat/ciena-sat:test-profile/ciena-sat:source-mac-
validation {
        deviate not-supported;
    }

    deviation /ciena-sat:sat/ciena-sat:test-profile/ciena-sat:inner-vid-
validation {
        deviate not-supported;
    }

    deviation /ciena-sat:sat/ciena-sat:test-profile/ciena-sat:measurement-
direction {
        deviate not-supported;
    }

    deviation /ciena-sat:sat-state/ciena-sat:emix-frame-size-for-sat-entity/
ciena-sat:emix-frame-size {
        deviate replace {
            type uint32 {
                range "64..9216";
            }
        }
    }
```


Copyright© 2022 Ciena® Corporation

```
    deviation /ciena-sat:sat-state/ciena-sat:emix-frame-size-for-sat-entity/
ciena-sat:avg-frame-size {
        deviate replace {
            type uint32 {
                range "64..9216";
            }
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-sat.yang

```
module ciena-sat {
  namespace "http://www.ciena.com/ns/yang/ciena-sat";
  prefix "ciena-sat";

  import ietf-yang-types {
    prefix yang;
  }

  import ieee-types
  {
    prefix ieee;
  }

  import ciena-types {
    prefix ciena;
  }

  import ietf-inet-types {
    prefix inet;
  }

  import ciena-mef-logical-port {
     prefix "mef-logical-port";
  }

  import ciena-mef-fp {
     prefix "mef-fp";
  }

  organization
   "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
    E-mail:  yang@ciena.com
    Postal:  7035 Ridge Road
    Hanover, Maryland 21076
    U.S.A.
    Phone:   +1 800-921-1144
    Fax:     +1 410-694-5750";

  description
    "This YANG module defines Ciena's data definition for the
    management of benchmark Service Activation Testing.

    Copyright (c) 2016 Ciena Corporation.  All rights
    reserved.

    All information contained herein is, and remains
    the property of Ciena Corporation. Dissemination of this
    information or reproduction of this material is strictly
    forbidden unless prior written permission is obtained from
    Ciena Corporation.";

  revision "2019-06-11" {
    description
```

```
      "benchmark support with port having 1:1 LAG/MC-LAG and Y-protection
config";
  }

  revision "2019-05-13" {
     description
       "Added new optional data for entity, test-instance, test-profile.
        New config attributes to the test profile to control the frameloss
        test decrements, measument direction and inner-vid-validation for
        qinq traffic.
        More details on the test progress captured in the
        gen-test-session-allocation object.
        More info provided with each result object including
        average emix frame size, u and h frame size if emix was used
        for testing, as well as interval and duration setting at the
        time of testing.";
     reference
       "Carrier Ethernet Service Activation Testing (MEF 48), Oct 2014";
  }

  revision "2019-02-26" {
    description
      "Added new enum value tCusom in duration and a new test-profile leaf
custom-duration.";
  }

  revision "2018-04-13" {
    description
      "Added new outage state.";
    reference
      "Carrier Ethernet Service Activation Testing (MEF 48), Oct 2014";
  }

  revision "2017-12-07" {
    description
      "Added support for outage test.";
    reference
      "Carrier Ethernet Service Activation Testing (MEF 48), Oct 2014";
  }

  revision "2017-10-26" {
    description
      "Changed entity's role to a leaf instead of a choice.
       Changed entity's port to be a logical-port reference
       Replaced test instance's sub-port ID for flow point ID - refence to
        an MEF flow point
       Added active dst MAC, active s-vid and active c-vid leaf to the
        test instance operational data";
    reference
      "Carrier Ethernet Service Activation Testing (MEF 48), Oct 2014";
  }

  revision "2016-11-01" {
    description
      "Initial Version.";
    reference
      "Carrier Ethernet Service Activation Testing (MEF 48), Oct 2014";
  }

  /*
```


Copyright© 2022 Ciena® Corporation

```
 * typedefs
 */

typedef name-string {
  type string {
    length "1..45";
  }
  description
    "A text string of up to 45 characters.";
}

typedef latency-pdv-test-state-type {
  type enumeration {
    enum idle {
      description
        "Test is not configured or hasn't started yet.";
    }
    enum sending-traffic {
      description
        "Sending test packets.";
    }
    enum waiting-for-timestamp-data {
      description
        "Collecting latency data.";
    }
    enum waiting-for-residual-packets {
      description
        "Waiting for all test packets to return back to collector.";
    }
    enum processing-results {
      description
        "Processing latency data collected.";
    }
    enum stopped-by-interval-timer {
      description
        "Test stopped by the interval timer.";
    }
    enum stopped-by-duration-timer {
      description
        "Test stopped by the duration timer.";
    }
    enum stopped-by-user {
      description
        "Test stopped by the user.";
    }
    enum done {
      description
        "Testing complete.";
    }
  }
  description
   "Possible states for the latency and packet delay variation tests.";
}

typedef throughput-test-state-type {
  type enumeration {
    enum idle {
      description
        "Test is not configured or hasn't started yet.";
    }
```


Copyright© 2022 Ciena® Corporation

```
      enum running {
        description
          "Test traffic is being sent.";
      }
      enum waiting-for-residual-packets {
        description
          "Waiting for all test packets to return back to collector.";
      }
      enum processing-results {
        description
          "Processing throughput results.";
      }
      enum stopped-by-interval-timer {
        description
          "Test stopped by the interval timer.";
      }
      enum stopped-by-duration-timer {
        description
          "Test stopped by the duration timer.";
      }
      enum stopped-by-user {
        description
          "Test stopped by the user.";
      }
      enum done {
        description
          "Test complete.";
      }
      enum tx-max-throughput-for-yellow-test {
        description
          "Transitting green test frames at maximum throughput found, while
           yellow test frames are sent in search of yellow maximum throughput.";
      }
    }
    description
     "Throughput test states";
  }

  typedef frameloss-test-state-type {
    type enumeration {
      enum idle {
        description
          "Test is not configured or hasn't started yet.";
      }
      enum running-first-test {
        description
          "Sending test packets for the first frame loss pass.";
      }
      enum WaitingForResidualFirstPackets {
        description
          "Waiting for all test packets from the first frame loss pass to
           return back to collector.";
      }
      enum processing-first-results {
        description
          "Processing results for the first frame loss pass.";
      }
      enum running-second-test {
        description
          "Sending test packets for the first frame loss pass.";
```


Copyright© 2022 Ciena® Corporation

```
      }
      enum waiting-for-residual-second-packets {
        description
          "Waiting for all test packets from the second frame loss pass
           to return back to collector.";
      }
      enum processing-second-results {
        description
          "Processing results for the second frame loss pass.";
      }
      enum stopped-by-interval-timer {
        description
          "Test stopped by the interval timer.";
      }
      enum stopped-by-duration-timer {
        description
          "Test stopped by the duration timer.";
      }
      enum stopped-by-user {
        description
          "Test stopped by the user.";
      }
      enum done {
        description
          "Test complete.";
      }
    }
    description
     "Frame loss test states";
  }

  typedef rfc2544-test-state-type {
    type enumeration {
      enum idle {
        description
          "Test is not configured or hasn't started yet.";
      }
      enum running {
        description
          "The RFC 1544 test suite is currently running.
           Refer to the throughput, frame loss and latency test state
           to find out which test from the suite is currently executing.";
      }
      enum stopped-by-interval-timer {
        description
          "Test stopped by the interval timer.";
      }
      enum stopped-by-duration-timer {
        description
          "Test stopped by the duration timer.";
      }
      enum stopped-by-user {
        description
          "Test stopped by the user.";
      }
      enum done {
        description
          "Test complete.";
      }
    }
```


Copyright© 2022 Ciena® Corporation

```
    description
     "RFC 2544 test suite state.";
  }

  typedef y1564-test-state-type {
    type enumeration {
      enum idle {
        description
          "Test is not configured or hasn't started yet.";
      }
      enum running {
        description
          "The Y.1564 test suite is currently running.
           Refer to the throughput, frame loss, latency and PDV test state
           to find out which test from the suite is currently executing.";
      }
      enum stopped-by-interval-timer {
        description
          "Test stopped by the interval timer.";
      }
      enum stopped-by-duration-timer {
        description
          "Test stopped by the duration timer.";
      }
      enum stopped-by-user {
        description
          "Test stopped by the user.";
      }
      enum done {
        description
          "Test complete.";
      }
    }
    description
     "Y1564 test suite state.";
  }

  typedef outage-test-state-type {
    type enumeration {
      enum idle {
        description
          "Test is not configured or hasn't started yet.";
      }
      enum running {
        description
          "The outage test is currently running.";
      }
      enum stopped-by-duration-timer {
        description
          "Test stopped by the duration timer.";
      }
      enum stopped-by-user {
        description
          "Test stopped by the user.";
      }
      enum processing-outage-records {
        description
          "Processing outage test records.";
      }
      enum waiting-for-residual-packets {
```


Copyright© 2022 Ciena® Corporation

```
        description
          "Waiting for all test packets to return back to collector.";
      }
      enum done {
        description
          "Test complete.";
      }
    }
    description
     "outage test state.";
  }

  typedef priority {
    type uint8 {
      range "0..7";
    }
    description
      "Range of priorities from 0-7 inclusive.";
  }

  typedef basic-color
  {
    type enumeration {
      enum green {
        description
          "Green traffic.";
      }
      enum yellow {
        description
          "Yellow traffic.";
      }
    }
    description
     "Traffic color which is associated with the dot1q DEI
      (drop eligible indictor) bit value.";
  }

  typedef color {
    type enumeration {
      enum green {
        description
          "The test is configured to run with green traffic only.";
      }
      enum yellow {
        description
          "The test is configured to run with yellow traffic only.";
      }
      enum green-yellow {
        description
          "The test is configured to run with green traffic starting at
           the configured bandwidth and yellow traffic starting at
           the configured excess bandwidth.";
      }
      enum green-yellow-red {
        description
          "The test is configured to run with green traffic starting at
           the configured bandwidth and yellow traffic starting at 1.25
           times the configured excess bandwidth.";
      }
    }
```


Copyright© 2022 Ciena® Corporation

```
    description
     "Defines which traffic color needs to be tested. Green
      test traffic is generated with DEI bit set to 0 and
      uses the profile bandwidth parameter as the starting
      bandwidth; yellow test traffic is generated with DEI bit
      set to 1 and uses the profile excess-bandwidth parameter
      as the starting bandwidth. When testing for red, the test
      stream has its DEI bit set to 1 and the starting bandwidth
      is (excess-bandwidth * 1.25)";
  }

  typedef kpi-result-type {
    type enumeration {
      enum not-available {
        description
          "No KPI (Key Performance Indicator) configured for the test.";
      }
      enum pass {
        description
          "The test results are within the configured KPI.
           The test is therefore considered a pass";
      }
      enum fail {
        description
          "The test results are outside the configured KPI boundary.
           The test is therefore considered a fail.";
      }
    }
    description
     "Provides a pass or fail for the test results compared to
      the selected KPI profile's pass crieteria for the test.";
  }

  typedef throughput-kpi-percent {
    type uint32;
    description
     "Percent of bandwidth that the maximum throughput result
      shouldn't go below for the test to be considered a pass.
      The value is given as an integer but represent a 4 decimal
      point percent. Ex: 0.2000 is reported as 2000.";
  }

  typedef frameloss-kpi-percent {
    type uint32;
    description
     "Percent of the frameloss test staring bandwidth, the frame
      loss shouldn't exceed for the test to be considered a pass.
      The value is given as an integer but represent a 4 decimal
      point percent. Ex: 0.2000 is reported as 2000.";
  }

  typedef throughput-result {
    type uint32;
    description
     "Throughput results in Mbps are sent as unsigned integer
      multiplied by 100 to provide a 2 decimal point accuracy.
      If result is 123.45 Mbps, it is sent as 12345 and should
      be divided by 100 by the application retrieving the
      data.";
  }
```


Copyright© 2022 Ciena® Corporation

```
typedef frameloss-result {
  type uint32;
  description
   "Frame loss results in Percent of frame loss start bandwidth
    are sent as unsigned integer multitplied by 100 to provide a
    2 decimal point accuracy.
    If result is 12.34 %, it is sent as 1234 and should
    be divided by 100 by the application retrieving the
    data.";
}

typedef pcp-bitmap {
  type bits {
    bit pcp0 {
      position 0;
      description "pcp 0.";
    }
    bit pcp1 {
      position 1;
      description "pcp 1.";
    }
    bit pcp2 {
      position 2;
      description
        "pcp 2.";
    }
    bit pcp3 {
      position 3;
      description
        "pcp 3.";
    }
    bit pcp4 {
      position 4;
      description
        "pcp 4.";
      }
    bit pcp5 {
      position 5;
      description
        "pcp 5.";
      }
    bit pcp6 {
      position 6;
      description
        "pcp 6.";
    }
    bit pcp7 {
      position 7;
      description
        "pcp 7.";}
  }
  description
   "Bitmap of the VLAN PCP (Priority Code) value to test
    with. When The RFC2544 test is selected in the profile,
    only 1 PCP bit can be set. For other tests, a test
    session will be run simultaneously for each PCP set
    in the bitmap. In the latter case, the bandwidth will
    be distributed among the set of PCPs according to the
    selected PCP Bandwidth allocation profile. If no
```


Copyright© 2022 Ciena® Corporation

```
      such profile is configured, the bandwidth will be
      evenly distributed among all set PCPs.";
  }

  typedef duration {
    type enumeration {
      enum t15min {
        description
          "15 minutes.";
      }
      enum t1hr {
        description
          "1 hour.";
      }
      enum t6hr {
        description
          "6 hours.";
      }
      enum t24hr {
        description
          "24 hours.";
      }
      enum tIndefinite {
        description
          "The test instance will run the configured tests
           indefinitely, or until the user sends a test
           instance stop RPC";
      }
      enum tOnce {
        description
          "Once.";
      }
      enum tCustom {
        description
          "Custom duration";
      }
    }
    description
     "Duration of the test. The test group is repeated until
      the duration ends.";
  }

  typedef interval {
    type enumeration {
      enum t15min {
        description
          "15 minutes.";
      }
      enum t1hr {
        description
          "1 hour.";
      }
      enum t6hr {
        description
          "6 hours.";
      }
      enum tCompletion {
        description
          "Interval between the start of one test group and the start
           of the nexxt test group configured on the profile (latency,
```


Copyright© 2022 Ciena® Corporation

```
            pdv, throughput, frameloss, rfc2544 and/or y1564).";
        }
        enum t2hr {
          description
            "2 hours.";
        }
      }
      description
       "Interval between the start of one test group to the start
        of the next test group.";
    }

    typedef measurement-direction {
      type enumeration {
        enum one-way-forward {
          description
        "The measurement provides uni-directional mesurement from the generator
to reflector.";
        }
        enum one-way-backward {
          description
        "The measurement provides uni-directional mesurement from the reflector
to generator.";
        }
        enum one-way-forward-and-backward {
          description
           "The measurement provides two uni-directional mesurement,
           one from the generator to reflector, and the other from the reflector
to generator.";
        }
        enum two-way {
          description
          "The measurement provides bi-directional mesurement for generator to
reflector and back.";
        }
      }
      description
      "This indicates the the type of measurement taken for latency and PDV
tests.";
    }

    typedef time-stamp-type-enum {
      type enumeration {
        enum NTP {
          description
          "This is the NTP type time-stamp.";
        }
        enum 1588 {
          description
           "This is the 1588 type time-stamp.";
        }
      }
      description
       "Type of timestamp used by the SAT engine.";
    }

    typedef rate-fixed-2-decimal {
      type uint32;
      description
```


Copyright© 2022 Ciena® Corporation

```
     "Rate in Mbps sent as unsigned integer multiplied by 100 to provide a 2
decimal
     point accuracy. If result is 123.45 Mbps, it is sent as 12345 and should be
       divided by 100 by the application retrieving the data.";
  }

  /*
   * Groupings
   */

  grouping kpi-values {
    description
      "Common fields for pcp KPIs and untagged KPIs";

    // may be extended to have entry per color
    // in future versions, but for 6.16.1 we do not support this

    leaf throughput {
      type ciena-sat:throughput-kpi-percent;
      description
        "Throughput KPI for the given KPI profile, PCP and color
         expressed in percent of bandwidth * 10000 to provide a 4
         decimal point value.";
    }

    leaf frameloss {
      type ciena-sat:frameloss-kpi-percent;
      description
        "Frameloss KPI for the given KPI profile, PCP and color
         expressed in percent of bandwidth * 10000 to provide a 4
         decimal point value.";
    }

    leaf latency {
      type uint32;
      units "milliseconds";
      description
        "Maximum latency in milliseconds that the latency result
         should not exceed in order for the test to be considered
         a pass.";
      }

    leaf pdv {
      type uint32;
      units "milliseconds";
      description
        "Maximum PDV in milliseconds that the PDV test result
         should not exceed in order for the test to be
         considered a pass.";
    }

    leaf single-outage-max {
      type uint32 {
        range "1..10000";
      }
      units "milliseconds";
      description
        "Maximum time in milliseconds that any single
         outage should not exceed in order for the test
         to be considered a pass.";
```


Copyright© 2022 Ciena® Corporation

```
    }

    leaf sum-outages-max {
      type uint32 {
        range "1..30000";
      }
      units "milliseconds";
      description
        "Maximum time in milliseconds that the sum of recorded
         outages should not exceed in order for the test
         to be considered a pass.";
    }

    leaf max-num-outages {
      type uint32 {
        range "0..128";
      }
      description
        "Maximum number of outages that can occur during
         the outage test in order for the test to be
         considered a pass.";
    }
  }

  grouping notification-base-fields {
    description
      "Common fields for all SAT notifications.";

    leaf severity {
      type ciena:event-severity;
      description
        "Severity.";
    }
    leaf mac-address {
      type yang:mac-address;
      description
        "Mac address.";
    }
    leaf test-instance-name {
      type name-string;
      mandatory true;
      description
        "Name of the test instance.";
    }
  }

  grouping notification-test-kpi-fields {
    description
      "Common fields for all SAT test KPI failure notifications";

    uses notification-base-fields;

    leaf kpi-profile {
      type name-string;
      description
        "KPI profile name.";
    }
    leaf kpi-pcp {
      type priority;
      description
```


Copyright© 2022 Ciena® Corporation

```
        "PCP - when not untagged.";
    }
    leaf kpi-untagged {
      type boolean;
      description
        "Untagged.";
    }
    leaf kpi-color {
      type basic-color;
      description
        "Color.";
    }
    leaf current-pkt-size {
      type uint32;
      description
        "current packet size.";
    }
    leaf emix-sequence {
      type name-string;
      description
        "emix sequence name.";
    }
  }

  grouping common-results-fields {
    description
     "Common fields reported in the results for all tests";

    leaf frame-size {
      type uint32;
      description
       "Packet size in bytes. If 0, then the emix-sequence indicates
        the name of the EMIX sequence used for the test";
    }

    leaf emix-sequence {
      type leafref {
        path "/ciena-sat:sat/ciena-sat:emix-sequence/ciena-sat:name";
      }
      description
       "ID of the EMIX sequence that was used for running the
        test which produced these results. When 0,
        frame size indicates the size of the test frame
        associated with these results.";
    }

    leaf avg-emix-frame-size {
      type uint32;
      description
       "Average frame size for the EMIX sequence for this test.";
    }

    leaf u-frame-size {
      type uint32;
      description
       "The frame size associated with the u letter
        in the given sequence. The default value is
        594.";
    }
```


Copyright© 2022 Ciena® Corporation

```
    leaf h-frame-size {
      type uint32;
      description
       "The frame size associated with the h letter
        in the given sequence.";
    }

    leaf duration {
      type ciena-sat:duration;
      description
       "The duration used for the current test";
    }

    leaf interval {
      type ciena-sat:interval;
      description
       "The interval used for the test";
    }
  }

  /*
   * Config data
   */

  container sat {
    description
      "SAT configuration data.";

    container sat-global {
      description
        "SAT global configuration data.";

      leaf admin-state {
        type ciena:admin-state;
        default disabled;
        description
         "Overall administrative state of the benchmark feature.";
      }

      leaf default-emix-sequence {
        type leafref {
          path "/ciena-sat:sat/ciena-sat:emix-sequence/ciena-sat:name";
        }
        description
         "Default EMIX sequence to be used on a test instance
          when no EMIX is specified in the test instance
          targeted test profile. This applies to generator
          test instances only.";
      }

      leaf generator-default-kpi-profile {
        type leafref {
          path "/ciena-sat:sat/ciena-sat:kpi-profile/ciena-sat:name";
        }
        description
         "Default KPI profile to be used to analyze the results
          of a test instance when no KPI profile is specifed in
          the targeted test profile. This applies to generator
          test instances only.";
      }
```


Copyright© 2022 Ciena® Corporation

```
    leaf generator-default-bw-alloc-profile {
      type leafref {
        path "/ciena-sat:sat/ciena-sat:bw-alloc-profile/ciena-sat:name";
      }
      description
       "Default bandwidth allocation profile to be used with
        a test instance when no bandwidth allocation profile is
        specified in the targeted test profile. This applies to
        generator test instances only.";
    }
  }

  /* Entity table */

  list entity {
    key "name";
    description
     "Service activation testing entity configuration.";

    leaf name {
      type name-string;
      description
       "Name of the service activation test entity.";
    }

    leaf port {
      type mef-logical-port:logical-port-ref;
      description
       "SAT entity's port under test";
    }

    leaf role {
      type enumeration {
        enum reflector {
          description
            "SAT entity's mode of operation is reflector.";
        }
        enum generator {
          description
            "SAT entity's mode of operation is generator.";
        }
      }
      description
       "SAT entity's mode of operation";
    }

    leaf generator-h-frame-size {
       when "../role='generator'";
      type uint32;
      description
       "Maximum frame size that should be generated by the generator
       device. This H frame size should be set to leave enough room for
       extra encapsulation bytes that will be pushed onto the
       packet by the switch when pushing this test packet out on
       the network side port.
       A value of 0 indicates that the maximum frame size for the
       entity port should be used.
       The generator device will ensure that any frame size list
       or EMIX used during the test is adjusted accordingly.
```


Copyright© 2022 Ciena® Corporation

```
         This field is only valid for a generator SAT entity.";
      }

      leaf assoc-gen-vendor-type {
         when "../role='reflector'";
        type enumeration {
           enum other {
             description
               "The remote generator is NOT a Ciena device.";
           }
           enum ciena {
             description
               "The remote generator is a Ciena device.";
           }
         }
         default ciena;
         description
          "This is an indicator as to what is the vendor of the associated
           generator entity connected to this reflector SAT entity.
           This field is only valid for a reflector SAT entity.";
      }

      leaf reflection-level {
         when "../role='reflector'";
        type enumeration {
           enum none {
             description
               "Not configured.";
           }
           enum l2-only {
             description
               "Only the L2 layer fields need to be swapped on reflected
packets.";
           }
           enum l2-to-l3-ipv4-only {
             description
               "The L2 MAC addresses as well as the IPv4 addresses need to
                be swapped on reflected packets.";
           }
           enum l2-to-l3-ipv6-only {
             description
               "The L2 MAC addresses as well as the IPv6 addresses need to
                be swapped on reflected packets.";
           }
           enum l2-to-l4-ipv4-only {
             description
               "L2 MAC addresses, IPv4 addresses and UDP ports need to
                be swapped on the reflected packets.";
           }
           enum l2-to-l4-ipv6-only {
             description
               "L2 MAC addresses, IPv6 addresses and UDP ports need to
                be swapped on the reflected packets.";
           }
           enum l2-to-l4 {
             description
               "L2 MAC addresses, IPv4 or IPv6 addresses and UDP ports need
                to be swapped on the reflected packets.";
           }
         }
```


Copyright© 2022 Ciena® Corporation

```
    description
     "For reflector entities, this indicates the minimum reflection
      level required. If test traffic is going over an ethernet L2
      cloud, l2-only is fine. But if the test traffic is going over
      an IP network, the reflector needs to be able to swap L2 and
      L3 fields. Depending on the type of IP network, the user
      needs to configure IPv4 or IPv6. L2 to L4 will swap MAC
      addresses, IP addresses and src and destination L4 port.
      Depending on the reflection level required, a different device
      is used to implement the reflector.
      This field is only valid for an 8.x reflector SAT entity.
      The default value is l2-to-l4.";
}

leaf reflector-mac-validation {
   when "../role='reflector'";
  type boolean;
  default true;
  description
    "Used to indicate whether the reflector should classify
     classify on the mac or not.";
}

leaf mode {
  type enumeration {
    enum in-service {
      description
        "In service mode; customer traffic is not blocked on the port
         under test while service activation testing is in progress.";
    }
    enum out-of-service {
      description
       "Out of service mode; all customer traffic is dropped at the port
          under test while service activation testing is in progress.";
    }
    enum vid-out-of-service {
      description
        "Vlan ID Out Of Servcie mode; only customer traffic for the
         Vlan IDs currently being tested is being dropped at the port
         under test; all other customer traffic goes through.";
    }
    enum evc-out-of-service { // not supported by 6.x
      description
        "Virtual Circuit Out Of Service; all customer traffic
         associated with the virtual circuit(s) currently under test
         is being dropped at the port under test;
         all other customer traffic goes through.";
    }
  }
  default in-service;
  description
   "SAT entity's mode of operation";
}

leaf admin-state {
  type ciena:admin-state;
  default disabled;
  description
   "SAT entity's administrative state.";
}
```


Copyright© 2022 Ciena® Corporation

```
}

/* Emix sequence table */

list emix-sequence {
  key "name";
  description
   "EMIX sequence entry.";

  leaf name {
    type name-string;
    description
     "Name of this EMIX sequence";
  }

  leaf sequence {
    type string {
      length "1..16";
    }
    description
     "EMIX sequence which is a sequence of up to
      16 letters from the following set, that can
      be repeated, where each letter is associated with
      the frame size below it:
           a    b    c    d    e    f    g    h    u     v   w   x
          64   128  256  512  1024 1280 1518 MTU custom 84  68  88
      The u frame size is user configurable via the
      emix-sequence-UFrameSize attribute
      while the h frame size is the maximum frame size
      configured at the entity level. If the h frame size is
      not configured at the entity level, h takes the value of
      the maximum frame size of the port under test.
      Letter v is the minimum frame size for IPv6 untagged or dot1q
      test frames, w is the minimum for IPv4 qinq test frames, and
      x is the minimum for IPv6 qinq test frames.
      The EMIX character set is available via the
      emix-sequence-character-set list.";
  }

  leaf u-frame-size {
    type uint32 {
      range "64..16000";
    }
    default 594;
    description
     "The frame size associated with the u letter
      in the given sequence. The default value is
      594.";
  }
}

/* kpi-profile table */

list kpi-profile {
  key "name";
  description
   "Key performance indicator pass/fail criteria.";

  leaf name {
    type name-string;
```


Copyright© 2022 Ciena® Corporation

```
      description
       "Name of the KPI profile";
    }

    list pcp {
      key "pcp-value";

      description
        "Each PCP within the KPI profile has its own configuration.";

      leaf pcp-value {
        type priority;
        description
         "PCP value associated with this entry";
      }

      uses kpi-values;
    }

    container untagged {
      description
       "Untagged traffic does not have a PCP, so it is handled separately.";

      uses kpi-values;
    }
  }

  /* bandwidth allocation profile table */

  list bw-alloc-profile {
    key "name";

    description
     "BW allocation profile entry";

    leaf name {
      type name-string;
      description
       "Name of the bandwidth allocation profile";
    }

    list pcp {
      key "pcp-value";

      description
        "Each PCP within the bw-alloc-profile has its own configuration.";

      leaf pcp-value {
        type priority;
        description
         "PCP value associated with this entry.";
      }

      leaf ratio {
        type uint32 {
          range "0..1000000";
        }
        description
         "Ratio of test bandwidth to allocate to the
          test session with this PCP value.";
```


Copyright© 2022 Ciena® Corporation

```
      }
    }
  }

  /* test profile table */

  list test-profile {
    key "name";
    description
     "Service activation test profile configuation information.";

    leaf name {
      type name-string;
      description
       "Name of the profile.";
    }

    leaf bandwidth {
      type uint32;
      description
       "Maximum bandwidth to use when generating green test
        traffic for the profile.";
    }

    leaf excess-bandwidth {
      type uint32;
      default 0;
      description
       "Maximum bandwidth to use when generating yellow test
        traffic for the profile. When testing for red color,
        this bandwidth is multiplied by 1.25";
    }

    leaf interval {
      type ciena-sat:interval;
      default tCompletion;
      description
       "Interval between the start of one test group to the start
        of the next test group.";
    }

    leaf duration {
      type ciena-sat:duration;
      default tOnce;
      description
       "Duration of the testing. The test group is repeated until
        the duration ends. When indefinite is set, test group is
        repeated until the user manually stops the test.";
    }

    leaf custom-duration {
      type uint32 {
        range 15..65535;
      }
      units "minutes";
      default 15;
      description
       "Custom duration of testing in minutes. This leaf is only used
        when test-profile duration leaf is set to tCustom.";
    }
```


Copyright© 2022 Ciena® Corporation

```
leaf-list frame-size {
  type uint32 {
    range 64..16000;
  }
  max-elements 10;
  description
 "Used to replace the existing frame size table with the specified list.
   List of frame sizes the test will run with. The selected tests are
    performed for one frames size, then the same test is repeated with
    the next frame size in the list.  Results are provided for each
    separate frame size from the list.  When rfc2544 is selected, the
    default frame size is made up of the following 7 frame sizes:
    64,128,256,512,1024,1280,1518; The first frame size may differ
    from 64 depending on the VLAN encapsulation type configured and/or
    the IP version used for the test:
    Untagged, non IP or IPv4          64
    Untagged IPv6                     84
    dot1q non IP or IPv4              64
    dot1q IPv6                        84
    QinQ non IP or IPv4               68
    QinQ IPv6                         88.";
}

leaf max-searches {
  type uint8 {
    range "1..16";
  }
  default 7;
  description
   "The throughput test uses a binary search to find the maximum
    rate at which test traffic can be sent without any packet drops.
    This attributes dictates the maximum number of searches to perform
    in the binary search algorithm, before stopping the test.";
}

leaf max-samples {
  type uint8 {
    range "2..20";
  }
  default 10;
  description
   "For the latency and PDV tests, this is the number of
    samples to take in order to determine the min, avg
    and max latency and avg PDV.";
}

leaf sampling-interval {
  type uint16 {
    range "1..600";
  }
  default 1;
  description
   "This is the number of 100ms steps to use when running any of the
    configured tests. A value of 1 represents 100ms, a value of 600
    makes a sampling interval of 1 minutes (600 * 100ms). For the
   throughput and frame loss tests, this is the number of milliseconds
    to generate test traffic for before stopping and checking for
    frame drops. For latency and PDV, it represents the number of
    milliseconds between samples.";
```


Copyright© 2022 Ciena® Corporation

```
      }

      leaf frameloss-start-bw {
        type enumeration {
          enum profile-bandwidth {
            description
              "Bandwidth specified in profile.";
          }
          enum maximum-throughput {
            description
              "Maximum throughput.";
          }
        }
        default maximum-throughput;
        description
         "For the frame loss test, this indicates which bandwidth
          value to use as the starting bandwidth; the maximum
          bandwidth determined via the throughput test or the
          bandwidth parameter configured in the profile. Note
          that when rfc2544 is enabled, this value is ignored and
          the maximum bandwidth of the link is used (minimum of the
          port under test speed and entity/FPGA link speed).";
      }

      leaf frameloss-decrement {
        type uint32 {
          range "1..100";
        }
        default 10;
        description
        "For the frame loss test, this indicates the value by which the
        frameloss-start-bw is decremented by, when a frameloss is recorded
        in an iteration of the frameloss test until no frameloss is
        detected";
      }

      leaf frameloss-decrement-type {
        type enumeration {
          enum percent {
            description
              "For the frame loss test, this indiacted that the frameloss-
start-bw
               for the frameloss test is decrement the by percentage of start
bandwidth for the
                frameloss test. This value is fixed at 10%";
          }
          enum Mbps {
            description
              "For the frame loss test, this indiactes that the frameloss-
start-bw
               for the frameloss test is decrement by the user-specified value
                (in Mbps) specified in the frameloss-decrement field.";
          }
        }
        default percent;
        description
        "For the frame loss test, this field defines whether to decrement the
         frameloss test bandwidth by percent (fixed at 10%) or by the user
         specified value given in the frameloss-decrement field";
      }
```


Copyright© 2022 Ciena® Corporation

```
      leaf vid-validation {
        type boolean;
        default true;
        description
         "Indicates whether the VID of the reflected test frame should
          be checked for a match when it arrives back at the generator.
          If VID validation is set to false, PCP and Color validation
          are ignored and automatically become false.";
      }

      leaf pcp-validation {
        type boolean;
        default true;
        description
         "Used to indicate whether the PCP value of the
          test packet should be used to classifier the reflected
          packets or not. When color remarking is used, this should
          be set to false. Default is true.";
      }

      leaf color-validation {
        type boolean;
        default true;
        description
         "Used to indicate whether the color bit of the
          test packet should be used to classifier the reflected
          packets or not. In color unaware testing, this should
          be set to false. Default is true.";
      }

      leaf source-mac-validation {
        type boolean;
        default true;
        description
        "Used to indicate whether the source MAC address of the reflected
         frames is used to classify the test frame against the correct test
         instance/session.Default is true.";
      }

      leaf inner-vid-validation {
        type boolean;
        default true;
        description
       "Used to indicate whether the inner VLAN is used to classify reflected
traffic.
        This is only valid when the encapsulation is set to QinQ. Default is
true";
      }

      leaf measurement-direction {
        type ciena-sat:measurement-direction;
        default two-way;
        description
        "This indicates the the type of measurement taken for latency and PDV
tests.";
      }


      leaf kpi-profile-name {
```


Copyright© 2022 Ciena® Corporation

```
      type leafref {
        path "/ciena-sat:sat/ciena-sat:kpi-profile/ciena-sat:name";
      }
      description
       "Name of a valid KPI profile, from the
        kpi-profile list, to use when
        analysing the test results for this profile.";
    }

    leaf bw-alloc-profile {
      type leafref {
        path "/ciena-sat:sat/ciena-sat:bw-alloc-profile/ciena-sat:name";
      }
      description
       "Name of a valid Bandwidth Allocation profile,
        from the bw-alloc-profile list,
        used for distributing the profile bandwidth amongst
        the configured PCP values for this profile when running
        tests.";
    }

    leaf emix-sequence {
      type leafref {
        path "/ciena-sat:sat/ciena-sat:emix-sequence/ciena-sat:name";
      }
      description
       "Name of a valid EMIX sequence configured in the
        emix-sequence list, to use when y1564
        is selected. The EMIX sequence is used instead of the
        frame size list if and only if y1564 is selected.";
    }

    leaf encap-type {
      type enumeration {
        enum untagged {
          description
            "Untagged.";
        }
        enum dot1Q {
          description
            "Dot1q.";
        }
        enum qinq {
          description
            "QinQ.";
        }
      }
      default untagged;
      description
       "Encapsulation type to use for generated packets.";
    }

    leaf pdu-type {
      type enumeration {
        enum ethernet {
          description
            "Ethernet.";
        }
        enum ip {
          description
```


Copyright© 2022 Ciena® Corporation

```
          "IP.";
      }
      enum udp-echo {
        description
          "UDP echo.";
      }
    }
    default ethernet;
    description
     "Test PDU type.";
  }

  leaf dst-mac {
    type yang:mac-address;
    description
     "MAC address to use as the destination MAC address on
      the generated packets.";
  }

  leaf s-vid { // Not supported in 6.x
    type ieee:vlanid;
    description
   "S-VLAN ID to used for generated packets when encapType is set to QinQ";
  }

  leaf s-pcp { // Not supported in 6.x
    type ciena-sat:pcp-bitmap;
    description
     "Bitmap of the S-VLAN PCP (Priority Code Point) value to
      test with. When RFC2544 test is enabled, only one
      PCP can be set in the mask. Otherwise, a test session
      will be run for each PCP bit set. Used when the encapType
      set to QinQ";
  }

  leaf s-color {  // Not supported in 6.x
    type ciena-sat:color;
    description
     "Color of the S-VLAN tag for the test traffic. This is
      directly related to the value of the DEI bit
      configured in the VLAN tag of the test traffic. Green
      traffic will have its DEI bit set to 0 while Yellow
      traffic will have its DEI bit set to 1.
      Green test traffic is generated based on the bandwidth
      configured in the test profile. Yellow traffic is
      generated based on the excess-bandwidth parameter.
      Finally, if the user selects green-yellow-red, the
      yellow traffic is generated at 1.25 * excess-bandwidth.
      This is used only when encapType is set to QinQ";
  }

  leaf c-vid {
    type ieee:vlanid;
    description
     "C-VLAN ID to used for generated packets when encapType is set to
      QinQ or dot1q.";
  }

  leaf c-pcp {
    type ciena-sat:pcp-bitmap;
```


Copyright© 2022 Ciena® Corporation

```
      default pcp0;
      description
       "Bitmap of the C-VLAN PCP (Priority Code Point) value to
        test with. When RFC2544 test is enabled, only one
        PCP can be set in the mask. Otherwise, a test session
        will be run for each PCP bit set. Only used when the
        encapType is set to dot1q. In the QinQ type, the c-pcp
        of the test packet takes the same value as the s-pcp to
        simplify software.";
    }

    leaf c-color {
      type ciena-sat:color;
      default green;
      description
       "Color of the C-VLAN tag for the test traffic. This is
        directly related to the value of the DEI bit
        configured in the VLAN tag of the test traffic. Green
        traffic will have its DEI bit set to 0 while Yellow
        traffic will have its DEI bit set to 1.
        Green test traffic is generated based on the bandwidth
        configured in the test profile. Yellow traffic is
        generated based on the excess-bandwidth parameter.
        Finally, if the user selects green-yellow-red, the
        yellow traffic is generated at 1.25 * excess-bandwidth.
        This is used only when encapType is set to dot1q.
        In a QinQ configuration, the c-dei bit is set to the
        same value as the s-dei bit in order to simplify
        software.";
    }

    leaf tpid {
      type enumeration {
        enum 0x8100 {
          description
            "VLAN tagged frame.";
        }
        enum 0x88a8 {
          description
            "Provider bridging.";
        }
        enum 0x9100 {
          description
            "Proprietary.";
        }
      }
      default 0x8100;
      description
       "Vlan tag protocol identifier to use for the generated packets.";
    }

    leaf dscp {
      type inet:dscp;
      description
       "IP DSCP value to use for the generated packets.";
    }

    leaf src-addr {
      type inet:ip-address;
      description
```


Copyright© 2022 Ciena® Corporation

```
   "Specifies the unicast source IP address to use in IP test packet.";
}

leaf dst-addr {
  type inet:ip-address;
  description
  "Specifies the unicast destination IP addr to use in IP test packet.";
}

leaf custom-payload {
  type binary {
    length "0..256";
  }
  description
   "Custom payload to be used in the generated packets.";
}

leaf throughput-test {
  type boolean;
  default false;
  description
   "Run throughput test";
}

leaf frameloss-test {
  type boolean;
  default false;
  description
   "Run frame loss test";
}

leaf latency-test {
  type boolean;
  default false;
  description
   "Run latency test";
}

leaf pdv-test {
  type boolean;
  default false;
  description
   "Run packet delay variation test";
}

leaf burst-test { // Not supported yet
  type boolean;
  description
   "Run packet burst test";
}

leaf rfc2544-suite {
  type boolean;
  default false;
  description
   "Run RFC2544 test suite which includes throughput, frame
    loss and latency tests";
}

leaf y1564-suite {
```


Copyright© 2022 Ciena® Corporation

```
        type boolean;
        default false;
        description
         "The Y1564 test includes the throughput, frameloss,
          latency and PDV tests and requires an EMIX sequence to be
          configured. If the profile's EMIX sequencei s not configured,
          the default generator EMIX sequence is used.
          Y1564 and RFC2544 are mutually exclusive.";
      }

      leaf outage-test {
        type boolean;
        default false;
        description
          "Run outage test";
      }
    }

    /* test instance table */

    list test-instance {
      key "name";
      description
       "Service activation test instance configuration data.";

      leaf name {
        type name-string;
        description
         "Unique name given to the test instance.";
      }

      leaf flow-point-id { // Not supported in 6.x
        type mef-fp:fp-ref;
        description
       "flow-point or sub-port to base this test instance on. The flow-point
has
          to be a child of the logical port under test associated with an
existing
          entity.";
      }

      leaf profile {
        type leafref {
          path "/ciena-sat:sat/ciena-sat:test-profile/ciena-sat:name";
        }
        description
        "Name of an existing test profile to be used with the test instance as
          the template for testing. A test profile ID has to be specified if
         this test instance is associated with a generator entity. If it's a
          reflector test instance, a profile ID CANNOT be specified, it has
          to remain unset.";
      }

      leaf s-vid { // Not supported in 6.x
        type uint16;
        description
         "S VLAN ID to use for testing.
         For generator test instances, this over-writes the value defined in
          the referenced test profile if applicable (encap-type set to qinq).
          For reflector test instances, this has to be set to a valid s-vid
```


Copyright© 2022 Ciena® Corporation

```
          accepted by the associated sub-port. A value of 0xFFFF indicates
          ANY s-vid.";
      }

      leaf c-vid {
        type uint16;
        description
         "C VLAN ID to use for testing.
          For generator test instances, this over-writes the value defined
          in the referenced test profile if applicable (encap-type set to
          dot1q or qinq).
          For reflector test instances, this has to be set to a valid c-vid
          accepted by the associated port/sub-port.
          A value of 0xFFFF indicates ANY c-vid.";
      }

      leaf untagged {
        type boolean;
        default false;
        description
         "When set to true, this indicates that the test instance
          is to be associated with untagged data traffic. When this
          is set, the cvid and svid parameters must be 0.";
      }

      leaf dst-mac {
        type yang:mac-address;
        description
       "Destination MAC address to use for generator test instances. When set,
          this over writes the value specified in the referenced test profile.
           This field is not applicable for reflector test instances.";
      }

      leaf admin-state {
        type ciena:admin-state;
        default disabled;
        description
         "Indicates the administrative state of the test instance.
         Upon enabling a test instance, resources are allocated in hardware.
         For reflector test instances, this triggers the start of reflection
          of any test traffic received that matches the specified vid
          combination on the specified port/sub-port.";
      }
    }
  }


  /*
   * Operational state data nodes
   */

  container sat-state {
    config false;

    description
      "Operational data for SAT.";

    container sat-global {
      description
        "SAT global operational data.";
```


Copyright© 2022 Ciena® Corporation

```
leaf platform-max-config-entities {
  type uint16;
  description
   "Maximum number of entities that can be created on
    this platform.";
}

leaf platform-max-config-test-instances {
  type uint16;
  description
   "Maximum number of test instances that can be created on
    this platform.";
}

leaf platform-max-config-test-profiles {
  type uint16;
  description
   "Maximum number of test profiles that can be created on
    this platform.";
}

leaf platform-max-config-kpi-profiles {
  type uint16;
  description
   "Maximum number of KPI profiles that can be created on
    this platform.";
}

leaf platform-max-config-bw-alloc-profiles {
  type uint16;
  description
   "Maximum number of bandwidth distribution profiles
    that can be created on this platform.";
}

leaf platform-max-config-emix-sequences {
  type uint16;
  description
   "Maximum number of EMIX sequences that can be created on
    this platform, including the system created ones.";
}

leaf platform-max-simultaneous-running-tests {
  type uint16;
  description
   "Maximum number of test instances that can be running
    simultaneously on this platform.";
}

leaf configured-entities {
  type uint16;
  description
   "Number of entities currently configured.";
}

leaf configured-test-instances {
  type uint16;
  description
   "Number of test instances currently configured.";
```


Copyright© 2022 Ciena® Corporation

```
    }

    leaf configured-profiles {
      type uint16;
      description
       "Number of test profiles currently configured.";
    }

    leaf configured-emix-sequences {
      type uint16;
      description
       "Number of EMIX sequences currently configured, including
        the system created ones.";
    }

    leaf configured-kpi-profiles {
      type uint16;
      description
       "Number of KPI profiles currently configured.";
    }

    leaf configured-bw-alloc-profiles {
      type uint16;
      description
       "Number of bandwidth distribution profiles currently
        configured.";
    }

    leaf enabled-test-instances {
      type uint16;
      description
       "Number of test instances currently enabled across
        the platform.";
    }

    leaf generator-running-test-instances {
      type uint16;
      description
       "Number of generator test instances currently running
        across the platform.";
    }
  }

  list entity {
    key "name";

    description
      "Operational data for entity.";

    leaf name {
      type name-string;
      description
       "Name of the service activation test entity.";
    }

    leaf id {
      type uint32;
      description
       "Entity unique ID assigned on creation by the system.";
    }
```


Copyright© 2022 Ciena® Corporation

```
      leaf oper-state {
        type ciena:oper-state;
        description
         "Entity operational enable/disable. An entity
          is operationally enabled when the feature is
          enabled, the entity is enabled and at least one
          test instance associated with that entity is
          enabled (reflector) or running (generator)";
      }

      leaf slot-id {
        type uint32;
        description
         "The slot Id of the line module associated with this entity.
          For non-distributed platforms or single card nodes, this is
          always 1.";
      }

      leaf local-mac {
        type yang:mac-address;
        description
         "Mac address assigned to the SAT entity.";
      }

      leaf gen-max-emix-length {
        type uint8;
        description
         "This is the maximum emix-sequence length that can be configured.
          for an EMIX sequence that the entity supports.";
      }

      leaf gen-seqnum-field-length {
        type uint8;
        description
         "This is the length of the sequence number field in the generated
packets";
      }

      leaf engine-bandwidth {
        type uint32;
        description
        "This is the maximum bandwidth supported by the SAT engine/entity in
Mbps.";
      }

      leaf time-stamp-type {
        type ciena-sat:time-stamp-type-enum;
        description
         "Type of timestamp used by the SAT engine.";
      }

      leaf max-config-test-instances {
        type uint16;
        description
         "Maximum number of test instances that can
          be configured on this entity.";
      }

      leaf max-simultaneous-test-instances {
```


Copyright© 2022 Ciena® Corporation

```
        type uint16;
        description
         "Maximum number of test instances that can
          be enabled (reflector) or running (generator)
          on this entity, given the availability of
          hw sessions.";
      }

      leaf num-test-instances-configured {
        type uint16;
        description
         "Number of test instances currently configured
          on this SAT entity.";
      }

      leaf num-test-instances-enabled {
        type uint16;
        description
         "Number of test instances currently enabled on this SAT entity.";
      }

      leaf num-test-instances-reflecting {
        type uint16;
        description
         "Number of test instances currently reflecting on this reflector
entity.
          This information is only valid for reflector entities.";
      }

      leaf gen-num-test-instances-running {
        type uint16;
        description
       "Number of test instances currently running on this generator entity.
          This information is only valid for generator entities.";
      }

      leaf bw-available {
        type uint32;
        description
         "Bandwidth in Mbps, available for test traffic if a reflector.
          Bandwidth remaining for other test instances to be started
          if this is a generator SAT entity.";
      }

      leaf gen-bw-used-by-running-tests {
        type uint32;
        description
         "Bandwidth used by test instances
          currently running on this generator entity, in Mbps.";
      }

      leaf available-hw-sessions {
        type uint32;
        description
         "Maximum number of simultaneous hardware sessions available
          on this SAT entity.";
      }

      leaf allocated-hw-sessions {
        type uint32;
```


Copyright© 2022 Ciena® Corporation

```
        description
       "Number of hw sessions currently allocated to enabled test instances.";
      }

      container statistics {
        description
          "A collection of entity-related statistics objects.";

        leaf tx-bytes {
          type yang:counter64;
          description
           "Number of bytes transmitted by the benchmark port";
        }

        leaf tx-pkts {
          type yang:counter64;
          description
           "Number of packets transmitted by the benchmark port";
        }

        leaf rx-bytes {
          type yang:counter64;
          description
           "Number of bytes received on the benchmark port.";
        }

        leaf rx-pkts {
          type yang:counter64;
          description
           "Number of packets received on the benchmark port.";
        }

        leaf miss-class-pkts {
          type yang:counter64;
          description
           "Test packets received that didn't match any enabled
            test session.";
        }

        leaf crc-err-pkts {
          type yang:counter64;
          description
           "Test packets received with CRC errors";
        }

        leaf udp-chksum-err-pkts {
          type yang:counter64;
          description
           "Test packets received with UDP checksum errors";
        }
      }
    }

    list emix-sequence {
      key "name";

      description
        "Emix sequence.";

      leaf name {
```


Copyright© 2022 Ciena® Corporation

```
      type name-string;
      description
       "Name of this EMIX sequence";
    }

    leaf id {
      type uint8 {
        range "1..32";
      }
      description
       "Unique Id of the EMIX sequence entry";
    }

    leaf num-of-ref {
      type uint16;
      description
       "Number of generator test profiles currently
        using this EMIX sequence for testing.
        If this is the default EMIX sequence
        set in the generator, that also counts
        as one reference.";
    }

    leaf user-created {
      type boolean;
      description
       "Flag indicating whether this sequence was
        created by the user or by the system at
        initialization. Only user created sequences
        are editable.";
    }
  }

  list gen-test-session-allocation {

    key "entity-name
         test-instance-name
         untagged
         pcp
         color";
    description
     "Test session (aka hardware session) allocation entry.
      If untagged is True, pcp and color are irrelevant.";

    leaf entity-name {
      type leafref {
        path "/ciena-sat:sat/ciena-sat:entity/ciena-sat:name";
      }
      description
       "Name of entity.";
    }
    leaf test-instance-name {
      type leafref {
        path "/ciena-sat:sat/ciena-sat:test-instance/ciena-sat:name";
      }
      description
       "Name of test instance.";
    }

    leaf untagged {
```


Copyright© 2022 Ciena® Corporation

```
    type boolean;
    description
      "Untagged.";
  }

  leaf pcp {
    type priority;
    description
     "PCP value.";
  }

  leaf color {
    type basic-color;
    description
     "Color - green or yellow only.";
  }

  leaf session-id {
    type uint32 {
      range "0..127";
    }
    description
     "Id of the hardware session allocated to the given
      test instance, pcp and color for this entity.";
  }

  leaf emix-sequence-name {
    type leafref {
      path "/ciena-sat:sat/ciena-sat:emix-sequence/ciena-sat:name";
    }
    description
     "EMIX sequence used with this session. If not set,
      this indicates that the the frame size list is being used
      for the test and current-frame-size
      indicates which packet size is currently being tested.";
  }

  leaf current-frame-size {
    type uint32;
    description
     "Packet size in bytes currently used by the running test.";
  }

  leaf throughput-test-state {
    type ciena-sat:throughput-test-state-type;
    description
     "State of the throughput test for this hardware session
      defined by the test instance, pcp and color";
  }

  leaf throughput-current-bw-search-depth {
    type uint32;
    description
     " This is the depth the throughput test is currently
       at, in its binary search.";
  }

  leaf throughput-max-search-depth {
    type uint32;
    description
```


Copyright© 2022 Ciena® Corporation

```
    "This is the maximum search depth configured for the
     throughput test in the corresponding test-profile.";
}

leaf throughput-searches-without-drops {
  type uint32;
  description
   "This is the number of searches(iterations) of the running
    throughput test where no packet drops were recorded.";
}

leaf frameloss-test-state {
  type ciena-sat:frameloss-test-state-type;
  description
   "State of the frameloss test for this hardware session
    defined by the test instance, pcp and color";
}

leaf frameloss-current-count {
  type uint64;
  description
   "The number of frames that have been lost in this
    current iteration of the frame loss test.";
}

leaf latency-test-state {
  type ciena-sat:latency-pdv-test-state-type;
  description
   "State of the latency test for this hardware session
    defined by the test instance, pcp and color";
}

leaf current-latency {
  type uint32;
  units "micro-seconds";
  description
   "Latency of the last frame sampled in current iteration.";
}

leaf pdv-test-state {
  type ciena-sat:latency-pdv-test-state-type;
  description
   "State of the PDV test for this hardware session
    defined by the test instance, pcp and color";
}

leaf current-pdv {
  type uint32;
  units "micro-seconds";
  description
   "PDV of the last two frames sampled in this current iteration.";
}

leaf rfc2544-test-state {
  type ciena-sat:rfc2544-test-state-type;
  description
   "State of the RFC2544 test for this hardware session
    defined by the test instance, pcp and color";
}
```


Copyright© 2022 Ciena® Corporation

```
    leaf y1564-test-state {
      type ciena-sat:y1564-test-state-type;
      description
       "State of the Y1564 test for this hardware session
        defined by the test instance, pcp and color";
    }

    leaf outage-test-state {
      type ciena-sat:outage-test-state-type;
      description
       "State of the outage test for this hardware session
        defined by the test instance, pcp and color";
    }

    leaf num-outages-so-far {
      type uint32;
      description
       "Outages detected so far";
    }

    leaf outage-progress {
      type uint8;
      description
        "Percentage of outage test completed";
    }

    leaf current-rate {
      type uint32;
      description
       "Traffic rate currently used to generate packets as percent
        of the line rate";
    }

    leaf samples-completed {
      type uint32;
      description
       "When latency and/or pdv is running, this indicates how
        many samples have been gathered so far.";
    }
  }

  list test-instance-statistics {
    key "entity-name
         test-instance-name";

    description
     "Benchmark test instance packet statistics";

    leaf entity-name {
      type leafref {
        path "/ciena-sat:sat/ciena-sat:entity/ciena-sat:name";
      }
      description
       "Entity name.";
    }

    leaf test-instance-name {
      type leafref {
        path "/ciena-sat:sat/ciena-sat:test-instance/ciena-sat:name";
      }
```


Copyright© 2022 Ciena® Corporation

```
  description
   "Test instance name.";
}

leaf rx-pkts {
  type yang:counter64;
  description
   "Total number of received packets for
    this test instance.";
}

leaf rx-ipv4-pkts {
  type yang:counter64;
  description
   "Total number of IPv4 packets received for
    this test instance.";
}

leaf rx-ipv4-udp-pkts {
  type yang:counter64;
  description
   "Total number of IPv4 UDP packets received for
    this test instance.";
}

leaf rx-ipv6-pkts {
  type yang:counter64;
  description
   "Total number of IPv6 packets received for
    this test instance.";
}

leaf rx-ipv6-udp-pkts {
  type yang:counter64;
  description
   "Total number of IPv6 UDP packets received for
    this test instance.";
}

leaf rx-non-ip-pkts {
  type yang:counter64;
  description
   "Total number of non-IP packets received for
    this test instance.";
}

leaf rx-unrecognized-pkts {
  type yang:counter64;
  description
   "Total number of unrecognized packets received for
    this test instance.";
}

leaf rx-duplicate-pkts {
  type yang:counter64;
  description
   "Total number of packets received with a duplicate
    this test instance.";
}
```


Copyright© 2022 Ciena® Corporation

```
    leaf rx-duplicate-pkts-overflow {
      type boolean;
      description
       "When set to true, this indicates that the duplicate
        pkt counter has rolled over in hardware and therefore,
        rxDuplicate-pkts might not be accurate.";
    }

    leaf rx-ooo-pkts {
      type yang:counter64;
      description
       "Total number of packets received with an out of
        order sequence number. The sequence number was less
        than the expected sequence number for this test instance.";
    }

    leaf rx-ooo-pkts-overflow {
      type boolean;
      description
       "When set to true, this indicates that the out of order
        pkt counter has rolled over in hardware and therefore,
        rxOOO-pkts might not be accurate.";
    }

    leaf rx-disc-seq-num-pkts {
      type yang:counter64;
      description
       "Total number of packets received with a sequence number
        greater then the expected sequence number for this test instance.";
    }

    leaf rx-disc-seq-num-pkts-overflow {
      type boolean;
      description
       "When set to true, this indicates that the discontinuity
        in sequence number pkt counter has rolled over in hardware
        and therefore, rxDiscSeqNum-pkts might not be accurate.";
    }

    leaf tx-pkts {
      type yang:counter64;
      description
       "Total number of packets transmitted for this test instance.";
    }
  }

  list gen-test-session-throughput-results {
    key "entity-name
         test-instance-name
         untagged
         pcp
         color
         profile-frame-size-index";

    description
     "generator test session throughput results.
      If untagged is True, pcp and color are irrelevant.";

    leaf entity-name {
      type leafref {
```


Copyright© 2022 Ciena® Corporation

```
      path "/ciena-sat:sat/ciena-sat:entity/ciena-sat:name";
    }
    description
     "Entity name.";
  }

  leaf test-instance-name {
    type leafref {
      path "/ciena-sat:sat/ciena-sat:test-instance/ciena-sat:name";
    }
    description
     "Test instance name.";
  }

  leaf untagged {
    type boolean;
    description
      "Untagged.";
  }

  leaf pcp {
    type priority;
    description
     "PCP value";
  }

  leaf color {
    type basic-color;
    description
     "Color - green or yellow only.";
  }

  leaf profile-frame-size-index {
    type uint8 {
      range "0..9";
    }
    description
     "Frame size index for which those throughput results are
      for. When EMIX is used, in which case emix-sequence is
      not NULL, there is only one frame size index for
      the given test instance, pcp and color. In such a case, the
      frame size index will be 0 and frame size will also be 0.";
  }

  leaf min {
    type ciena-sat:throughput-result;
    description
     "Minimum throughput recorded for the given packet size
      in Mbps * 100. The value is multiplied by 100 to be able
      to send the results as an integer but should be divided
      by 100 when displayed to provide a 2 decimal point accuracy.
      A result of 123.45 is sent as 12345.";
  }

  leaf max {
    type ciena-sat:throughput-result;
    description
     "Maximum throughput recorded for the given packet size
      in Mbps * 100. The value is multiplied by 100 to be able
      to send the results as an integer but should be divided
```


Copyright© 2022 Ciena® Corporation

```
          by 100 when displayed to provide a 2 decimal point accuracy.
          A result of 123.45 is sent as 12345.";
      }

      leaf avg {
        type ciena-sat:throughput-result;
        description
         "Average throughput recorded for the given packet size
          in Mbps * 100. The value is multiplied by 100 to be able
          to send the results as an integer but should be divided
          by 100 when displayed to provide a 2 decimal point accuracy.
          A result of 123.45 is sent as 12345.";
      }

      leaf iterations {
        type uint32;
        description
         "Number of times the test has been run for this packet size.";
      }

      leaf max-bw-search-depth {
        type uint32;
        description
        "The search depth for finding the maximum throughput bandwidth.";
      }

      leaf kpi-result {
        type ciena-sat:kpi-result-type;
        description
         "Pass or fail results for the test which is determined
          by comparing the max throughput against the selected
        KPI test instance's throughput pass criteria. If no KPI test instance
          is selected for the given test instance Id, the result will
          be 'notApplicable'.";
      }

      leaf rate {
        type ciena-sat:rate-fixed-2-decimal;
        description
         "Rate in Mbps.";
      }

      leaf suspect-flag {
        type boolean;
        description
         "with port under test having 1:1 LAG/MC-LAG/Y-protection config and
        while a test was running and port protection swicth event is detected
then this flag shall
          be marked to true indicating results might no be valid as more or
more protection switch event detected
          while a test was running. Default value of flag is false.";
      }

      uses common-results-fields;
    }

    list gen-test-session-frameloss-results {
      key "entity-name
           test-instance-name
           untagged
```


Copyright© 2022 Ciena® Corporation

```
      pcp
      color
      profile-frame-size-index
      rate-index";

  description
   "generator test session frameloss results.
    If untagged is True, pcp and color are irrelevant.";

  leaf entity-name {
    type leafref {
      path "/ciena-sat:sat/ciena-sat:entity/ciena-sat:name";
    }
    description
     "Entity name.";
  }

  leaf test-instance-name {
    type leafref {
      path "/ciena-sat:sat/ciena-sat:test-instance/ciena-sat:name";
    }
    description
     "Test instance name.";
  }

  leaf untagged {
    type boolean;
    description
      "Untagged.";
  }

  leaf pcp {
    type priority;
    description
     "PCP value.";
  }

  leaf color {
    type basic-color;
    description
     "Color - green or yellow only.";
  }

  leaf profile-frame-size-index {
    type uint8 {
      range "0..9";
    }
    description
     "Frame size index.";
  }

  leaf rate-index {
    type uint8 {
      range "0..9";
    }
    description
     "Rate index.";
  }

  leaf rate {
```


Copyright© 2022 Ciena® Corporation

```
        type uint32;
        description
         "Rate in percent of line rate.";
      }

      leaf first {
        type ciena-sat:frameloss-result;
        description
         "Percentage of frames lost on first test sequence.";
      }

      leaf second {
        type ciena-sat:frameloss-result;
        description
         "Percentage of frames lost on second test sequence.";
      }

      leaf kpi-result {
        type ciena-sat:kpi-result-type;
        description
         "Pass or fail results for the test which is determined
          by comparing the frameloss result against the selected
         KPI test instance's frameloss pass criteria. If no KPI test instance
          is selected for the given test instance Id, the result will
          be 'notApplicable'.";
      }

      leaf result {
        type ciena-sat:frameloss-result;
        description
         "Percentage of frames lost on the largest test sequence results.";
      }

      leaf frameloss-count {
        type uint64;
        description
       "This indicates the number of frames lost during the test for a given
frame-size or EMIX";
      }

      leaf suspect-flag {
        type boolean;
        description
         "with port under test having 1:1 LAG/MC-LAG/Y-protection config and
         while a test was running and port protection swicth event is detected
then this flag shall
          be marked to true indicating results might no be valid as more or
more protection switch event detected
          while a test was running. Default value of flag is false.";
      }

      uses common-results-fields;
    }

    list gen-test-session-latency-results {
      key "entity-name
           test-instance-name
           untagged
           pcp
           color
```


Copyright© 2022 Ciena® Corporation

```
      profile-frame-size-index";

  description
   "generator test session latency results.
    If untagged is True, pcp and color are irrelevant.";

  leaf entity-name {
    type leafref {
      path "/ciena-sat:sat/ciena-sat:entity/ciena-sat:name";
    }
    description
     "Entity name.";
  }

  leaf test-instance-name {
    type leafref {
      path "/ciena-sat:sat/ciena-sat:test-instance/ciena-sat:name";
    }
    description
     "Automagically generated leafref leaf.";
  }

  leaf untagged {
    type boolean;
    description
      "Untagged.";
  }

  leaf pcp {
    type priority;
    description
     "PCP value.";
  }

  leaf color {
    type basic-color;
    description
     "Color - green or yellow only.";
  }

  leaf profile-frame-size-index {
    type uint8 {
      range "0..9";
    }
    description
     "Frame size index.";
  }

  leaf min {
    type uint32;
    description
     "Minimum latency recorded for the given packet size.";
  }

  leaf max {
    type uint32;
    description
     "Maximum latency recorded for the given packet size.";
  }
```


Copyright© 2022 Ciena® Corporation

```
      leaf avg {
        type uint32;
        description
         "Average latency recorded for the given packet size.";
      }

      leaf samples {
        type uint32;
        description
         "Number of samples taken during the test.";
      }

      leaf kpi-result {
        type ciena-sat:kpi-result-type;
        description
         "Pass or fail results for the test which is determined
          by comparing the frameloss result against the selected
          KPI test instance's latency pass criteria. If no KPI test instance
          is selected for the given test instance Id, the result will
          be 'notApplicable'.";
      }

      leaf measurement-direction {
        type ciena-sat:measurement-direction;
        description
        "This indicates the direction of latency measurement used for the
latency test.";
      }

      leaf latency-median {
        type uint64;
        units "microseconds";
        description
        "The latency median";
      }

      leaf rate {
        type ciena-sat:rate-fixed-2-decimal;
        description
         "Rate in Mbps.";
      }

      leaf suspect-flag {
        type boolean;
        description
         "with port under test having 1:1 LAG/MC-LAG/Y-protection config and
        while a test was running and port protection swicth event is detected
then this flag shall
          be marked to true indicating results might no be valid as more or
more protection switch event detected
          while a test was running. Default value of flag is false.";
      }

      uses common-results-fields;
    }

    list gen-test-session-pdv-results {
      key "entity-name
           test-instance-name
           untagged
```


Copyright© 2022 Ciena® Corporation

```
      pcp
      color
      frame-size-index";

  description
   "generator test session packet delay variation results.
    If untagged is True, pcp and color are irrelevant.";

  leaf entity-name {
    type leafref {
      path "/ciena-sat:sat/ciena-sat:entity/ciena-sat:name";
    }
    description
     "Entity name.";
  }
  leaf test-instance-name {
    type leafref {
      path "/ciena-sat:sat/ciena-sat:test-instance/ciena-sat:name";
    }
    description
     "Test instance name.";
  }

  leaf untagged {
    type boolean;
    description
      "Untagged.";
  }

  leaf pcp {
    type priority;
    description
     "PCP value.";
  }

  leaf color {
    type basic-color;
    description
     "Color - green or yellow only.";
  }

  leaf frame-size-index {
    type uint8 {
      range "0..9";
    }
    description
     "Frame size index associated with these PDV stats.";
  }

  leaf avg {
    type uint32;
    description
     "Average PDV recorded for the given packet size.";
  }

  leaf samples {
    type uint32;
    description
     "Number of samples used for the test.";
  }
```


Copyright© 2022 Ciena® Corporation

```
      leaf kpi-result {
        type ciena-sat:kpi-result-type;
        description
         "Pass or fail results for the test which is determined
          by comparing the frameloss result against the selected
         KPI test instance's throughput PDV criteria. If no KPI test instance
          is selected for the given test instance Id, the result will
          be 'notApplicable'.";
      }

      leaf measurement-direction {
        type ciena-sat:measurement-direction;
        description
        "This indicates the direction of latency measurement used for the
latency test.";
      }

      leaf rate {
        type ciena-sat:rate-fixed-2-decimal;
        description
         "Rate in Mbps";
      }

      leaf suspect-flag {
        type boolean;
        description
         "with port under test having 1:1 LAG/MC-LAG/Y-protection config and
        while a test was running and port protection swicth event is detected
then this flag shall
          be marked to true indicating results might no be valid as more or
more protection switch event detected
          while a test was running. Default value of flag is false.";
      }

      uses common-results-fields;
    }

    list gen-test-session-outage-results {
      key "entity-name
           test-instance-name
           untagged
           pcp
           color
           frame-size-index";

      description
       "generator test session frameloss results.
        If untagged is True, pcp and color are irrelevant.";

      leaf entity-name {
        type leafref {
          path "/ciena-sat:sat/ciena-sat:entity/ciena-sat:name";
        }
        description
         "Entity name.";
      }

      leaf test-instance-name {
        type leafref {
```


Copyright© 2022 Ciena® Corporation

```
      path "/ciena-sat:sat/ciena-sat:test-instance/ciena-sat:name";
    }
    description
     "Test instance name.";
  }

  leaf untagged {
    type boolean;
    description
      "Untagged.";
  }

  leaf pcp {
    type priority;
    description
     "PCP value.";
  }

  leaf color {
    type basic-color;
    description
     "Color - green or yellow only.";
  }

  leaf frame-size-index {
    type uint8 {
      range "0..9";
    }
    description
     "Frame size index associated with these PDV stats.";
  }

  leaf outage-test-start-time {
    type yang:date-and-time;
    description
     "Date and time this outage test was started.";
  }

  leaf outage-test-fractional-start-time {
    type uint64;
    units "microseconds";
    description
     "The start time of this outage test in microseconds";
  }

  leaf outage-test-end-time {
    type yang:date-and-time;
    description
     "Date and time this outage test ended.";
  }

  leaf outage-test-fractional-end-time {
    type uint64;
    units "microseconds";
    description
     "The end time of this outage test in microseconds";
  }

  leaf total-outages {
    type uint32;
```


Copyright© 2022 Ciena® Corporation

```
    description
     "Total number of outages, both recorded and unrecorded.";
  }

  leaf recorded-outages {
    type uint32;
    description
     "Number of recorded outages with detailed statistics.";
  }

  leaf sum-total-outages {
    type uint64;
    units "microseconds";
    description
     "Sum of total outages, both recorded and unrecorded.";
  }

  leaf sum-recorded-outages {
    type uint64;
    units "microseconds";
    description
     "Sum of recorded outages.";
  }

  leaf average-throughput {
    type ciena-sat:throughput-result;
    description
     "Average throughput recorded in Mbps * 100.
      The value is multiplied by 100 to be able
      to send the results as an integer but should be divided
      by 100 when displayed to provide a 2 decimal point accuracy.
      A result of 123.45 is sent as 12345.";
  }

  leaf overall-kpi-result {
    type ciena-sat:kpi-result-type;
    description
     "Overall Pass or fail result for the test.
      FAIL if either sum-outages-kpi-result or
      max-outages-kpi-result or ANY of the single outage
      kpi-result leaves are FAIL.
      If no KPI instance is selected for the given
      test instance Id, the result will be 'notApplicable'.";
  }

  leaf sum-outages-kpi-result {
    type ciena-sat:kpi-result-type;
    description
     "Fail if the sum of recorded outages exceeds
      the maximum KPI specified.
      If no KPI test instance is selected for the given
      test instance Id, the result will be 'notApplicable'.";
  }

  leaf max-outages-kpi-result {
    type ciena-sat:kpi-result-type;
    description
     "Fail if the total number of outages exceeds
      the maximum KPI specified.
      If no KPI test instance is selected for the given
```


Copyright© 2022 Ciena® Corporation

```
          test instance Id, the result will be 'notApplicable'.";
      }

      leaf suspect-flag {
        type boolean;
        description
         "with port under test having 1:1 LAG/MC-LAG/Y-protection config and
        while a test was running and port protection swicth event is detected
then this flag shall
          be marked to true indicating results might no be valid as more or
more protection switch event detected
          while a test was running. Default value of flag is false.";
      }

      uses common-results-fields;

      list per-outage {
        key "outage-index";

        description
          "Each of the first 128 outages has its own details.";

        leaf outage-index {
          type uint8 {
            range "1..128";
          }
          description
           "Index of this individual outage entry";
        }

        leaf start-time {
          type yang:date-and-time;
          description
           "Date and time this individual outage began.";
        }

        leaf end-time {
          type yang:date-and-time;
          description
           "Date and time this individual outage ended.";
        }

        leaf length {
          type uint64;
          units "microseconds";
          description
           "Length of this individual outage in usecs.";
        }

        leaf frames-dropped {
          type uint64;
          description
           "Number of frames dropped during this outage.";
        }

        leaf kpi-result {
          type ciena-sat:kpi-result-type;
          description
           "Pass or fail results for the test which is determined
            by comparing the duration against the selected KPI
```


Copyright© 2022 Ciena® Corporation

```
            test instance's single outage duration pass criteria.
            If no KPI test instance is selected for the given test
            instance Id, the result will be 'notApplicable'.";
      }
    }
  }

  list emix-character-set {
    key "index";
    description
     "EMIX character and its associated frame size.";

    leaf index {
      type uint8 {
        range "1..26";
      }
      description
       "Character index.";
    }

    leaf character {
      type string {
        length "1";
      }
      description
       "EMIX character that can be used for creating an EMIX sequence.
        The character set currently includes: a b c d e f g h u v w x  ";
    }

    leaf frame-size {
      type string {
        length "1..6";
      }
      description
       "Frame size associated with the character. For h, the frame size
        is the port under test's MTU and the u character is a custom
        frame size configurable for each EMIX. The u frame size can take
        a value of 64 to 10000 and defaults to 594. If u is bigger than
        the MTU of the port under test, it will be reduced to the same
        size as h.";
    }
  }

  list emix-frame-size-for-sat-entity {
    key "entity-name
         emix-sequence";

    description
     "Frame size associated with given EMIX
      sequence and frame size index on a given Entity";

    leaf entity-name {
      type leafref {
        path "/ciena-sat:sat/ciena-sat:entity/ciena-sat:name";
      }
      description
       "Entity name.";
    }
    leaf emix-sequence {
      type leafref {
```


Copyright© 2022 Ciena® Corporation

```
        path "/ciena-sat:sat/ciena-sat:emix-sequence/ciena-sat:name";
      }
      description
       "Emix sequence name.";
    }

    leaf-list emix-frame-size {
      type uint32 {
        range "64..16000";
      }
      description
       "List of Frame sizes associated with the letters
        in the EMIX sequence.";
    }

    leaf avg-frame-size {
      type uint32 {
        range "64..16000";
      }
      description
       "Average frame size for the EMIX sequence on this entity.";
    }
  }

  list kpi-profile {
    key "name";
    description
     "Key performance indicator pass/fail criteria.";

    leaf name {
      type name-string;
      description
       "Name of the KPI profile";
    }

    leaf id {
      type uint32 {
        range "1..32";
      }
      description
       "Unique Id of the KPI profile entry";
    }

    leaf num-of-ref {
      type uint32;
      description
       "Number of generator test profiles currently
        using this KPI profile for results analysis.
        If this is the default KPI profile set in
        the generator, that also counts as one reference.";
    }
  }

  list bw-alloc-profile {
    key "name";

    description
     "BW allocation profile operational data.";

    leaf name {
```


Copyright© 2022 Ciena® Corporation

```
        type name-string;
        description
         "Name of the bandwidth allocation profile";
      }

      leaf id {
        type uint32 {
          range "1..32";
        }
        description
         "Unique Id of the bandwidth allocation
          profile entry";
      }

      leaf num-of-ref {
        type uint32;
        description
         "Number of generator test profiles currently
          using this bandwidth allocation profile
          for testing.
          If this is the default BW allocation
          profile set in the generator, that also counts
          as one reference.";
      }
    }

    list test-profile {
      key "name";

      description
        "Operational data for each test profile.";

      leaf name {
        type name-string;
        description
         "Name of the test profile";
      }

      leaf id {
        type uint32 {
          range "1..256";
        }
        description
         "index for test profile records.";
      }

      leaf hw-sessions-required {
        type uint32;
        description
         "This represents the number of hardware sessions
          required to enable/run a test instance using
          this profile according to the profile's current
          configuration.
          One hardware session is required for each outer tag
          PCP and color configured.
          Ex 1) Encap type set to dot1q, c-pcp is 0,3 and
                c-color is set to green, 2 PCPs * 1 color =
                2 hw sessions.
          Ex 2) Encap type set to qinq, s-pcp is 0,4,5 and
                s-color is set to green-yellow, then you
```


Copyright© 2022 Ciena® Corporation

```
                need 3 PCPs * 2 colors = 6 hw sessions.
            Ex 3) Encap type set to untagged, only 1 hw session
                  is required.";
        }

        leaf num-of-ref {
          type uint32;
          description
           "Number of generator test instances currently
            using this test profile for testing.";
        }

        leaf oper-bandwidth {
          type uint32;
          description
           "The bandwidth that is configured for this test-profile.
            It can either be the bandwidth configured in the
            test-profile or the maximum possible bandwidth on the platform.";
        }

        leaf oper-emix {
          type leafref {
            path "/ciena-sat:sat/ciena-sat:emix-sequence/ciena-sat:name";
          }
          description
         "This represents the EMIX sequence used with this test profile, which
is either
           the configured profile EMIX sequence or the global EMIX sequence .
When neither
           is configured the oper-emix is not reported in the object.";
        }

        leaf-list active-frame-size {
          type uint32;
          max-elements 10;
          description
           "If EMIX is not configured and the user has not configured any
            custom frame size list, this represent the active frame size list
           to be used for testing based on the configured encapsulation and IP
            address version.";
        }

        leaf oper-src-ip-addr {
          type inet:ip-address;
          description
         "This is the operational source IP address for the test packets. This
is only reported
           if PDU type in the test-profile is IP or UDP echo.";
        }

        leaf oper-dest-ip-addr {
          type inet:ip-address;
          description
         "This is the operational destination IP address for the test packets.
This is only reported
           if PDU type in the test-profile is IP or UDP echo.";
        }
      }

      list test-instance {
```


Copyright© 2022 Ciena® Corporation

```
key "name";

description
  "Operational data for a test instance.";

leaf name {
  type name-string;
  description
   "Name of the test instance";
}

leaf id {
  type uint32 {
    range "1..1280";
  }
  config false;
  description
   "Unique identifier for the test instance.";
}

leaf current-interval {
  type uint32;
  description
   "Applicable to generator test instances, this indicates the
    number of intervals of the selected tests have been run,
    including the current one, in relation to the total number of
    intervals to complete. This is based on the profile's
    interval and duration configured.";
}

leaf total-intervals {
  type uint32;
  description
   "Applicable to generator test instances, this indicates the
    total number of intervals of the selected tests to be run
    by a single start command on the test instance. This is
    based on the profile's interval and duration configured.
    A value of 0 should be interpreted as unknown and -1 as
    indefinite.";
}

leaf last-iteration-start-timestamp {
  type yang:date-and-time;
  description
   "Date and time of the last iteration that started since the
    last time the system was rebooted or the statistics were
    cleared on the test instance.";
}

leaf assoc-entity {
  type leafref {
    path "/ciena-sat:sat/ciena-sat:entity/ciena-sat:name";
  }
  description
   "Name of the entity this test instance is associated with.";
}

leaf oper-state {
  type ciena:oper-state;
  description
```


Copyright© 2022 Ciena® Corporation

```
    "Test instance operational enable/disable.";
}

leaf active-s-vid {
  type uint32;
  description
   "Active s-vid that is used by the Test-Instance.
   For Generator: it is user supplied vtag-stack when the
   test instance is created OR if not supplied, the value
   configured in the test profile.
   For Reflector: The active s-vid is provided on create
   Test-Instance, if not * or untagged 0 means not configured,
   and FFFF means any s-vid.";
}

leaf active-c-vid {
  type uint32;
  description
   "Active c-vid that is used by the Test-Instance.
   For Generator: it is user supplied vtag-stack when the
   test instance is created OR if not supplied, the value
   configured in the test profile.
   For Reflector: The active c-vid is provided on create
   Test-Instance, if not * or untagged 0 means not configured,
   and FFFF means any c-vid.";
}

leaf active-untagged-data {
  type boolean;
  description
   "When set to true, this indicates that the test instance
    is associated with untagged data traffic. When this is
    set, the active-s-vid and active-c-vid parameters are 0.";
}

leaf active-dst-mac {
type yang:mac-address;
description
  " Destination MAC address used for generator test
    instances. When set, this over writes the value
    specified in the referenced test profile.This
    field is not applicable for reflector TI.";
}

leaf active-start-bw {
  type uint32;
  description
   "Port Under Test's max bandwidth, if RFC-2544 is
    set in bandwidth profile, else bandwidth specified
    in bandwidth profile. This field is not applicable
    for reflector TI.";
}

leaf required-start-bw {
  type decimal64 {
      fraction-digits 2;
  }
  description
  "The required-start-bw is based on the profile configured color,
   bandwidth and excess bandwidth.
```


Copyright© 2022 Ciena® Corporation

```
          If color is green, required-start-bw = bandwidth,
          if yellow, it's = excess-bandwidth,
        if green-yellow, the required-start-bw = bandwidth + excess-bandwidth,
          and if green-yellow-red, the excess-bandwidth is multiplied by
          1.25 which can lead to a fraction.
          The required-start-bw indicates the bandwidth required in Mbps
          to start this test instance. It is based on the configured test
          profile configured tests, encapsulation and color.
        If rfc2544 is enabled, it is the max bandwidth of the port under test.
          Otherwise, if encapsulation is:
          A) Untagged, it is the profile bandwidth.
        B) dot1q or qinq, it uses c-color or s-color respectively, along with
            bandwidth and/or excess-bandwidth.";
      }

      leaf-list active-frame-size {
        type uint32;
        description
        "This is the list of frame sizes used for the test by this test-
instance. It is
         based on the encap type, pdu type and version of IP address fields.
This leaf is
        only available when EMIX sequence is not used for the test using this
test-instance";
      }

      leaf active-emix {
        type leafref {
          path "/ciena-sat:sat/ciena-sat:emix-sequence/ciena-sat:name";
        }
        description
        "This provides the name of the EMIX sequence used to run the test.
This is only reported
         if EMIX sequence is used for the test";
      }

      leaf active-src-ip-addr {
        type inet:ip-address;
        description
        "This field represents the source IP address used in the generated
test frame.
         This is only reported if PDU type is IP or UDP echo.";
      }

      leaf active-dest-ip-addr {
        type inet:ip-address;
        description
       "This field represents the destination IP address used in the generated
test frame.
         This is only reported if PDU type is IP or UDP echo.";
      }

      leaf test-start-timestamp {
        type yang:date-and-time;
        description
        "This is the time-stamp from the last time this test-instance was
started";
      }

      leaf test-end-timestamp {
```


Copyright© 2022 Ciena® Corporation

```
        type yang:date-and-time;
        description
        "This is the time-stamp when the test-instance has completed its
configured tests or
         whe the test was stopped by the user or some other event";
      }

      leaf suspect-flag {
        type boolean;
        description
         "with port under test having 1:1 LAG/MC-LAG/Y-protection config and
        while a test was running and port protection swicth event is detected
then this flag shall
          be marked to true indicating results might no be valid as more or
more protection switch event detected
          while a test was running. Default value of flag is false.";
      }

      leaf  port-protection-switch-count{
        type uint32;
        description
        "Applicable to test instances when port under test is having 1:1 LAG/
MC-LAG/Y-protectionconfig,
         this indicates the total number of times a port under test protection
event detected when
          reflector test operational state was reflecting.";
      }

      leaf port-last-protection-switch-time {
        type yang:date-and-time;
        description
         "Applicable to test instances when port under test is having
         1:1 LAG/MC-LAG/Y-protection config, indicates date and time of port
last protection
          switch event detected while a test instance operationl state was
reflecting.";
      }
    }
  }

  /*
   * RPC Operations
   */

  rpc entity-clear-statistics {
    description
      "Clear the statistics/counters of the targeted SAT entity.";

    input {
      leaf entity-name {
        type leafref {
          path "/ciena-sat:sat/ciena-sat:entity/ciena-sat:name";
        }
        mandatory "true";
        description
          "Name of a SAT entity to clear counters for. A reflector SAT entity
           has to be disabled in order to be able to clear its counters. A
           generator SAT entity cannot have any running test instances for
           a request to clear its counters to be successful.";
      }
```


Copyright© 2022 Ciena® Corporation

```
    }
    output {
      leaf status {
        type string;
        description
          "Status of the clear statistics operation.";
      }
    }
  }

  rpc test-instance-start {
    description
      "Start testing for the targeted generator test instance.";

    input {
      leaf test-instance-name {
        type leafref {
          path "/ciena-sat:sat/ciena-sat:test-instance/ciena-sat:name";
        }
        mandatory "true";
        description
         "Name of the generator test instance for which to initiate testing.";
      }
    }
    output {
      leaf status {
        type string;
        description
          "Status of the start test operation.";
      }
    }
  }

  rpc test-instance-stop {
    description
      "Stop testing for the targeted generator test instance.";

    input {
      leaf test-instance-name {
        type leafref {
          path "/ciena-sat:sat/ciena-sat:test-instance/ciena-sat:name";
        }
        mandatory "true";
        description
          "Name of the generator test instance for which to abort testing.";
      }
    }
    output {
      leaf status {
        type string;
        description
          "Status of the stop test operation.";
      }
    }
  }

  rpc test-instance-connectivity-test {
    description
      "Perform a connectivity test for the targeted generator test instance.";
    input {
```


Copyright© 2022 Ciena® Corporation

```
      leaf test-instance-name {
        type leafref {
          path "/ciena-sat:sat/ciena-sat:test-instance/ciena-sat:name";
        }
        mandatory "true";
        description
          "Name of the generator test instance targeted by this action.";
      }
    }
    output {
      leaf status {
        type string;
        description
          "Status of the trigger test operation.";
      }
    }
  }

  rpc test-instance-clear-statistics {
    description
      "Clear the results and counters for the targeted test instance.";

    input {
      leaf test-instance-name {
        type leafref {
          path "/ciena-sat:sat/ciena-sat:test-instance/ciena-sat:name";
        }
        mandatory "true";
        description
          "Name of the test instance targeted by this action. Reflector test
           instances have to be disabled prior to clearing their counters,
           and geneartor test instances cannot be running tests.";
      }
    }
    output {
      leaf status {
        type string;
        description
          "Status of the clear statistics operation.";
      }
    }
  }


  /*
   * Notifications  - Not supported in 6.16.1
   */

  notification test-started {
    description
      "Testing for this generator test instance has been initiated.";

    uses notification-base-fields;

    leaf port {
      type mef-logical-port:logical-port-ref;
      description
        "port under test.";
    }
```


Copyright© 2022 Ciena® Corporation

```
    leaf flow-point-id {
      type mef-fp:fp-ref;
      description
        "Flow-point related to test instance.";
    }

    leaf svid {
      type ieee:vlanid;
      description
        "S-VID.";
    }
    leaf cvid {
      type ieee:vlanid;
      description
        "C-VID.";
    }
    leaf untagged {
      type boolean;
      description
        "When true, this indicates that the test instance
         is associated with untagged data traffic.";
    }
    leaf dst-mac {
      type yang:mac-address;
      description
        "Mac address.";
    }
  }

  notification test-stopped {
    description
      "Testing for this generator test instance has been stopped.";

    uses notification-base-fields;

    leaf stopped-by {
      type string;
      description
        "Stopped-by is the cause of test stoppage. Test can stop
         either by user intervention or because of an error.";
    }
  }

  notification test-completed {
    description
      "Testing for this generator test instance has completed.";

    uses notification-base-fields;
  }

  notification test-iteration-completed {
    description
      "Testing for this generator test instance has completed.";

    uses notification-base-fields;

    leaf iteration {
      type uint32;
      description
        "iteration.";
```


Copyright© 2022 Ciena® Corporation

```
    }
  }

  notification test-throughput-kpi-failure {
    description
      "This generator test instance's iteration throughput results are
       below the set KPI value.";

    uses notification-test-kpi-fields;

    leaf throughput-result {
      type uint32;
      description
        "Maximum throughput recorded for the given packet size
         in Mbps * 100. The value is multiplied by 100 to be able
         to send the results as an integer but should be divided
         by 100 when displayed to provide a 2 decimal point accuracy.
         A result of 123.45 is sent as 12345.";
    }
  }

  notification test-frameloss-kpi-failure {
    description
      "This generator test instance's iteration frameloss results are
       above the set KPI value.";

    uses notification-test-kpi-fields;

    leaf frameloss-result {
      type uint32;
      description
        "Frame loss results in Percent of frame loss start bandwidth
         are sent as unsigned integer multitplied by 100 to provide a
         2 decimal point accuracy.
         If result is 12.34 %, it is sent as 1234 and should
         be divided by 100 by the application retrieving the
         data.";
    }
  }

  notification test-latency-kpi-failure {
    description
      "This generator test instance's iteration latency results are
       above the set KPI value.";

    uses notification-test-kpi-fields;

    leaf latency-result {
      type uint32;
      description
        "Maximum latency recorded for the given packet size.";
    }
  }

  notification test-pdv-kpi-failure {
    description
      "This generator test instance's iteration packet delay variation
       results are above the set KPI value.";

    uses notification-test-kpi-fields;
```


Copyright© 2022 Ciena® Corporation

```
    leaf pdv-result {
      type uint32;
      description
        "Average PDV recorded for the given packet size.";
    }
  }

  notification test-outage-max-number-kpi-failure {
    description
      "This generator test instance's outage results do not
       meet the set KPI values for max number of outages.";

    uses notification-test-kpi-fields;

    leaf num-outages {
      type uint32;
      description
        "Number of outages that occurred during the outage test.";
    }
  }

  notification test-outage-single-outage-kpi-failure {
    description
      "This generator test instance's iteration outage
       results do not meet the set KPI values.";

    uses notification-test-kpi-fields;

    leaf single-outage-max-duration {
      type uint64;
      units "microseconds";
      description
        "Length of maximum individual outage in usecs.";
    }
  }

  notification test-outage-sum-outages-kpi-failure {
    description
      "This generator test instance's outage results do not
       meet the set KPI values for max time of total recorded outages.";

    uses notification-test-kpi-fields;

    leaf sum-recorded-outages {
      type uint64;
      units "microseconds";
      description
        "Time in microseconds of the sum of recorded outages.";
    }
  }

  notification entity-port-protection-switch-event {
    description
      "entity port under test protection switch occured
       applicable when port under test is having 1:1 LAG/MC-LAG/Y-protection
config";

    leaf severity {
      type ciena:event-severity;
```


Copyright© 2022 Ciena® Corporation

```
      description
        "Severity.";
    }

    leaf mac-address {
      type yang:mac-address;
      description
        "Mac address.";
    }

    leaf name {
        type name-string;
        description
         "Name of the service activation test entity.";
    }

    leaf port {
      type mef-logical-port:logical-port-ref;
      description
        "port under test.";
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-security-certs.yang

```
module ciena-security-certs {
  namespace "http://www.ciena.com/ns/yang/ciena-security-certs";
  prefix "ciena-security-certs";

  import ietf-inet-types {
    prefix "inet";
  }

  import ietf-yang-types {
    prefix "yang";
  }

  import ciena-types {
    prefix ciena;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
    E-mail:  support@ciena.com
    Postal:  7035 Ridge Road
             Hanover, Maryland 21076
             U.S.A.
    Phone:   +1 800-921-1144
    Fax:     +1 410-694-5750";

  description
    "This YANG module defines the nodes
     for Security Certs CA and CRL for the NETCONF protocol.

    Copyright (c) 2016 Ciena Corporation.  All rights
         reserved.

         All information contained herein is, and remains
         the property of Ciena Corporation. Dissemination of this
         information or reproduction of this material is strictly
         forbidden unless prior written permission is obtained from
         Ciena Corporation.";

  revision "2016-04-21"
    {
      description
          "Initial version";
      reference
          "RFC 6020: YANG - A Data Modeling Language for
              the Network Configuration Protocol (NETCONF).
              No specific reference; standard not available.";
    }

  /* NOTES:
   * 1. Designed for 6x. Did not find any security cert stuff for 8x.
   *    CLI: system security ca-certificate ,and system security crl.
   * 2. Are there any notifications that Security Cert should have/has?
  *    If a module is new; should we create/consider adding new notifications?
   * 3.
   *
```


Copyright© 2022 Ciena® Corporation

```
   */

  /*
   * Groupings
   */

  /*
   * No configuration data nodes
   *
   * All configuration is done with RPCs.
   *
   */

  /*
   * Operational state data nodes
   */
  container security-certs-state {
    description "CA and CRL Certificate state.";

    container ca-certificates {
      description
        "CA Certificates.";

      list ca-certificate {
        key certificate-hash; // system security ca-certificate show ca-cert-
hash d2cdd458;
       description          // should be able to select by key/ca-cert-hash.
          "The CA certificate list.";

        leaf certificate-hash {
          type yang:hex-string;
          description
            "The CA certificate hash.";
        }

        leaf subject-common-name {
          type string;
          description
            "The CA subject common name.";
        }

        leaf issuer-common-name {
          type string;
          description
            "The CA issuer common name.";
        }

        leaf valid-to {
          type string;
          description
            "The CA valid to.";
        }

      } // end list certificate
    } // end container certificates

    container crl-certificates {
      description
        "Certificate Revocation Lists.";
```


Copyright© 2022 Ciena® Corporation

```
      list crl-certificate {
        key crl-hash;

        leaf crl-hash {
          type yang:hex-string;
          description
            "The CRL Hash.";
        }

        leaf issuer-common-name {
          type string;
          description
            "The CRL issuer common name.";
        }

        leaf last-update {
          type string;
          description
            "The CRL last update.";
        }

        leaf revoked-cert-count {
          type int32;
          description
            "The CRL revoked cert count.";
        }
        description
          "The CRL Certificate List.";

      } // end list crl certificate
    } // end container crl certificate
  } // end container security certs state

  /*
   * RPC Operations
   */
  rpc ca-install {
    description
      "CA Certificate Install.";

    input {
      description
        "The CA Install Arguments.";

      leaf filename {
        type string;
        description
          "The CA certificate filename.";
      }

      leaf force {
        type boolean;
        description
          "Use force to overwrite a CA certificate with same filename.";
      }

      leaf server-type {
        type enumeration {
          enum default-server {
            description
```


Copyright© 2022 Ciena® Corporation

```
            "The default server.";
        }
        enum default-tftp-server {
          description
            "The default tftp server.";
        }
        enum default-ftp-server {
          description
            "The default ftp server.";
        }
        enum default-sftp-server {
          description
            "The default sftp server.";
        }
        enum tftp-server {
          description
            "The tftp server.";
        }
        enum ftp-server {
          description
            "The ftp server.";
        }
        enum sftp-server {
          description
            "The sftp server.";
        }
        enum url {
          description
            "The url.";
        }
      }
      description
        "The server type.";
    } // end leaf server type

    leaf server-address {
      type inet:host;
      description
        "ftp-server IP address or host name.";
    }

    leaf login-id {
      type string;
      description
        "The server user name / login id.";
    }

    leaf password {
      type string;
      description
        "The server password.";
    }

  }
  output {
    leaf status {
      type string;
      description
        "Result of operation.";
    }
```


Copyright© 2022 Ciena® Corporation

```
    }
  }

  rpc ca-uninstall {
    description
      "CA Certificate Uninstall.";

    input {
      description
        "The CA Certificate Uninstall Arguments.";

      leaf ca-cert-hash {
        type yang:hex-string;
        description
          "The CA Cert Hash.";
      }

    }
    output {
      leaf status {
        type string;
        description
          "Result of operation.";
      }
    }
  }

  rpc crl-install {
    description
      "CRL Certificate Install.";

    input {
      description
        "The CRL Certificate Install Arguments.";

      leaf filename {
        type string;
        description
          "The CRL Certificate Filename.";
      }

      leaf force {
        type boolean;
        description
          "Use force to overwrite a CRL with same filename.";
      }

      leaf server-type {
        type enumeration {
          enum default-server {
            description
              "The default server.";
          }
          enum default-tftp-server {
            description
              "The default tftp server.";
          }
          enum default-ftp-server {
            description
              "The default ftp server.";
```


Copyright© 2022 Ciena® Corporation

```
          }
          enum default-sftp-server {
            description
              "The default sftp server.";
          }
          enum tftp-server {
            description
              "The tftp server.";
          }
          enum ftp-server {
            description
              "The ftp server.";
          }
          enum sftp-server {
            description
              "The sftp server.";
          }
          enum url {
            description
              "The url.";
          }
        }
        description
          "The server type.";
      } // end leaf server type

      leaf server-address {
        type inet:host;
        description
          "ftp-server IP address or host name.";
      }

      leaf login-id {
        type string;
        description
          "The server user name / login id.";
      }

      leaf password {
        type string;
        description
          "The server password.";
      }

    }
    output {
      leaf status {
        type string;
        description
          "Result of operation.";
      }
    }
  }

  rpc crl-uninstall {
    description
      "CRL Certificate Uninstall.";

    input {
      description
```


Copyright© 2022 Ciena® Corporation

```
        "CRL Certificate Uninstall Arguments.";

      leaf crl-hash {
        type yang:hex-string;
        description
          "The CRL Hash.";
      }

    }
    output {
      leaf status {
        type string;
        description
          "Result of operation.";
      }
    }
  }

  /*
  rpc <template> {
    description
      "";

    input {

    }
    output {
      leaf status {
        type string;
        description
          "Result of operation.";
      }
    }
  }
  */

  /*
   * Notifications
   */
  // <notification template>

  /*
  notification <rpc template> {
    description
     "";

    container <rpc template> {
      description
        "";
    }
  }
  */

} // end ciena template model
```


Copyright© 2022 Ciena® Corporation

ciena-security-log.yang

```
module ciena-security-log {
  namespace "http://www.ciena.com/ns/yang/ciena-security-log";
  prefix "ciena-security-log";

  import ietf-inet-types {
    prefix inet;
  }

  import ciena-types {
    prefix ciena-types;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module defines the nodes for
     Security Logging for the NETCONF protocol.

     Copyright (c) 2016 Ciena Corporation.  All rights
     reserved.

     All information contained herein is, and remains
     the property of Ciena Corporation. Dissemination of this
     information or reproduction of this material is strictly
     forbidden unless prior written permission is obtained from
     Ciena Corporation.";

  revision 2017-11-01 {
    description
      "Added export-port to the security-log/transfer container.";
  }

  revision 2017-08-07 {
    description
      "Removed dependency on ciena-xftp.";
    reference
      "";
  }

  revision "2016-02-22"
  {
    description
      "Initial version";
    reference
      "RFC 6020: YANG - A Data Modeling Language for
       the Network Configuration Protocol (NETCONF).
       No specific reference; standard not available.";
  }
```

```
typedef server-types {
  type enumeration {
    enum none {
      description
        "The none server type.";
    }
    enum tftp-server {
      description
        "The system's TFTP server.";
    }
    enum ftp-server {
      description
        "The system's FTP server.";
    }
    enum sftp-server {
      description
        "The system's SFTP server.";
    }
    enum xftp-server {
      description
        "The system's default file transfer server.";
    }
    enum http-server {
      description
        "The http server type.";
    }
  }
  description
    "Type used to define the file transfer servers.";
}

/* Nodes with unresolved questions:
 * a. Only in 6x; not in 8x.
 * 1. operstate/ log-file usage?
 *
 */

/*
 * Configuration data nodes
 */
container security-log {
  description
    "The security log.";

  container transfer {
    description
      "The security log transfer settings.";

    leaf admin-state {
      type ciena-types:admin-state;
      description
        "The admin state of the security log.
         Also called Transfers (auto)";
    }

    leaf export-mode {
      type server-types;
      description
        "The file transfer sever types.";
```


Copyright© 2022 Ciena® Corporation

```
}

leaf export-server {
  type string;
  description
    "The hostname or ip address of the export server.";
}

leaf export-port {
  type inet:port-number;
  description
    "The port on the export server. A value of 0 is returned if
     no port is set to indicate the server will use the default
     port for that export-mode.";
}

leaf export-username {
  type string {
    length "0..32";
  }
  description
    "The username for the export server.";
}

uses ciena-types:security-password;

leaf export-path {
  type string {
    length "0..40";
  }
  description
    "The file path used on the export server.";
}

leaf periodic-export {
  type int32;
  description
    "The time delay or interval in seconds before the next file
     export will occur.";
}

leaf export-discards {
  type ciena-types:on-off-state;
  description
    "The on/off value to allow export discards.";
}

leaf transfer-timeout {
  type int32 {
    range "0..100";
  }
  description
    "A value in seconds to represent the transfer timeout time
     interval.";
}

leaf transfer-retries {
  type int32 {
    range "0..10";
  }
```


Copyright© 2022 Ciena® Corporation

```
        description
          "The number of transfer retries.";
      }

      leaf retry-interval {
        type int32 {
          range "0..300";
        }
        description
          "The retry interval in seconds.";
      }
    }

  } // end command-log container

  /*
   * Operational state data nodes
   */
  container security-log-state {
    config false;
    description
      "The security log state.";

    container transfer {
      description
        "The log transfer state.";

      leaf oper-state { // system security log show status
        type ciena-types:oper-state;
        description
          "The security log oper state.";
      }

      leaf export-state {
        type enumeration {
          enum Idle {
            description
              "This state is either 'busy...' or 'idle' ";
          }
          enum Busy {
            description
              "The busy element.";
          }
        }
        description
          "The export state.";
      }

      leaf random-staggered-start {
        type string;
        description
          "A duration string.";
      }
    }  // end transfer container

    // Need to review how transfer of files is done in netconf.
    // Should the file transfer logic be in ciena-types as a standard
    // way to transfer files?
    // The Binary type is not supported in NETCONF development
    // scripts... right now.
```


Copyright© 2022 Ciena® Corporation

```
container log-file {
  description
    "Log file transfer.";
  // system security log show / system security log file
  leaf security-log-contents { // or security-log-file?
    type binary;
    description
      "The security log file. Not tested.
       The assumption is that his leaf will be able to send an
       entire file to the user.
       This leaf is attempting to send the entire security log
       output from 'system security log show'. OR -- what files does
       the 'system security log show' command read? when displayed
       on the CLI?
       Are we sending the security log files in /flash0/seclog':?
       secLog.1437329412

       The security log files can be extremely long.
       Are there concerns about netconf server excessive load
       / possible processing delay.
        -- can one request block other netconf server requests?


       The binary built-in type represents any binary data, i.e.,
       a sequence of octets.
       Binary values are encoded with the base64 encoding scheme.
       Base64  binary-to-text encoding schemes.
       Base 64 Encoding
       https://tools.ietf.org/html/rfc4648#section-4.";
  } // end security-log-... leaf

  leaf argument-type {
    type enumeration {
      enum containing { // containing <String[1..128]>
        description
          "The containing element.";
      }
      enum failed {
        description
          "The failed element.";
      }
      enum from-date { // from-date <date: yyyy-mm-dd or yy-mm-dd or
                       // mm-dd>
        description
          "The from-data element.";
      }
      enum from-time { // from-time <time: hh:mm:ss or hh:mm>
        description
          "The from-time element.";
      }
      enum last { //  last <duration: {N[yMwdhms]}* e.g. 1h10m3s>
        description
          "The last element.";
      }
      enum recent {
        description
          "The recent element.";
      }
      enum since-activated {
        description
```


Copyright© 2022 Ciena® Corporation

```
            "The since-activated element.";
        }
        enum tail { // tail <NUMBER: 1..4294967295>
          description
            "The tail element.";
        }
        enum today {
          description
            "The today element.";
        }
        enum to-date { // to-date <date: yyyy-mm-dd or yy-mm-dd or
                       // mm-dd>
          description
            "The to-date element.";
        }
        enum to-time {  // to-time <time: hh:mm:ss or hh:mm>
          description
            "The to-time element.";
        }
        enum yesterday {
          description
            "The yesterday element.";
        }
      }
      description
        "The argument type.";
    } // end log-argument-type leaf

    leaf argument-value {
      type string;
      description
        "The argument value.  This value will be taken as a string,
         and converted to the type value.  The value will be checked
         for the format expected by the argument type.  This expects
         that each argument type only has one argument value.";
    }

  } // end log-file container

  // stats oper. data
  list statistics {
    description
      "Statistics State Data.";

    leaf event-id {
      type string;
      description
        "The event id.
         A hex value.";
    }

    leaf disable {
      type string;
      description
        "Disables the count.";
    }

    leaf count {
      type int32;
      description
```


Copyright© 2022 Ciena® Corporation

```
          "The count of how many event-ids have went by.";
      }
    }

  } // end oper data

  /*
   * RPC
   */

  // replaced security-admin-state container in config.
  rpc security-log-eventid-state {
    description
      "Set the EventID state in the security log.
       The security admin state.
       Block/unblock Events from the secLog.
       We only want to write, and not read it, ...
       with a RPC we can write, and not read, ...";

    input {

      leaf admin-state {
        type ciena-types:admin-state;
        description
          "The admin state of the event id number.";
      }
      leaf event-id {
        type int32;
        description
          "The event id number.";
      }
    }

    output {
      leaf status {
        type string;
        description
          "Result of operation.";
      }
    }
  }

  rpc security-log-clear {
    description
      "Security Log Clear.";

    output {
      leaf status {
        type string;
        description
          "Result of operation.";
      }
    }
  }

  rpc security-log-clear-statistics {
    description
      "Security Log Clear Statistics";

    output {
```


Copyright© 2022 Ciena® Corporation

```
      leaf status {
        type string;
        description
          "Result of operation.";
      }
    }
  }

  rpc security-log-transfer-now {
    description
      "Security Log Transfer Now";

    output {
      leaf status {
        type string;
        description
          "Result of operation.";
      }
    }
  }

} // end ciena-security-log module
```


Copyright© 2022 Ciena® Corporation

ciena-snmp.yang

```
module ciena-snmp {
  namespace "http://ciena.com/ns/yang/ciena-snmp";
  prefix ciena-snmp;

  import ietf-snmp {
    prefix snmp;
  }

  import ietf-yang-types {
    prefix yang;
  }

  import ietf-netconf-acm {
    prefix nacm;
  }

  import ietf-inet-types {
    prefix inet;
  }

  import ciena-types {
    prefix ciena;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
    E-mail:  yang@ciena.com
    Postal:  7035 Ridge Road
             Hanover, Maryland 21076
             U.S.A.
    Phone:   +1 800-921-1144
    Fax:     +1 410-694-5750";

  description
    "This YANG module defines Ciena augmentations to the ietf-snmp.yang
     module.

    Copyright (c) 2016 Ciena Corporation.  All rights reserved.

         All information contained herein is, and remains
         the property of Ciena Corporation. Dissemination of this
         information or reproduction of this material is strictly
         forbidden unless prior written permission is obtained from
         Ciena Corporation.";

  revision "2017-10-02"
  {
    description
      "Removed mandatory statements from augmented nodes.
       Added various descriptions.
       Adjusted maximum line length to 72 characters.";
    reference
      "RFC 6020: YANG - A Data Modeling Language for
       the Network Configuration Protocol (NETCONF).
       Section 7.15: The augment Statement.";
```

```
  }

  revision "2017-04-24"
  {
    description
      "Added context-map feature and context-map list for attaching
       routing protocol instances to SNMP contexts.
       Added SNMP bulk repetitions and PDU size.";
    reference
      "Ciena Functional Specification.";
  }

  revision "2017-01-19"
  {
    description
      "Added notification list for management of the configuration of
       SNMP notifications.";
    reference
      "Ciena Functional Specification.";
  }

  revision "2016-08-13"
  {
    description
      "Initial version";
    reference
      "RFC 6020: YANG - A Data Modeling Language for
       the Network Configuration Protocol (NETCONF).
       No specific reference; standard not available.";
  }

  feature context-map {
    description
      "A server implements this feature if it has the ability to attach
       routing protocol instances to SNMP contexts.";
  }

  grouping key {
    description
      "Base grouping for secret keys.";

    leaf key {
      type yang:hex-string;
      nacm:default-deny-all;
      description
        "Localized key specified as a list of colon-specified
         hexadecimal octets.";
    }
  }

  grouping triple-des {
    description
      "Base grouping for 3DES nodes.";

    container triple-des {
      description
        "Configurable parameters for 3DES encryption.";

      uses key;
      reference
```


Copyright© 2022 Ciena® Corporation

```
        "The 3DES-EDE Symmetric Encryption Protocol.
         This is the protocol as specified in
         draft-reeder-snmpv3-usm-3desede-00, and in the
         updates to that draft available at
         http://www.snmp.com/eso/.";
    }
  }

  augment
  /snmp:snmp/snmp:usm/snmp:local/snmp:user/snmp:priv/snmp:protocol {
    description
      "Augmentation for the addition of 3DES encryption nodes.";

    case triple-des {
      uses triple-des;
    }
  }

  augment
  /snmp:snmp/snmp:usm/snmp:remote/snmp:user/snmp:priv/snmp:protocol {
    description
      "Augmentation for the addition of 3DES encryption nodes.";

    case triple-des {
      uses triple-des;
    }
  }

  augment /snmp:snmp/snmp:engine {
    description
      "SNMP engine configuration settings";

    leaf pdu-size     {
      type uint32 {
        range "256..9216";
      }
      units bytes;
        description
          "Maximum allowable PDU size generated by the SNMP entity";
        reference
          "RFC 1157 Section 4";
    }
    leaf bulk-repetitions {
      type uint32 {
        range "0..1024";
      }
      units repetitions;
      description
        "Maximum allowable bulk-repetitions returned when a bulk
        get request is received.";
      reference
        "RFC 3416 Section 2.3";
    }
  }

  augment /snmp:snmp {
    description
      "Transport targets to be used in the generation of SNMP
       operations";
```


Copyright© 2022 Ciena® Corporation

```
    list target {
      key name;
      description
        "List of targets.";
      reference
        "RFC 3413: Simple Network Management Protocol (SNMP)
           Applications.
           SNMP-TARGET-MIB.snmpTargetAddrTable";

      leaf name {
        type snmp:identifier;
        description
          "Identifies the target.";
        reference
          "RFC 3413: Simple Network Management Protocol (SNMP)
             Applications.
             SNMP-TARGET-MIB.snmpTargetAddrName";
      }

      leaf udp-ip {
        type inet:ip-address;
        description
          "The UDP transport address for the target.";
        reference
          "RFC 3413: Simple Network Management Protocol (SNMP)
           Applications.
           SNMP-TARGET-MIB.snmpTargetAddrTAddress";
      }

      leaf udp-port {
        type inet:port-number;
        default 162;
        description
          "UDP port number for the target.";
        reference
          "RFC 3413: Simple Network Management Protocol (SNMP)
           Applications.
           SNMP-TARGET-MIB.snmpTargetAddrTAddress";
      }

      leaf prefix-length {
        type uint8 {
          range "0..128";
        }
        description
          "The value of this leaf must match the value of
           ../snmp:ip.  If ../snmp:ip contains an IPv4 address,
           this leaf must be less than or equal to 32.  If it
           contains an IPv6 address, it must be less than or
           equal to 128.

           Note that the prefix-length is currently only used
           by the Community-based Security Model to filter
           incoming messages.  Furthermore, the prefix-length
           filtering does not cover all possible filters
           supported by the corresponding MIB object.";


        reference
          "RFC 3584: Coexistence between Version 1, Version 2,
```


Copyright© 2022 Ciena® Corporation

```
            and Version 3 of the Internet-standard
            Network Management Framework.
            SNMP-COMMUNITY-MIB.snmpTargetAddrTMask";
      }

      leaf-list tag {
        type snmp:tag-value;
        description
          "List of tag values used to select target addresses.";
        reference
          "RFC 3413: Simple Network Management Protocol (SNMP).
             Applications.
             SNMP-TARGET-MIB.snmpTargetAddrTagList";
      }

      leaf timeout {
        type uint32;
        units "0.01 seconds";
        default 1500;
        description
          "Needed only if this target can receive
           InformRequest-PDUs.";
        reference
          "RFC 3413: Simple Network Management Protocol (SNMP).
             Applications.
             SNMP-TARGET-MIB.snmpTargetAddrTimeout";
      }

      leaf retries {
        type uint8;
        default 3;
        description
          "Needed only if this target can receive
           InformRequest-PDUs.";
        reference
          "RFC 3413: Simple Network Management Protocol (SNMP).
             Applications.
             SNMP-TARGET-MIB.snmpTargetAddrRetryCount";
      }

      leaf target-params {
        type snmp:identifier;
        description
          "SNMP target information to be used in the generation of SNMP
           messages";
        reference
          "RFC 3413: Simple Network Management Protocol (SNMP).
             Applications.
             SNMP-TARGET-MIB.snmpTargetAddrParams";
      }
    } // end list target

    list context-map {
      if-feature context-map;
      key context;
      description
        "A list of SNMP contexts to which routing instances can be
         attached";

      leaf context {
```


Copyright© 2022 Ciena® Corporation

```
        type snmp:context-name;
        description
          "An SNMP context is a collection of management information
           accessible by an SNMP entity. This leaf refers to the name
           of a specific SNMP context.";
         reference
           "RFC 3415: View-based Access Control Model (VACM) for
            the Simple Network Management Protocol (SNMP).";
      }

      leaf-list isis-instance {
        type string {
          length "1..32";
        }
        description
          "A list of a routing instances configured for Intermediate
           System to Intermediate System (IS-IS) protocol that are
           attached to this SNMP context.";
      }

      leaf-list ospf-instance {
        type string {
          length "1..32";
        }
        description
          "A list of a routing instances configured for Open Shortest
           Path First (OSPF) protocol that are attached to this SNMP
           context.";
      }
    } // end list context-map

    list notification {
      key name;
      description
        "Configuration of SNMP notifications.";

      leaf name {
        type string {
          length "0..128";
        }
        description
          "The name of the notification.";
      }

      leaf class {
        type string {
          length "0..32";
        }
        description
          "The name of the MIB class to which the notification
           belongs.";
      }

      leaf mib {
        type string {
          length "0..128";
        }
        description
          "The name of the MIB to which the notification belongs.";
      }
```


Copyright© 2022 Ciena® Corporation

```
      leaf state {
        type ciena:admin-state;
        description
          "The administrative state of the notification.";
      }

      leaf count {
        type uint16 {
          range "0..65535";
        }
        description
          "An accumulation threshold for the number of events sent
           within the delay period.";
      }

      leaf delay {
        type uint16 {
          range "1..300";
        }
        description
          "Delay period for the accumulation threshold in seconds.";
      }
    } // end list notification
  } // end augment /snmp:snmp
}
```


Copyright© 2022 Ciena® Corporation

ciena-soam-pm-sds.yang

```
module ciena-soam-pm-sds {
  namespace "http://www.ciena.com/ns/yang/ciena-soam-pm-sds";
  prefix "ciena-soam-pm-sds";

  import mef-cfm {
    prefix "mef-cfm";
    revision-date 2012-04-16;
  }

  import ietf-yang-types {
    prefix yang;
    revision-date 2010-09-24;
  }

  import mef-soam-pm {
    prefix "mef-soam-pm";
    revision-date "2012-04-16";
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module contains the management data definitions for the
management of
    Ethernet Services Operations, Administration and maintenance for
Performance
    Monitoring and extends the Connectivity Fault Management (CFM) YANG
modules.";

  revision 2017-09-13 {
    description
      "Augment MEF 35 Loss/Delay session:
          - To add MD/MA/MEP leaf references in session key of create RPCs.
          - To support creation of offloaded MEF sessions.
         - To support configuration of lower-bound values of each bin for a
given number of frame delay, jitter and frame delay range bins.
       Augment legacy SLM session create RPC to support configuration of DEI
bit.
        Add support for configuration for frame delay range bins.";
  }

  revision 2016-12-19 {
    description
      "Initial version";
  }

//
// Type Definitions related to ciena proprietary extansion of Y.1731 SOAM
```


Copyright© 2022 Ciena® Corporation

```
//

  identity loss-measurement-type {
    description
     "Indicates the type of loss measurement type ";
  }

  identity loss-measurement-type-lmm {
    base loss-measurement-type;
    description
     "Indicates the loss measurement type as lmm (FrameLoss)";
  }

  identity loss-measurement-type-slm {
    base loss-measurement-type;
    description
     "Indicates the loss measurement type as slm (SyntheticLoss)";
  }

  identity message-interval-type {
    description
     "Indicates the message interval type being used in the PM session";
  }

  identity message-interval-type-3.3ms {
    base message-interval-type;
    description
     "Indicates the message interval type to be 3.3ms";
  }

  identity message-interval-type-10ms {
    base message-interval-type;
    description
     "Indicates the message interval type to be 10ms";
  }

  identity message-interval-type-100ms {
    base message-interval-type;
    description
     "Indicates the message interval type to be 100ms";
  }

  identity message-interval-type-1s {
    base message-interval-type;
    description
     "Indicates the message interval type to be 1sec";
  }

  identity message-interval-type-10s {
    base message-interval-type;
    description
     "Indicates the message interval type to be 10sec";
  }

  identity message-interval-type-1m {
    base message-interval-type;
    description
     "Indicates the message interval type to be 1min";
  }
```


Copyright© 2022 Ciena® Corporation

```
  identity message-interval-type-10m {
    base message-interval-type;
    description
     "Indicates the message interval type to be 10min";
  }

//
// Groupings related to ciena proprietary extension of y.1731
//

  grouping saos-cfm-service-identifier {
    description
      "This grouping uniquely identifies a saos cfm service.";

    leaf maintenance-domain-id {
       type leafref {
         path "/mef-cfm:maintenance-domain/mef-cfm:id";
       }
    }

    leaf maintenance-association-id {
       type leafref {
         path "/mef-cfm:maintenance-domain/mef-cfm:maintenance-association/
mef-cfm:id";
       }
    }

    leaf local-mep-id {
       type leafref {
         path "/mef-cfm:maintenance-domain/mef-cfm:maintenance-association/
mef-cfm:maintenance-association-end-point/mef-cfm:mep-identifier";
       }
    }
  }

  grouping saos-remote-mep-identifier {
    description
      "This grouping includes objects which identify a remote MEP.";

    choice remote-mep {
      description
       "The remote MEP can be identified by either a MAC address or a MEP ID";

      case mac-address {
        leaf mac-address {
          type yang:mac-address;

          description
            "The Target MAC Address Field to be transmitted: A unicast
             destination MAC address.";
        }
      }

      case mep-id {
        leaf mep-id {
          type string;

          description
           "The Maintenance Association End Point Identifier of another MEP in
             the same Maintenance Association to which the SOAM Loss or Delay
```


Copyright© 2022 Ciena® Corporation

```
              Measurement frame is to be sent.";
          }
        }
      }
    }

  grouping soam-pm-saos-attr {
    description
      "This grouping includes additional attributes that SAOS can take as an
input
       to start a Performance monitoring session. These attributes are common
for both
        loss measurement and delay measurement sessions";

    leaf test-id {
      type uint32;
      description
        "An integer value that uniquely identifies a PM session
        between a pair of MEP.";
    }

    leaf count {
      type uint32 {
        range "2..65535";
      }
      description
       " An integer value which specifies the number of frames that need to be
        transmitted within a given iteration of measurement session.";
    }

    leaf duration {
      type uint32 {
        range "1..65535" ;
      }
      units seconds;
      description
        "Indicates measurement duration of single y1731 iteration ";
    }

    leaf message-interval {
      type identityref {
        base message-interval-type;
      }

      description
        "This object specifies the interval between Performance Measurement
OAM message
         transmission.";
    }

    leaf priority {
      type mef-cfm:priority-type;

      description
        "This object specifies the priority of frames with Performance
         Monitoring OAM message information.";
    }

    leaf repeat-delay {
      type uint32 {
```


Copyright© 2022 Ciena® Corporation

```
        range "0..1440";
      }
      units minutes;
      description
        "Indicates delay between multiple iteration of y1731 session";
    }

    leaf iterate {
      type uint8 {
        range "0..60";
      }
      description
        "An Integral Value indicating the number of iteration that need to
performed within a
       given duration of scheduled measurement session. The value 0 indicates
a non-terminating session";
    }

    leaf timeout {
      type uint16 {
        range "2000..10000";
      }
      units milliseconds;
      description
        "Indicates the timeout value of given iteration";
    }

    leaf frame-size {
      type uint16 {
        range "64 .. 1400";
      }
      units bytes;
      default 64;

      description
       "This object specifies the Performance Measurement frame size between
64 bytes
         and the maximum transmission unit of the EVC.";
    }

    leaf accelerate {
      type boolean;
      description
        "Indicate whether a scheduled session is HW accelerated or not.";
    }
  }

  grouping soam-pm-saos-loss-measurement-config-attr {
    description
      "This grouping includes additional attributes that SAOS can take as an
input
      to start a legacy mode loss measurement Performance monitoring session";

    leaf measurement-type {
      type identityref {
        base loss-measurement-type;
      }
      default loss-measurement-type-slm;

      description
```


Copyright© 2022 Ciena® Corporation

```
        "This object specifies what type of Loss Measurement will be
         performed: Frame loss(lmm) or Synthetic Loss(slm).";
    }

    leaf fln-threshold {
      type uint32 {
        range "0..2147483647";
      }
      description
        "Specifies the frame loss near threshold.";
    }

    leaf flf-threshold {
      type uint32 {
        range "0..2147483647";
      }
      description
        "Specifies the frame loss far threshold.";
    }

    leaf seq-threshold {
      type uint32 {
        range "0..65535";
      }
    }

    leaf data-priority {
      type mef-cfm:priority-type;
      description
        "Specifies the priority of data frames";
    }

    leaf drop-eligibility {
      type boolean;
      description
        "SLM message Drop Eligibility Indicator.";
      default false;
    }
  }

  grouping soam-pm-saos-delay-measurement-config-attr {
    description
      "This grouping includes additional attributes that SAOS can take as an
input
       to start a delay measurement Performance monitoring session";

    leaf delay-threshold {
      type yang:gauge32;
      units microseconds;
      description
        "Specifies the Delay threshold in microseconds.";
    }

    leaf jitter-threshold {
      type yang:gauge32;
      units microseconds;
      description
        "Specifies the Jitter threshold in microseconds.";
    }
```


Copyright© 2022 Ciena® Corporation

```
    leaf max-delay-threshold {
      type yang:gauge32;
      units microseconds;
      description
        "Specifies the maximum Delay threshold in microseconds.";
    }

    leaf max-jitter-threshold {
      type yang:gauge32;
      units microseconds;
      description
        "Specifies the maximum Jitter threshold in microseconds.";
    }

    leaf cos-mode {
      type boolean;
    }

    leaf delay-bins {
      type leafref {
      path "/ciena-soam-pm-sds:profile-bins/ciena-soam-pm-sds:delay-profile-
bins/ciena-soam-pm-sds:name";
      }
      description
        "Specifies the Delay profile name.";
    }

    leaf jitter-bins {
      type leafref {
        path "/ciena-soam-pm-sds:profile-bins/ciena-soam-pm-sds:jitter-
profile-bins/ciena-soam-pm-sds:name";
      }
      description
        "Specifies the jitter profile name.";
    }

    leaf forward-jitter-bins {
      type leafref {
        path "/ciena-soam-pm-sds:profile-bins/ciena-soam-pm-sds:jitter-
profile-bins/ciena-soam-pm-sds:name";
      }
      description
        "Specifies the jitter profile name for forward direction.";
    }

    leaf backward-jitter-bins {
      type leafref {
        path "/ciena-soam-pm-sds:profile-bins/ciena-soam-pm-sds:jitter-
profile-bins/ciena-soam-pm-sds:name";
      }
      description
        "Specifies the jitter profile name for backward direction.";
    }
  }

  grouping saos-pm-test-session-identifier {
    uses saos-cfm-service-identifier;
    uses saos-remote-mep-identifier;

    leaf test-id {
```


Copyright© 2022 Ciena® Corporation

```
      type uint32;
      description
        "An integer value that uniquely identifies a PM session
         between a pair of MEP.";
    }

    description
      "It is the values identifying the PM session configured";
  }

//
// Config Data related to ciena proprietary extension of y.1731
//
  container profile-bins {
    list delay-profile-bins {
      key "name";
      leaf name {
        type string;
        description
          "Specifies the name of the delay profile bin and used to identify
the profile.";
      }

      list delay-bins {
        key "bin-index";
        leaf bin-index {
          type uint8 {
            range "1..10";
          }
          description
            "Specifies the bin index value.";
        }

        leaf threshold {
          type yang:gauge32;
          description
           "Specifies the threshold value. The value must be greater than the
value of the preceding bin";
        }
      }
    }

    list jitter-profile-bins {
      key "name";
      leaf name {
        type string;
        description
          "Specifies the name of the jitter profile bin and used to identify
the profile.";
      }

      list jitter-bins {
        key "bin-index";
        leaf bin-index {
          type uint8 {
            range "1..10";
          }
          description
            "Specifies the bin index value.";
        }
```


Copyright© 2022 Ciena® Corporation

```
        leaf threshold {
          type yang:gauge32;
          description
           "Specifies the threshold value. The value must be greater than the
value of the preceding bin";
        }
      }
    }

    list fdr-profile-bins {
      key "name";
      leaf name {
        type string;
        description
          "Specifies the name of the Frame Delay Range profile bin and used
to identify the profile.";
      }

      list fdr-bins {
        key "bin-index";
        leaf bin-index {
          type uint8 {
            range "1..10";
          }
          description
            "Specifies the bin index value.";
        }

        leaf threshold {
          type yang:gauge32;
          description
           "Specifies the threshold value. The value must be greater than the
value of the preceding bin";
        }
      }
    }
  }

//
// RPCs
//
  rpc create-saos-loss-measurement {
    description
      "Schedule a one-way and two-way on-demand and proactive performance
       monitoring loss measurements on a specific MEP in legacy mode.";

    input {
      uses saos-cfm-service-identifier;
      uses saos-remote-mep-identifier;
      uses soam-pm-saos-attr;
      uses soam-pm-saos-loss-measurement-config-attr;
    }
  }

  rpc abort-saos-loss-measurement {
    description
      "Abort a currently running loss measurement session.";

    input {
```


Copyright© 2022 Ciena® Corporation

```
      uses saos-pm-test-session-identifier;

      leaf measurement-type {
        type identityref {
          base loss-measurement-type;
        }
        default loss-measurement-type-slm;

        description
          "This object specifies what type of Loss Measurement will be
           performed: Frame loss(lmm) or Synthetic Loss(slm).";
      }
    }
  }

  rpc delete-saos-loss-measurement {
    description
      "Delete a currently running loss measurement session.";

    input {
      uses saos-pm-test-session-identifier;

      leaf measurement-type {
        type identityref {
          base loss-measurement-type;
        }
        default loss-measurement-type-slm;

        description
          "This object specifies what type of Loss Measurement will be
           performed: Frame loss(lmm) or Synthetic Loss(slm).";
      }
    }
  }

  rpc clear-saos-loss-history-stats {
    description
      "Clear currently running loss measurement session stats and history.";

    input {
      uses saos-cfm-service-identifier;

      leaf measurement-type {
        type identityref {
          base loss-measurement-type;
        }
        default loss-measurement-type-slm;

        description
          "This object specifies what type of Loss Measurement will be
           performed: Frame loss(lmm) or Synthetic Loss(slm).";
      }
    }
  }

  rpc create-saos-loss-responder {
    description
      "Creates a loss measurement responder.";

    input {
```


Copyright© 2022 Ciena® Corporation

```
      uses saos-pm-test-session-identifier;

      leaf measurement-type {
        type identityref {
          base loss-measurement-type;
        }
        default loss-measurement-type-slm;

        description
          "This object specifies what type of Loss Measurement will be
           performed: Frame loss(lmm) or Synthetic Loss(slm).";
      }

      leaf accelerate {
        type boolean;
        description
          "Indicate whether a scheduled session is HW accelerated or not.";
      }

      leaf data-priority {
        type enumeration {
          enum 0 {
              description "Priority to be set to 1.";
          }
          enum 1 {
              description "Priority to be set to 1.";
          }
          enum 2 {
              description "Priority to be set to 1.";
          }
          enum 3 {
              description "Priority to be set to 1.";
          }
          enum 4 {
              description "Priority to be set to 1.";
          }
          enum 5 {
              description "Priority to be set to 1.";
          }
          enum 6 {
              description "Priority to be set to 1.";
          }
          enum 7 {
              description "Priority to be set to 1.";
          }
          enum all {
              description "Priority to be set to 1.";
          }
        }

        description
          "Specifies that priority of data frames";
      }
    }
  }

  rpc delete-saos-loss-responder {
    description
      "Delete a loss measurement responder.";
```


Copyright© 2022 Ciena® Corporation

```
    input {
      uses saos-pm-test-session-identifier;

      leaf measurement-type {
        type identityref {
          base loss-measurement-type;
        }
        default loss-measurement-type-slm;

        description
          "This object specifies what type of Loss Measurement will be
           performed: Frame loss(lmm) or Synthetic Loss(slm).";
      }
    }
  }

  rpc clear-saos-loss-responder {
    description
      "Clear a loss measurement responder.";

    input {
      uses saos-pm-test-session-identifier;

      leaf measurement-type {
        type identityref {
          base loss-measurement-type;
        }
        default loss-measurement-type-slm;

        description
          "This object specifies what type of Loss Measurement will be
           performed: Frame loss(lmm) or Synthetic Loss(slm).";
      }
    }
  }

  rpc create-saos-delay-measurement {
    description
      "Schedule a one-way and two-way on-demand and proactive performance
      monitoring frame delay measurements on a specific MEP in legacy mode.";

    input {
      uses saos-cfm-service-identifier;
      uses saos-remote-mep-identifier;
      uses soam-pm-saos-attr;
      uses soam-pm-saos-delay-measurement-config-attr;
    }
  }

  rpc abort-saos-delay-measurement {
    description
      "Abort a currently running delay measurement session.";

    input {
      uses saos-pm-test-session-identifier;
    }
  }

  rpc delete-saos-delay-measurement {
    description
```


Copyright© 2022 Ciena® Corporation

```
      "Delete a currently running delay measurement session.";

    input {
      uses saos-pm-test-session-identifier;
    }
  }

  rpc clear-saos-delay-history-stats {
    description
      "Clear currently running delay measurement session stats and history.";

    input {
      uses saos-cfm-service-identifier;
    }
  }

  rpc create-saos-delay-responder {
    description
      "Creates a delay measurement responder.";

    input {
      uses saos-pm-test-session-identifier;

      leaf accelerate {
        type boolean;
        description
          "Indicate whether a scheduled session is HW accelerated or not.";
      }
    }
  }

  rpc delete-saos-delay-responder {
    description
      "Delete a delay measurement responder.";

    input {
      uses saos-pm-test-session-identifier;
    }
  }

  rpc clear-saos-delay-responder {
    description
      "Clear a delay measurement responder.";

    input {
      uses saos-pm-test-session-identifier;
    }
  }

  grouping mef-create-augmentations-common-group {
    leaf accelerate {
      type boolean;
      description
        "Indicate whether a scheduled session is offloaded or not.";
      default false;
    }
  }

  // Adding the missing key mandatory parameters in MEF LM RPC
  augment /mef-soam-pm:create-loss-measurement/mef-soam-pm:input {
```


Copyright© 2022 Ciena® Corporation

```
    leaf maintenance-domain-id {
       type leafref {
         path "/mef-cfm:maintenance-domain/mef-cfm:id";
       }
       description
         "The ID of a Maintenance Domain of the MEP on which the loss
          measurement should be created";
    }

    leaf maintenance-association-id {
       type leafref {
         path "/mef-cfm:maintenance-domain/mef-cfm:maintenance-association/
mef-cfm:id";
       }
       description
         "The ID of a Maintenance Association of the MEP on which the
          loss measurement should be created";
    }

    leaf local-mep-id {
       type leafref {
         path "/mef-cfm:maintenance-domain/mef-cfm:maintenance-association/
mef-cfm:maintenance-association-end-point/mef-cfm:mep-identifier";
       }
       description
        "The MEPID of the MEP on which the loss measurement should be created";
    }
  }

  // Adding the missing key mandatory parameters in MEF DM RPC
  augment /mef-soam-pm:create-delay-measurement/mef-soam-pm:input {
    leaf maintenance-domain-id {
       type leafref {
         path "/mef-cfm:maintenance-domain/mef-cfm:id";
       }
       description
         "The ID of a Maintenance Domain of the MEP on which the delay
          measurement should be created";
    }

    leaf maintenance-association-id {
       type leafref {
         path "/mef-cfm:maintenance-domain/mef-cfm:maintenance-association/
mef-cfm:id";
       }
       description
         "The ID of a Maintenance Association of the MEP on which the
          delay measurement should be created";
    }

    leaf local-mep-id {
       type leafref {
         path "/mef-cfm:maintenance-domain/mef-cfm:maintenance-association/
mef-cfm:maintenance-association-end-point/mef-cfm:mep-identifier";
       }
       description
         "The MEPID of the MEP on which the delay measurement should be
          created";
    }
  }
```


Copyright© 2022 Ciena® Corporation

```
  augment /mef-soam-pm:create-loss-measurement/mef-soam-pm:input {
    uses mef-create-augmentations-common-group;
  }

  augment /mef-soam-pm:create-delay-measurement/mef-soam-pm:input {
    uses mef-create-augmentations-common-group;

    leaf cos-mode {
      type boolean;
      description
       "If true then indicates, substitute DMR p-bits for test-id when TestId
TLV is missing.";
      default false;
    }
  }

  grouping delay-bins-configuration-group {
    choice delay-bins-configuration {
      description
        "Choice to Specify the lower boundaries for a measurement bins. The
upper boundary is defined by the bin value or infinite for the last bin.";

      case bins-profile {
            leaf name {
              type string;
              description
                "Specifies the name of the profile bin.";
            }
          }
      case bins-lower-bounds {
        list bins {
          key "bin-index";
          leaf bin-index {
            type uint8 {
              range "2..10";
            }
            description
              "Specifies the bin index value. The first bin boundary (number
set to 1) always contains the value of 0. Attempting to write a non-zero value
to bin-index=1 will result in an error.";
          }

          leaf lower-bound {
            type yang:gauge32;
            description
             "Specifies the lower-bound value. The value must be greater than
the value of the preceding bin.";
          }
        }
      }
    }
  }

  augment /mef-soam-pm:create-delay-measurement/mef-soam-pm:input {
    container delay-measurement-bins {
      description
        "This container defines parameters to specify the lower bounds of
individual bins for number of Frame Delay, Inter-Frame Delay Variation and
Frame Delay Range bins.";
```


Copyright© 2022 Ciena® Corporation

```
      container fd {
        description
          "This container defines parameters to specify the lower bounds of
individual bins for number of Frame Delay bins.";

        uses delay-bins-configuration-group;
      }
      container fdr {
        description
          "This container specifies the lower bounds of individual bins for
number of Frame Delay Range bins.";

         uses delay-bins-configuration-group;
      }
      container ifdv {
        description
          "This container specifies the lower bounds of individual bins for
number of Inter-Frame Delay Variation bins.";

         uses delay-bins-configuration-group;
      }
      container forward-ifdv {
        description
          "This container specifies the lower bounds of individual bins for
number of Inter-Frame Delay Variation bins used to store the forward jitter
results.";

         uses delay-bins-configuration-group;
      }
      container backward-ifdv {
        description
          "This container specifies the lower bounds of individual bins for
number of Inter-Frame Delay Variation bins used to store the backward jitter
results.";

         uses delay-bins-configuration-group;
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-software-licensing.yang

```
module ciena-software-licensing {
  namespace
    "http://www.ciena.com/ns/yang/ciena-software-licensing";

  prefix "license";

  import ciena-types {
    prefix ciena-types;
  }

  import ciena-file-transfer-types {
    prefix ciena-ftt;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module defines Ciena's license management options.

     Copyright (c) 2017 Ciena Corporation.  All rights
     reserved.

     All information contained herein is, and remains
     the property of Ciena Corporation. Dissemination of this
     information or reproduction of this material is strictly
     forbidden unless prior written permission is obtained from
     Ciena Corporation.";

  revision 2018-02-08 {
    description
      "Added list for license feature operational data in container software-
license-state.";
    reference
      "Ciena Functional Specification.";
   }

  revision 2017-02-16 {
    description
      "Initial version";
    reference
      "39XX 51XX SAOS 6.16 Software Licensing & Management Standard
       Revision A

       8700 Packetwave Platform Software Management and Licensing,
       Standard Revision A";
  }

  /*
```


Copyright© 2022 Ciena® Corporation

```
 * typedefs
 */

typedef license-key-type {
  type string {
    length "14";
  }
  description
    "Software license product key.";
}

typedef license-module-type {
  type uint16 {
    range "1..3";
  }
  description
    "The module number that is associated with a software license.";
}

typedef license-feature-status-type {
  type enumeration {
    enum disabled {
      description
        "License feature is disabled.";
    }
    enum violation {
      description
        "License feature is in violation.";
    }
    enum fully-active {
      description
        "License feature is fully active.";
    }
    enum some-licence-key-missing {
      description
        "Licence feature is partial.";
    }
    enum unknown {
      description
        "License feature status is unknown.";
    }
  }
  description
    "License features statuses.";
}

typedef license-install-status-type {
  type enumeration {
    enum installed {
      description
        "The license is installed.";
    }
    enum not-installed {
      description
        "The license is not installed.";
    }
    enum partially-installed {
      description
        "The license is partially installed.";
    }
```


Copyright© 2022 Ciena® Corporation

```
      enum unknown {
        description
          "The license status is unknown.";
      }
    }
    description
      "License installed statuses.";
  }

  typedef software-license-file-transfer-status {
    type enumeration {
      enum idle {
        description
          "No files are being downloaded at this time.";
      }
      enum busy {
        description
          "A file is currently being downloaded.";
      }
    }
    description
      "Configuration XFTP status values.";
  }

  /*
   * Groupings
   */

  /*
   * Identities
   */

  /*
   * Features
   */

  /*
   * Configuration data nodes
   */
  container software-license {
    description
      "SAOS license options.";

    list license {
      key key;
      description
        "List of installed licenses.";

      leaf key {
        type license-key-type;
        description
          "Software license product key.";
      }

      leaf module {
        type license-module-type;
        description
          "The module that this license-key is applied to.";
      }
    }
```


Copyright© 2022 Ciena® Corporation

```
}

/*
 * Operational state data nodes
 */
container software-license-state {
  config false;
  description
    "SAOS license operational data.";

  leaf file-transfer-status {
    type software-license-file-transfer-status;
    description
      "Status of the latest download request coming from NETCONF
       when downloading a license using an RPC. This does not
       apply to files transferred using the CLI or SNMP.";
  }

  list features {
    key name;
    description
       "Feature operational information.";

    leaf name {
       type string;
       description
          "Feature name.";
    }

    leaf state {
       type license-feature-status-type;
       description
          "Operational status of feature.";
    }
  }

  list licenses {
    key key;
    description
      "Installed license key information.";

    leaf key {
      type license-key-type;
      description
        "Software license product key.";
    }

    leaf domain {
      type string;
      description
        "License domain.";
    }

    leaf name {
      type string;
      description
        "License name.";
    }

    leaf sequence-number {
```


Copyright© 2022 Ciena® Corporation

```
        type uint32;
        description
          "License sequence number.";
      }
    }

    list installed-licenses {
      key "name domain";
      description
        "List of installed licenses.";

      leaf name {
        type string;
        description
          "Name of the licensed SAOS feature.";
      }

      leaf domain {
        type string;
        description
          "Domain of the licensed SAOS feature.";
      }

      leaf status {
        type license-install-status-type;
        description
          "Status of the installed license.";
      }

      leaf state {
        type ciena-types:oper-state;
        description
          "Operational state of the license.";
      }
    }
  }

  /*
   * RPCs
   */

  rpc license-file-install {
    description
      "Download and install a new license file.";

    input {
      uses ciena-ftt:file-transfer-parameters;
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-software-mgmt.yang

```
module ciena-software-mgmt {
  namespace "http://www.ciena.com/ns/yang/ciena-software-mgmt";
  prefix ciena-software-mgmt;

  import ciena-types {
    prefix ciena;
  }

  import ciena-file-transfer-types {
    prefix ciena-ftt;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module defines the nodes for
     Software Licensing and Upgrade Management
     for the NETCONF protocol.

     Copyright (c) 2017 Ciena Corporation.  All rights reserved.

     All information contained herein is, and remains the property of
     Ciena Corporation. Dissemination of this information or
     reproduction of this material is strictly forbidden unless prior
     written permission is obtained from Ciena Corporation.";

  revision 2017-12-19 {
    description
      "Initial version";
    reference
      "39XX/51XX Service Delivery, Aggregation and Virtualization
       Switches, Software Management and Licensing (009-3273-018)
       (Ciena Corporation, December 2016);
       RFC 6020: YANG - A Data Modeling Language for the Network
       Configuration Protocol (NETCONF).";
  }

  feature rolling-upgrade {
    description
      "On systems with multiple line modules, this feature allows groups
       of one or more line modules to be upgraded.  Each group is
       upgraded in sequence.  This avoids having all the the line
       modules out of service at one time.";
  }

  typedef software-status {
    type enumeration {
      enum good {
```

```
        description
          "Good";
      }
      enum valid {
        description
          "Valid";
      }
      enum invalid {
        description
          "Invalid";
      }
      enum not-validated {
        description
          "Not validated";
      }
      enum empty {
        description
          "Empty";
      }
      enum running {
        description
          "Running";
      }
      enum burning {
        description
          "Burning";
      }
      enum syncing {
        description
          "Syncing";
      }
      enum loading {
        description
          "Loading";
      }
      enum waiting {
        description
          "Waiting";
      }
      enum upgrading {
        description
          "Upgrading";
      }
      enum idle {
        description
          "Idle";
      }
    }
    description
      "Software state";
  }

  typedef rpc-status {
    type enumeration {
      enum idle {
        description
          "No NETCONF software operation is currently in progress.";
      }
      enum busy {
        description
```


Copyright© 2022 Ciena® Corporation

```
          "A NETCONF software operation is currently in progress.";
      }
    }
    description
      "Status values for a software operation intiated through a
       NETCONF RPC.";
  }

  container software {
    description
      "Software configuration";

    container rolling-upgrade {
      if-feature rolling-upgrade;
      description
        "Rolling upgrade configuration";

      leaf admin-state {
        type ciena:admin-state;
        description
          "The rolling upgrade administrative state.";
      }

      list group-table {
        key slot;
        description
          "Table of slots and rolling upgrade groups to which they
           belong.";

        leaf slot {
          type string;
          description
            "Slot name";
        }

        leaf group {
          type uint8 {
            range 1..255;
          }
          description
            "The group number of the rolling upgrade group to which this
             slot belongs.";
        }
      }
    }
  }

  container software-state {
    config false;
    description
      "Software operational state.  On success, returns the contained
       data.  On failure, returns an error message describing the reason
       for the failure.";

    leaf operation {
      type software-status;
      description
        "The software management operation currently being performed
         (SAS only)";
      }
```


Copyright© 2022 Ciena® Corporation

```
leaf rpc-status {
  type rpc-status;
  description
    "Status of the latest software operation initiated through
     a NETCONF RPC. This does not apply to software operations
     initiated by other sources.";
}

leaf auto-revert-timer {
  type uint32;
  units seconds;
  description
    "Auto-revert timer  (SAS only)";
}

leaf user-revert-timer {
  type uint32;
  units seconds;
  description
    "User-revert timer  (SAS only)";
}

container running-package {
  description
    "Running software package information";

  leaf package-version {
    type string;
    description
      "Package version";
  }

  leaf package-info {
    type string;
    description
      "Additional package information";
  }

  leaf package-status {
    type string;
    description
      "Status of the running software package";
  }

  leaf release-status {
    type string;
    description
      "Release status of the running software package";
  }

  list component {
    key component-name;

    leaf component-name {
      type string;
      description
        "Component name";
    }
```


Copyright© 2022 Ciena® Corporation

```
    leaf version {
      type string;
      description
        "Component version";
    }

    description
      "Software component information";
  }

  container active-bootchain {
    leaf version {
      type string;
      description
        "Version";
    }
    leaf status {
      type software-status;
      description
        "Status";
    }
    description
      "Active bootchain software information";
  }

  container standby-bootchain {
    leaf version {
      type string;
      description
        "Version";
    }
    leaf status {
      type software-status;
      description
        "Status";
    }
    description
      "Standby bootchain software information";
  }

  list release-package {
    key location;

    leaf location {
      type string;
      description
        "Location";
    }

    leaf version {
      type string;
      description
        "Version";
    }

    leaf status {
      type software-status;
      description
        "Status";
    }
```


Copyright© 2022 Ciena® Corporation

```
    leaf validation-status {
      type string;
      description
        "Validation Status";
    }

    container bootloader {
      leaf version {
        type string;
        description
          "Version";
      }

      leaf status {
        type software-status;
        description
          "Status";
      }

      description
        "Bootloader";
    }

    description
      "Release package software information";
  }

  list kernel {
    key location;

    leaf location {
      type string;
      description
        "Location";
    }

    leaf version {
      type string;
      description
        "Version";
    }

    leaf timestamp {
      type string;
      description
        "Timestamp";
    }

    description
      "Kernel software information";
  }

  list uboot {
    key location;

    leaf location {
      type string;
      description
        "Location:  UBOOT bank 0, 1 or gold";
```


Copyright© 2022 Ciena® Corporation

```
      }

      leaf version {
        type string;
        description
          "Version";
      }

      leaf timestamp {
        type string;
        description
          "Timestamp";
      }

      description
        "UBOOT software information";
    }
  }

  list package {
    key module;

    leaf module {
      type string;
      description
        "Module identifier";
    }

    leaf package-version {
      type string;
      description
        "Software package version";
    }

    leaf package-status {
      type string;
      description
        "Software package status";
    }

    container active-bootchain {
      leaf version {
        type string;
        description
          "Version";
      }

      leaf status {
        type software-status;
        description
          "Status";
      }

      description
        "Active bootchain software information";
    }

    container standby-bootchain {
      leaf version {
        type string;
```


Copyright© 2022 Ciena® Corporation

```
        description
          "Version";
      }

      leaf status {
        type software-status;
        description
          "Status";
      }

      description
        "Standby bootchain software information";
    }

    list release-package {
      key location;

      leaf location {
        type string;
        description
          "Software package location.  For SDS the locations are
           banks A and B.  For SAS, the locations are partitions
           0, 1 or 2.";
      }

      leaf version {
        type string;
        description
          "Software package version";
      }

      leaf status {
        type software-status;
        description
          "Software package status.";
      }

      container bootloader {
        leaf version {
          type string;
          description
            "Version";
        }

        leaf status {
          type software-status;
          description
            "Status";
        }

        description
          "Bootloader";
      }

      description
        "Software release package information for this location.";
    }

    list kernel {
      key location;
```


Copyright© 2022 Ciena® Corporation

```
        leaf location {
          type string;
          description
            "Location";
        }

        leaf version {
          type string;
          description
            "Version";
        }

        leaf timestamp {
          type string;
          description
            "Timestamp";
        }
        description
          "Kernel software information";
      }

      list uboot {
        key location;

        leaf location {
          type string;
          description
            "Location:  UBOOT bank 0, 1 or gold";
        }

        leaf version {
          type string;
          description
            "Version";
        }

        leaf timestamp {
          type string;
          description
            "Timestamp";
        }

        description
          "UBOOT software information";
      }

      description
        "Software information for this processor";
    }
  }

  rpc software-activate {
    description
      "Activate the software installed in the release directory.  On
       success, returns <ok/>.  On failure, returns an <rpc-error>
       containing an <error-message> describing the reason for the
       failure.";

    input {
```


Copyright© 2022 Ciena® Corporation

```
      leaf package {
        type string;
        description
          "Package name.";
      }

      leaf allow-service-disruption {
        type boolean;
        description
          "Determines whether or not this software activation is allowed
           to disrupt service.";
      }

      leaf force-standby {
        type boolean;
        description
          "When true, forces the activation of the standby software
           bank.";
      }

      leaf revert-timeout {
        type uint32;
        units seconds;
        description
          "The time interval between the completion of the software
           activate command and the automatic reversion of the software
           to the previous running version.  The user must issue the
           software-cancel revert-timer command before the end of this
           interval to prevent the reversion of the software.";
      }
    }
  }

  rpc software-cancel {
    description
      "Cancel the specified software operation.  On success, returns
       <ok/>.  On failure, returns an <rpc-error> containing an
       <error-message> describing the reason for the failure.  (SAS
       only)";

    input {
      leaf operation {
        type enumeration {
          enum revert-timer {
            description
              "Cancel the revert timer.";
          }
        }
        description
          "The software operation to be canceled.";
      }
    }
  }

  rpc software-download {
    description
      "Download a software release package to a specified destination.
       On success, returns <ok/>.  On failure, returns an <rpc-error>
       containing an <error-message> describing the reason for the
       failure.";
```


Copyright© 2022 Ciena® Corporation

```
    input {
      uses ciena-ftt:file-transfer-parameters;

      leaf destination-path {
        type string;
        description
          "The destination path.";
      }
    }
  }

  rpc software-install {
    description
      "Install the new software package into flash.  On success, returns
       <ok/>.  On failure, returns an <rpc-error> containing an
       <error-message> describing the reason for the failure.";

    input {
      leaf package {
        type string;
        description
          "The package release name.";
      }

      leaf package-path {
        type string;
        description
          "The path to the package files.";
      }

      uses ciena-ftt:file-transfer-parameters;

      leaf defer-activation {
        type boolean;
        description
          "When true, activation is deferred until the next reboot.";
      }
    }
  }

  rpc software-protect {
    description
      "Back up the new software image.  On success, returns <ok/>.  On
       failure, returns an <rpc-error> containing an <error-message>
       describing the reason for the failure.  (SAS only)";
  }

  rpc software-remove {
    description
      "Remove a specific release package from the system.  On success,
       returns <ok/>.  On failure, returns an <rpc-error> containing an
       <error-message> describing the reason for the failure.  (SAS
       only)";

    input {
      leaf package {
        type string;
        description
          "The package release name or partition number.";
```


Copyright© 2022 Ciena® Corporation

```
      }
    }
  }

  rpc software-validate {
    description
      "Validate the software.  The input may include an optional
       package-path (SAS only).  On success, returns <ok/>.  On failure,
       returns an <rpc-error> containing an <error-message> describing
       the reason for the failure.";

    input {
      leaf package-path {
        type string;
        description
          "The path to the package files.  When empty or not present,
           validate the currently running version (which is the same as
           specifying the system-software attribute in SAS).";
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-ssh-types.yang

```
module ciena-ssh-types {
  namespace "http://ciena.com/ns/yang/ciena-ssh-types";

  prefix sshtypes;

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This module contains a collection of YANG type definitions for
     SSH.

     This module contains the YANG definition of SSH.

     Copyright (c) 2016 Ciena Corporation.  All rights reserved.

     All information contained herein is, and remains
     the property of Ciena Corporation. Dissemination of this
     information or reproduction of this material is strictly
     forbidden unless prior written permission is obtained from
     Ciena Corporation.";

  revision 2017-08-21 {
    description
      "Added the x509 public key authentication types.";
    reference
      "";
  }

  revision 2016-03-01 {
    description
      "Initial revision. Defines data types used by Ciena
       YANG models.";
    reference
      "";
  }

  typedef kex-algorithm {
    type enumeration {
      enum curve25519-sha256@libssh.org {
        description
          "The curve25519-sha256@libssh.org algorithm.";
      }
      enum ecdh-sha2-nistp256 {
        description
          "The ecdh-sha2-nistp256 algorithm.";
      }
      enum ecdh-sha2-nistp384 {
        description
```

```
          "The ecdh-sha2-nistp384 algorithm.";
      }
      enum ecdh-sha2-nistp521 {
        description
          "The ecdh-sha2-nistp521 algorithm.";
      }
      enum diffie-hellman-group-exchange-sha256 {
        description
          "The diffie-hellman-group-exchange-sha256 algorithm.";
      }
      enum diffie-hellman-group-exchange-sha1 {
        description
          "The diffie-hellman-group-exchange-sha1 algorithm.";
      }
      enum diffie-hellman-group14-sha1 {
        description
          "The diffie-hellman-group14-sha1 algorithm.";
      }
      enum diffie-hellman-group1-sha1 {
        description
          "The diffie-hellman-group1-sha1 algorithm.";
      }
    }
    description
      "List of kex algorithms.";
  }

  typedef encryption-algorithm {
    type enumeration {
      enum aes128-ctr {
        description
          "The aes128-ctr algorithm.";
      }
      enum aes256-cbc {
        description
          "The aes256-cbc algorithm.";
      }
      enum aes192-ctr {
        description
          "The aes192-ctr algorithm.";
      }
      enum aes256-ctr {
        description
          "The aes256-ctr algorithm.";
      }
      enum arcfour256 {
        description
          "The arcfour256 algorithm.";
      }
      enum arcfour128 {
        description
          "The arcfour128 algorithm.";
      }
      enum aes128-gcm@openssh.com {
        description
          "The aes128-gcm@openssh.com algorithm.";
      }
      enum aes256-gcm@openssh.com {
        description
          "The aes256-gcm@openssh.com algorithm.";
```


Copyright© 2022 Ciena® Corporation

```
      }
      enum chacha20-poly1305@openssh.com {
        description
          "The chacha20-poly1305@openssh.com algorithm.";
      }
      enum aes128-cbc {
        description
          "The aes128-cbc algorithm.";
      }
      enum 3des-cbc {
        description
          "The 3des-cbc algorithm.";
      }
      enum blowfish-cbc {
        description
          "The blowfish-cbc algorithm.";
      }
      enum cast128-cbc {
        description
          "The cast128-cbc algorithm.";
      }
      enum aes192-cbc {
        description
          "The aes192-cbc algorithm.";
      }
      enum arcfour {
        description
          "The arcfour algorithm.";
      }
      enum rijndael-cbc@lysator.liu.se {
        description
          "The rijndael-cbc@lysator.liu.se algorithm.";
      }
    }
    description
      "List of encryption algorithms.";
  }

  typedef mac-algorithm {
    type enumeration {
      enum hmac-md5-etm@openssh.com {
        description
          "The hmac-md5-etm@openssh.com algorithm.";
      }
      enum hmac-sha1-etm@openssh.com {
        description
          "The hmac-sha1-etm@openssh.com algorithm.";
      }
      enum umac-64-etm@openssh.com {
        description
          "The umac-64-etm@openssh.com algorithm.";
      }
      enum umac-128-etm@openssh.com {
        description
          "The umac-128-etm@openssh.com algorithm.";
      }
      enum hmac-sha2-256-etm@openssh.com {
        description
          "The hmac-sha2-256-etm@openssh.com algorithm.";
      }
```


Copyright© 2022 Ciena® Corporation

```
      enum hmac-sha2-512-etm@openssh.com {
        description
          "The hmac-sha2-512-etm@openssh.com algorithm.";
      }
      enum hmac-ripemd160-etm@openssh.com {
        description
          "The hmac-ripemd160-etm@openssh.com algorithm.";
      }
      enum hmac-sha1-96-etm@openssh.com {
        description
          "The hmac-sha1-96-etm@openssh.com algorithm.";
      }
      enum hmac-md5-96-etm@openssh.com {
        description
          "The hmac-md5-96-etm@openssh.com algorithm.";
      }
      enum hmac-md5 {
        description
          "The hmac-md5 algorithm.";
      }
      enum hmac-sha1 {
        description
          "The hmac-sha1 algorithm.";
      }
      enum umac-64@openssh.com {
        description
          "The umac-64@openssh.com algorithm.";
      }
      enum umac-128@openssh.com {
        description
          "The umac-128@openssh.com algorithm.";
      }
      enum hmac-sha2-256 {
        description
          "The hmac-sha2-256 algorithm.";
      }
      enum hmac-sha2-512 {
        description
          "The hmac-sha2-512 algorithm.";
      }
      enum hmac-ripemd160 {
        description
          "The hmac-ripemd160 algorithm.";
      }
      enum hmac-ripemd160@openssh.com {
        description
          "The hmac-ripemd160@openssh.com algorithm.";
      }
      enum hmac-sha1-96 {
        description
          "The hmac-sha1-96 algorithm.";
      }
      enum hmac-md5-96 {
        description
          "The hmac-md5-96 algorithm.";
      }
    }
    description
      "List of mac algorithms.";
  }
```


Copyright© 2022 Ciena® Corporation

```
  typedef pka-algorithm {
    type enumeration {
      enum ssh-dss {
        description
          "The ssh-dss algorithm.";
      }
      enum ssh-rsa {
        description
          "The ssh-rsa algorithm.";
      }
      enum ssh-ed25519 {
        description
          "The ssh-ed25519 algorithm.";
      }
      enum ecdsa-sha2-nistp256 {
        description
          "The ecdsa-sha2-nistp256 algorithm.";
      }
      enum ecdsa-sha2-nistp384 {
        description
          "The ecdsa-sha2-nistp384 algorithm.";
      }
      enum ecdsa-sha2-nistp521 {
        description
          "The ecdsa-sha2-nistp521 algorithm.";
      }
      enum x509v3-sign-rsa {
        description
          "The x509v3-sign-rsa algorithm.";
      }
      enum x509v3-sign-dss {
        description
          "The x509v3-sign-dss algorithm.";
      }
      enum x509v3-ecdsa-sha2-nistp256 {
        description
          "The x509v3-ecdsa-sha2-nistp256 algorithm.";
      }
      enum x509v3-ecdsa-sha2-nistp384 {
        description
          "The x509v3-ecdsa-sha2-nistp384 algorithm.";
      }
      enum x509v3-ecdsa-sha2-nistp521 {
        description
          "The x509v3-ecdsa-sha2-nistp521 algorithm.";
      }
      enum rsa-sha2-256 {
        description
          "The rsa-sha2-256 algorithm.";
      }
      enum rsa-sha2-512 {
        description
          "The rsa-sha2-512 algorithm.";
      }
    }
    description
      "List of public key authentication algorithms.";
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-ssh.yang

```
module ciena-ssh {
  namespace "http://www.ciena.com/ns/yang/ciena-ssh";

  prefix "ciena-ssh";

  import ietf-inet-types {
    prefix inet;
  }

  import ciena-ssh-types {
    prefix sshtype;
  }

  import ciena-types {
    prefix ciena;
  }

  import ciena-ocsp {
    prefix ciena-ocsp;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This module contains the YANG definition of SSH.

     Copyright (c) 2016 Ciena Corporation.  All rights reserved.

     All information contained herein is, and remains
     the property of Ciena Corporation. Dissemination of this
     information or reproduction of this material is strictly
     forbidden unless prior written permission is obtained from
     Ciena Corporation.";

  revision "2017-08-21" {
    description
      "Added the auth-policy leaf, x509-host-key leaf,
       OCSP container and leafs, system-client container and leaf.
       Corrected the range of client-alive-interval.";
    reference
      "";
  }

  revision "2016-08-25" {
    description
      "Initial version";
    reference
      "";
```

```
}

grouping algorithm-group {
  description
    "Holds the leaf nodes used for configuration of
     ssh algorithms.";
  leaf priority {
    type uint8;
    description
      "This leaf specifies the priority
       level of the algorithm.";
  }
  leaf admin-state {
    type ciena:admin-state;
    description
      "This leaf specifies the administrative
       state of the key.";
  }
}

grouping kex-group {
  description
    "Kex algorithm grouping.";
  list kex {
    key algorithm;
    description
      "This list holds an instance of each kex key";
    leaf algorithm {
      type sshtype:kex-algorithm;
      description
        "This leaf specifies the kex algorithm
         name.";
    }
    uses algorithm-group;
  }
}

grouping kex-state-group {
  description
    "Kex algorithm state grouping.";
  list kex {
    key algorithm;
    description
      "This list holds an instance of each kex key";

    leaf algorithm {
      type sshtype:kex-algorithm;
      description
        "This leaf specifies the kex algorithm
         name.";
    }

    leaf oper-state {
      type ciena:admin-state;
      description
        "This leaf specifies the operational
         state of the key.";
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

```
  grouping encryption-group {
    description
      "Encryption algorithm grouping.";
    list encryption {
      key algorithm;
      description
        "This list holds an instance of each encryption
         key and its corresponding configuration.";

      leaf algorithm {
        type sshtype:encryption-algorithm;
        description
          "This leaf specifies the encryption algorithm
           name.";
      }
      uses algorithm-group;
    }
  }

  grouping encryption-state-group {
    description
      "Encryption algorithm state grouping.";
    list encryption {
      key algorithm;
      description
        "This list holds an instance of each encryption
         key and its corresponding state.";

      leaf algorithm {
        type sshtype:encryption-algorithm;
        description
          "This leaf specifies the encryption algorithm
           name.";
      }

      leaf oper-state {
        type ciena:admin-state;
        description
          "This leaf specifies the operational
           state of the key.";
      }
    }
  }

  grouping mac-group {
    description
      "Mac algorithm grouping.";
    list mac {
      key algorithm;
      description
        "This list holds an instance of each mac key
         and its corresponding configuration.";

      leaf algorithm {
        type sshtype:mac-algorithm;
        description
          "This leaf specifies the mac algorithm name.";
      }
      uses algorithm-group;
```


Copyright© 2022 Ciena® Corporation

```
    }
  }

  grouping mac-state-group {
    description
      "Mac algorithm state grouping.";
    list mac {
      key algorithm;
      description
        "This list holds an instance of each mac key
         and its corresponding state.";

      leaf algorithm {
        type sshtype:mac-algorithm;
        description
          "This leaf specifies the mac algorithm name.";
      }

      leaf oper-state {
        type ciena:admin-state;
        description
          "This leaf specifies the operational
           state of the key.";
      }
    }
  }

  grouping pka-group {
    description
      "Pka algorithm grouping.";
    list pka {
      key algorithm;
      description
        "This list holds an instance of each pka key
         and its corresponding configuration.";

      leaf algorithm {
        type sshtype:pka-algorithm;
        description
          "This leaf specifies the pka algorithm name.";
      }

      leaf admin-state {
        type ciena:admin-state;
        description
          "This leaf specifies the administrative
           state of the key.";
      }
    }
  }

  grouping pka-state-group {
    description
      "Pka algorithm state grouping.";
    list pka {
      key algorithm;
      description
        "This list holds an instance of each pka key
         and its corresponding state.";
```


Copyright© 2022 Ciena® Corporation

```
      leaf algorithm {
        type sshtype:pka-algorithm;
        description
          "This leaf specifies the pka algorithm name.";
      }

      leaf oper-state {
        type ciena:admin-state;
        description
          "This leaf specifies the operational
           state of the key.";
      }
    }
  }

  grouping host-key-authentication-group {
    description
      "Host-key-authentication algorithm state grouping.";
    list host-key-authentication {
      key algorithm;
      description
        "This list holds an instance of each host authentication key
         and its corresponding state.";

      leaf algorithm {
        type sshtype:pka-algorithm;
        description
          "This leaf specifies the host-key-authentication algorithm name.";
      }
      leaf priority {
        type uint8;
        description
          "This leaf specifies the priority
           level of the algorithm.";
      }

      leaf admin-state {
        type ciena:admin-state;
        description
          "This leaf specifies the operational
           state of the key.";
      }
    }
  }

  /*
   * Configuration data nodes
   */
  container ssh {
    description
      "This container holds all SSH server/client
       configuration.";

    // Configuration data
    container configuration {
      description
        "Holds all of the SSH configuration data.";

      container server {
        description
```


Copyright© 2022 Ciena® Corporation

```
    "SSH server configuration data.";

  leaf global-admin-state {
    type ciena:admin-state;
    description
      "This leaf specifies the global admin state.";
  }

  leaf-list client {
    type union {
      type inet:host;
      type inet:ip-prefix;
    }
    description
      "List of allowed SSH clients.
       Supports hostname, ip-address, or ip-address/prefix";
  }

  container algorithm {
    description
      "This container holds all ssh client
       algorithms along with the respective
       configuration.";
    uses kex-group;
    uses encryption-group;
    uses mac-group;
    uses pka-group;
  }

  container attributes {
    description
      "This container holds all configurable ssh
       attributes.";

    leaf listener-address-groups {
      type enumeration {
        enum all {
          description
            "Search for ssh connections through all
             interfaces.";
        }
        enum mgmt-interfaces {
          description
            "Search for ssh connections through
             management interfaces.";
        }
      }
      description
        "This leaf contains the address group
         that listens for ssh connections.";
    }

    leaf listener-port {
      type uint16 {
        range "22 .. 65535";
      }
      description
        "This leaf identifies the ssh port number
         that will listen for ssh connections.";
    }
```


Copyright© 2022 Ciena® Corporation

```
leaf authentication-retries {
  type uint8 {
    range "1 .. 3";
  }
  description
    "This leaf specifies the max allowable
     number of ssh connection retries.";
}

leaf client-alive-interval {
  type uint32 {
    range "0 .. 2147483646";
  }
  description
    "This leaf specifies the client alive interval";
}

leaf client-alive-count {
  type uint8 {
    range "0 .. 10";
  }
  description
    "This leaf specifies the number of client alive
     messages";
}

leaf login-grace-time {
  type uint32 {
    range "0 .. 2147483647";
  }
  description
    "This leaf specifies the login grace timeout";
}

leaf max-shared-sessions {
  type uint8 {
    range "1 .. 10";
  }
  description
    "This leaf sets the max sessions per network
     connection.";
}

leaf multi-factor-auth {
  type ciena:on-off-state;
  description
    "This leaf sets ssh multi-factor authentication
     on or off.";
}

leaf authentication-policy {
  type enumeration {
    enum public-key {
      description
        "Public ssh key.";
    }
    enum all {
      description
        "All authentication types.";
```


Copyright© 2022 Ciena® Corporation

```
      }
    }
    description
      "This leaf sets ssh authentication-policy.";
  }

  leaf rekey-limit {
    type enumeration {
      enum 1G {
        description
          "Transmit 1G bytes.";
      }
      enum 2G {
        description
          "Transmit 2G bytes.";
      }
      enum 4G {
        description
          "Transmit 4G bytes.";
      }
      enum 500M {
        description
          "Transmit 500M bytes.";
      }
      enum default {
        description
          "Transmit default bytes.";
      }
    }
    description
      "This leaf specifies the number of bytes that are
       transmitted before the session key is renegotiated.";
  }

  leaf rekey-timeout {
    type uint32;
    description
      "This leaf specifies the max time (in seconds) that
       may pass before the session key is renegotiated.";
  }

  leaf strict-modes {
    type ciena:on-off-state;
    description
      "This leaf sets ssh strict-modes.";
  }

  leaf tcp-forwarding {
    type ciena:on-off-state;
    description
      "This leaf enables/disables tcp-forwarding.";
  }

  leaf tcp-keepalive {
    type ciena:on-off-state;
    description
      "This leaf sets the tcp-keepalive value.";
  }

  leaf x509-host-key {
```


Copyright© 2022 Ciena® Corporation

```
            type enumeration {
              enum disabled {
                description
                 "Specifies that the SSH server shall present only non-X.509
host keys.";
              }
              enum enabled {
                description
                 "Specifies that the SSH server may present x509 or non-x509
host-keys.";
              }
              enum only {
                description
                "Specifies that the SSH server shall present only x509 based
host-keys.";

              }
            }
            description
              "Specifies whether the ssh server should present
               non-x509 host-keys, x509 host-keys, or both.";
          }
        }
        container ocsp {
          description
           "Configurable parameters for the management of Online Certificate
Status Protocol (OCSP).";
          uses ciena-ocsp:ocsp;

        }
      }

      container client {
        description
          "SSH client configuration data.";

        container algorithm {
          description
            "This container holds all ssh client
             algorithms along with the respective
             configuration.";
          uses kex-group;
          uses encryption-group;
          uses mac-group;
          uses host-key-authentication-group;
        }

        container ocsp {
          description
            "Holds all of the OCSP configuration data.";
          uses ciena-ocsp:ocsp;

        }
        container system-client {
          leaf host-key-checking {
            type enumeration {
              enum ignore {
                description
                  "Disables host-key checking for SSH system client
                   connections.";
```


Copyright© 2022 Ciena® Corporation

```
            }
            enum non-strict {
              description
                "Enables non-strict non-interactive host-key
                 checking for SSH system client connections.";
            }
            enum strict {
              description
                "Enables strict non-interactive host-key
                 checking for SSH system client connections.";
            }
          }

          description
            "Contains the system managed client attributes.";
        }
      }
    }

  }

  // Operational data
  container state {
    config false;
    description
      "Holds all of the SSH operational data.";

    leaf global-oper-state {
      type ciena:oper-state;
      description
        "This leaf specifies the global operational state.";
    }

    container server {
      description
        "SSH server operational data.";

      leaf-list resolved-client-address {
        type inet:host;
        description
          "List of allowed SSH client resolved addresses.";
      }

      container key {
        description
          "SSH server key information.";
        leaf status {
          type enumeration {
            enum generated {
              description
                "Key exists.";
            }
            enum unknown {
              description
                "Key status unknown.";
            }
          }
          description
            "This leaf specifies the status of the
             ssh key.";
```


Copyright© 2022 Ciena® Corporation

```
      }

      leaf fingerprint {
        type string;

        description
          "This leaf displays the ssh key if it has
           been generated.";
      }
    }
    list listener {
      key listener-address;
      description
        "This list contains a list of SSH listeners.";

      leaf listener-address {
        type inet:host;
        description
          "This leaf specifies the listener hostname";
      }

      leaf interface {
        type enumeration {
          enum all {
            description
              "All interfaces.";
          }
          enum mgmt-interfaces {
            description
              "Management interfaces.";
          }
          enum unknown {
            description
              "Interface type unknown.";
          }
        }
        description
          "This leaf specifies the current listener
           interface.";
      }
    }

    container algorithm {
      description
        "This container holds all ssh server
         algorithms along with the respective
         state.";
      uses kex-state-group;
      uses encryption-state-group;
      uses pka-state-group;
    }
  }

  container client {
    description
      "SSH client operational data.";
    container algorithm {
      description
        "This container holds all ssh client
         algorithms along with the respective
```


Copyright© 2022 Ciena® Corporation

```
            state.";
          uses kex-state-group;
          uses encryption-state-group;
          uses mac-state-group;
        }
      }
    }
  }

  ////
  // RPCs
  ////

  rpc ssh-key-generate {
    description
      "Requests the server to generate the ssh key, returns
       the status of the request.";
    input {
      leaf key-generate {
        type enumeration {
          enum generate {
            description
              "Generate ssh key.";
          }
        }
        description
          "This leaf generates an ssh key. ";
      }
    }
    output {
      leaf status {
        type enumeration {
          enum success {
            description
              "Key generation succeeded.";
          }
          enum failure {
            description
              "Key generation failure.";
          }
        }
        description
          "This leaf specifies the result of the ssh key
           generation";
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-syslog.yang

```
module ciena-syslog {
  namespace "http://www.ciena.com/ns/yang/ciena-syslog";
  prefix "ciena-syslog";

  import ietf-inet-types {
    prefix inet;
  }

  import ciena-types {
    prefix ciena;
  }

  import ietf-syslog {
    prefix syslog;
  }

  import ietf-syslog-types {
    prefix syslogtypes;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
   "This YANG module defines Ciena's global configuration for syslog settings.

    Copyright (c) 2016 Ciena Corporation.  All rights
    reserved.

    All information contained herein is, and remains
    the property of Ciena Corporation. Dissemination of this
    information or reproduction of this material is strictly
    forbidden unless prior written permission is obtained from
    Ciena Corporation.";

  revision 2016-08-14 {
    description
      "Initial version";
    reference
      "RFC 6020: YANG - A Data Modeling Language for
       the Network Configuration Protocol (NETCONF).
       No specific reference; standard not available.";
  }

  /*
   * Configuration data nodes
   */
  augment /syslog:syslog/syslog:log-actions/syslog:remote {
    description
```

```
     "A replacement for the syslog destination in the ietf-syslog YANG model.
        SAOS only supports one facility per syslog collector and a list of
severities per facility.";

    leaf admin-state {
      type ciena:admin-state;
      description
        "The administrative state of Syslog.";
    }

    list destination {
      key address;
      description
        "A replacement for the ietf-syslog destination list.";

      leaf address {
        type "inet:host";
        description
          "Syslog UDP address.";
      }

      leaf admin-state {
        type ciena:admin-state;
        description
        "The administrative state of the Syslog server.";
      }

      leaf port {
        type "inet:port-number" {
          range "514..65535";
        }
        description
          "UDP port number.";
      }

      leaf facility {
        type identityref {
          base syslogtypes:syslog-facility;
        }
        description
          "Describes the Syslog facility.";
      }

      leaf-list severity {
        type syslogtypes:severity;
        description
          "Set a list of severities for the facility.";
      }

      leaf custom-prefix {
        type string {
          length "0..15";
        }
        description
          "Syslog message prefix.";
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-system.yang

```
module ciena-system {
  namespace "http://www.ciena.com/ns/yang/ciena-system";
  prefix "ciena-sys";

  import ietf-system {
    prefix sys;
  }

  import ciena-types {
    prefix ciena;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module defines Ciena's global
     configuration for system settings.

     Copyright (c) 2016 Ciena Corporation.  All rights
     reserved.

     All information contained herein is, and remains
     the property of Ciena Corporation. Dissemination of this
     information or reproduction of this material is strictly
     forbidden unless prior written permission is obtained from
     Ciena Corporation.";

  revision 2017-09-12 {
    description
      "Removed state-dump RPC, shortened lines to
       comply with yang RFC standards.";
    reference
      "Ciena Functional Specification.";
  }

  revision 2017-05-25 {
    description
      "Initial version";
    reference
      "RFC 6020: YANG - A Data Modeling Language for
       the Network Configuration Protocol (NETCONF).
       No specific reference; standard not available.";
  }

  /*
   * typedefs
   */
```


Copyright© 2022 Ciena® Corporation

```
/*
 * Identities
 */

/*
 * Features
 */

/*
 * Groupings
 */


/*
 * Configuration data nodes
 */
augment /sys:system {
  description
    "Configuration of the system and its properties.";

  container server {
    description
      "Server security administrative settings.";

    leaf scp {
      type ciena:admin-state;
      description
        "The SCP server administrative state.";
    }

    leaf sftp {
      type ciena:admin-state;
      description
        "The SFTP server administrative state.";
    }
  }
}

/*
 * Operational state data nodes
 */
augment /sys:system-state {

  description
    "Operational data for the system.";

  container server {
    description
      "Server security operational settings.";

    leaf scp {
      type ciena:oper-state;
      description
        "The SCP server operational state.";
    }

    leaf sftp {
      type ciena:oper-state;
      description
        "The SFTP server operational state.";
```


Copyright© 2022 Ciena® Corporation

```
      }
    }

  }

  /*
   * RPCs
   */
  rpc process-list {
    if-feature ciena:saos-6x;
    description
      "Show the currently running processes.";

    output {
      list processes {
        key pid;

        leaf pid {
          type uint32;
          description
            "PID.";
        }

        leaf tty {
          type string;
          description
            "TTY.";
        }

        leaf stat {
          type string;
          description
            "STAT";
        }

        leaf time {
          type string;
          description
            "TIME";
        }

        leaf command {
          type string;
          description
            "COMMAND.";
        }

        description
          "List of the running processes, their process ID,
           and other useful information";
      }
    }
  }

  rpc sleep {
    if-feature ciena:saos-8x;
    description
      "Sleep command for debugging/testing purposes.";

    input {
```


Copyright© 2022 Ciena® Corporation

```
      leaf time {
        type uint64;
        units "milliseconds";
        description
          "Time to sleep.";
      }

      leaf message {
        type string;
        description
          "Optional message that will be returned
           to you in the output of the RPC.";
      }
    }
    output {
      leaf message {
        type string;
        description
          "Message from the input if provided.";
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-tacacs.yang

```
module ciena-tacacs {
  namespace "http://www.ciena.com/ns/yang/ciena-tacacs";
  prefix "ciena-tacacs";

  import ietf-inet-types {
    prefix "inet";
  }

  import ietf-yang-types {
    prefix "yang";
  }

  import ciena-types {
    prefix "ciena";
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
    E-mail:  yang@ciena.com
    Postal:  7035 Ridge Road
             Hanover, Maryland 21076
             U.S.A.
    Phone:   +1 800-921-1144
    Fax:     +1 410-694-5750";

  description
    "This YANG module defines the nodes
    for the TACACS+ Server for the NETCONF protocol.

    Copyright (c) 2016 Ciena Corporation.  All rights
    reserved.

    All information contained herein is, and remains
    the property of Ciena Corporation. Dissemination of this
    information or reproduction of this material is strictly
    forbidden unless prior written permission is obtained from
    Ciena Corporation.";

  revision "2017-04-25" {
     description
        "Added preferred source address support.";
  }

  revision "2016-02-10" {
    description
      "Initial version";
    reference
      "RFC 6020: YANG - A Data Modeling Language for
       the Network Configuration Protocol (NETCONF).
       No specific reference; standard not available.";
  }

  /*
   * Typedefs
   */
```

```
  typedef auth-server-enum {
    type enumeration {
      enum authentication-server {
        description "";
      }
      enum authorization-server {
        description "";
      }
      enum accounting-server {
        description "";
      }
    }
    description "";
  }

  /*
   * Groupings
   */

  grouping server-state {
    description "";

    leaf address {
      type inet:host;
      description
        "Host name or IP address of the TACACS server.";
    }

    leaf resolved-address {
      type inet:ip-address;
      description
        "When the address leaf represents:
         Host name    : The resolved address is either the Ipv4 address or
Ipv6 address.
         Ipv4 address : The resolved address is the same Ipv4 address.
         Ipv6 address : The resolved address is the same Ipv6 address.";
    }

    leaf last-used {
      type string;
      description "";
    }

  } // end server-state grouping

  grouping server-statistics {
    description "";

    list statistics {
      key resolved-address;
      description "";

      leaf resolved-address {
        type inet:host;
        description
          "Host name or IP address of the TACACS server.";
      }

      leaf access-requests {
```


Copyright© 2022 Ciena® Corporation

```
    type yang:counter32;
    description
      "The number of TACACS Access-Request packets sent
       to this server. This does not include retransmissions.";
  }

  leaf timeouts {
    type yang:counter32;
    config false;
    description
      "The number of authentication timeouts to this server.
       After a timeout the client may retry to the same
       server, send to a different server, or
       give up. A retry to the same server is counted as a
       retransmit as well as a timeout. A send to a different
       server is counted as a Request as well as a timeout.";
  }

  leaf access-rejects {
    type yang:counter32;
    description
      "The number of TACACS Access-Reject packets
       (valid or invalid) received from this server.";
  }

  leaf access-accepts {
    type yang:counter32;
    description
      "The number of TACACS Access-Accept packets
       (valid or invalid) received from this server.";
  }

  leaf access-retransmissions {
    type yang:counter32;
    description
      "The number of TACACS Access-Request packets
       retransmitted to this TACACS authentication server.";
  }

  leaf bad-header-sequence {
    type yang:counter32;
    description
      "The number of TACACS packets that were
       received from this server on the authentication port
       and dropped for some other reason.";
  }

  leaf malformed-access-responses {
    type yang:counter32;
    description
      "The number of malformed TACACS Access-Response
       packets received from this server.
       Malformed packets include packets with
       an invalid length. Bad authenticators or
       Signature attributes or unknown types are not
       included as malformed access responses.";
  }

  leaf bad-authenticators {
    type yang:counter32;
```


Copyright© 2022 Ciena® Corporation

```
        description
          "The number of TACACS Access-Response packets
           containing invalid authenticators or Signature
           attributes received from this server.";
      }

      leaf unknown-types {
        type yang:counter32;
        description
          "The number of TACACS packets of unknown type that
           were received from this server on the authentication port.";
      }

    } // end list
  } // end server-stats group

  grouping server-attributes {
    description "";

    leaf address {
      type inet:host;
      description
        "Host name or IP address of the TACACS server.";
    }

    leaf priority {
      type int32;
      description
        "Specifies the priority of TACACS servers configured on the device.
         This is the order in which the servers are accessed.";
    }

    leaf port {
      type inet:port-number {
        range "0..65535";
      }
      description
        "The destination TCP port number to which TACACS
         messages should be sent.  The TACACS server is not
         used for authentication if this port number is 0.";
    }

    leaf admin-enabled {
      type ciena:admin-state;
      description
      "Setting this node administratively enables/disables this TACACS server
entry on the device.";
    }

  } // end server-attributes grouping

  /*
   * Configuration data nodes
   */
  container tacacs {
    description "";

    leaf admin-enabled {
      type ciena:admin-state;
      description
```


Copyright© 2022 Ciena® Corporation

```
       "Setting this node administratively enables/disables the TACACS client
on the device.";
    }

    leaf authentication-enhanced-multi-factor {
      type ciena:on-off-state;
      description
       "Setting this node turns on enhanced multi-factor authentication with
TACACS+.";
    }

    leaf preferred-source-address {
       type ciena:preferred-source-address;
       description
          "The setting for which address to use with radius.";
    }


    leaf timeout {
      type int32 {
        range "1..30";
      }
      units "seconds";
      description
        "This is the time in seconds between retransmissions
         to the TACACS server.";
    }

    // Security key and secret
    uses ciena:security-key;

    leaf key-min-len {
      type int32 {
        range "2..64";
      }
      description
        "Setting this node sets the minimum length for TACACS secret key.";
    }

    leaf search-method {
      type enumeration {
        enum priority {
          description "";
        }
        enum cached {
          description "";
        }
      }
      description
        "Setting this node sets the search method.";
    }

    leaf global-servers {
      type ciena:on-off-state;
      description
        "Setting this node turns off global servers.";
    }

    leaf privilege-level-admin {
      type int32 {
```


Copyright© 2022 Ciena® Corporation

```
        range "1..13";
      }
      description
        "Indicates the privilege level used for mapping a range of TACACS
         privilege levels to Admin user privilege level. A privilege level
returned
         by a server is compared to the privilege-level-rw leaf.
         If the server privilege level is less than privilege-level-rw
         and greater than or equal to the privilege-level-admin, the
         privilege level is Admin. Not supported on all platforms.";
    }

    leaf privilege-level-super {
      type int32 {
        range "2..14";
      }
      description
        "Indicates the privilege level used for mapping a range of TACACS
         privilege levels to R-W user privilege level.  A privilege level
returned
         by a server is compared to privilege-level-diag.
         If the server privilege level is less than privilege-level-diag
         and greater than or equal to the privilege-level-rw, the
         privilege level is super-user.";
    }

    leaf privilege-level-diag {
      type int32 {
        range "3..15";
      }
      description
        "Indicates the privilege level used for mapping a range of TACACS
        privilege levels to Diagnostic user privilege level. A privilege level
returned
         by a server is compared to this value. If server privilege level is
greater than
         or equal to the privilege-level-diag, the privilege level is diag.";
    }

    container global {
      description "";
      list server {
        key address;
        description "";
        uses server-attributes;
      }

    } // end global-server container

    container authentication {
      description "";

      leaf admin-enabled {
        type ciena:admin-state;
        description
          "Setting this node administratively enables/disables TACACS
authentication on the device.";
      }

      leaf syslog-admin-state {
```


Copyright© 2022 Ciena® Corporation

```
        type ciena:admin-state;
        description
          "Setting this node administratively enables/disables TACACS Syslog
messages on the device.
           When enabled it sends authentication events.";
      }

      list server {
        key address;
        description "";
        uses server-attributes;
      }
    } // end authentication-server container

    container authorization {
      description "";

      leaf admin-enabled {
        type ciena:admin-state;
        description
          "Setting this node administratively enables/disables TACACS
authorization on the device.";
      }

      list server {
        key address;
        description "";
        uses server-attributes;
      }

    } // end authorization-server container

    container accounting {
      description "";

      leaf admin-enabled {
        type ciena:admin-state;
        description
       "Setting this node administratively enables/disables TACACS accounting
on the device.";
      }

      leaf accounting-session {
        type ciena:on-off-state;
        description
          "Setting this node turns on session logon/logoff logging with
tacacs.";
      }

      leaf accounting-command {
        type ciena:on-off-state;
        description
          "Setting this node turns on command start/stop logging.";
      }

      list server {
        key address;
        description "";
        uses server-attributes;
      }
```


Copyright© 2022 Ciena® Corporation

```
    } // end accounting-server container

  } // end tacacs container

  /*
   * Operational state data nodes
   */
  container tacacs-state {
    config false;
    description "";

    container global-server {
      description "";
      list server {
        description "";
        uses server-state;
      }
    }

    container authentication {
      description "";
      list server {
        key address;
        description "";
        uses server-state;
      }
    }

    container authorization {
      description "";
      list server {
        key address;
        description "";
        uses server-state;
      }
    }

    container accounting {
      description "";
      list server {
        key address;
        description "";
        uses server-state;
      }
    }

    container statistics {
      description "";

      container authentication {
        description "";
        uses server-statistics;
      }

      container authorization {
        description "";
        uses server-statistics;
      }
```


Copyright© 2022 Ciena® Corporation

```
      container accounting {
        description "";
        uses server-statistics;
      }
    }     // End Statistics
  } // end tacacs-state container

  /* RPC Operations */

  rpc clear-statistics {
    description "An RPC operation to clear the statistics for a server and
should should return the status of the operation.";
    input {
      choice clear-range {
        description "";
        case server-type {
          container server-type {
            description "";
            leaf server-select {
              type auth-server-enum;
              description "";
            }
          }
        }
        case server-and-address {
          container server-and-address {
            description "";
            leaf server-select {
              type auth-server-enum;
              description "";
            }
            leaf address {
              type inet:host;
              description
                "Host name or IP address of the TACACS server.";
            }
          }
        }
        case all {
          container all {
            description "";
            leaf all {
              type empty;
              description "";
            }
          }
        }
      } // end clear-range choice
    }
    output {
      leaf status {
        type string;
        description "";
      }
    }
  } // end rpc clear-statistics
} // end ciena-tacacs module
```


Copyright© 2022 Ciena® Corporation

ciena-telnet.yang

```
module ciena-telnet {
  namespace "http://ciena.com/ns/yang/ciena-telnet";
  prefix "ciena-telnet";

  import ciena-types {
    prefix ciena;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module defines Ciena's proprietary
     configuration of Telnet.

     Copyright (c) 2016 Ciena Corporation.  All rights reserved.

     All information contained herein is, and remains
     the property of Ciena Corporation. Dissemination of this
     information or reproduction of this material is strictly
     forbidden unless prior written permission is obtained from
     Ciena Corporation.";

  revision "2017-08-04" {
    description
      "Added nodes for client enable/disable";
    reference
      "";
  }

  revision "2016-06-14" {
    description
      "Initial version";
    reference
      "Enter a reference here";
  }

  /*
   * typedefs
   */

  /*
   * Identities
   */

  /*
   * Features
   */
```

```
  /*
   * Configuration data nodes
   */

  container telnet {
    description
      "This container holds all telnet related
       configuration.";
    container server {
      description
        "This container holds all telnet server
         configuration.";

      leaf admin-state {
        type ciena:admin-state;
        description
          "This leaf specifies the administrative
           state of the telnet server";
      }
    }
    container client {
      description
        "This container holds all telnet client
         configuration.";

      leaf admin-state {
        type ciena:admin-state;
        description
          "This leaf specifies the administrative
           state of the telnet client";
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-test-1.yang

```
module ciena-test-1 {
    namespace "http://www.ciena.com/ns/yang/ciena-test-1";
    prefix "ciena-test1";
    revision "2017-01-26";

    typedef fish-type {
        type enumeration {
            enum onefish;
            enum twofish;
            enum redfish;
            enum bluefish;
        }
        description "Basic type is enumeration(one two three)";
    }

    typedef minibits-type {
        type bits {
            bit alpha;
            bit beta;
            bit charlie;
            bit delta;
        }
        description "Basic type is bits(abcd)";
    }

    typedef bits-type {
        type bits {
            bit zero;
            bit one;
            bit two {
                position 2;
            }
            bit after_two;
            bit four;
            bit five;
            bit six;
            bit seven;
            bit eight;
            bit nine;
            bit ten;
            bit alpha;
            bit beta;
            bit charlie;
            bit delta;
            bit echo;
            bit foxtrot;
            bit golf;
            bit hotel;
            bit india;
            bit juliett;
            bit kilo;
            bit lima;
            bit mike;
            bit november;
            bit oscar;
            bit papa;
            bit quebec;
            bit romeo;
```


Copyright© 2022 Ciena® Corporation

```
            bit sierra;
            bit tango;
            bit uniform;
            bit victor;
            bit whiskey;
            bit xray;
            bit yankee;
            bit zulu;
        }
        description "Basic type is bits(zero one two etc)";
    }

    typedef season-enum {
        type enumeration {
            enum summer;
            enum winter;
            enum fall;
            enum spring;
            enum onemonth;
            enum twomonths;
        }
        description "Basic type is enumeration(abcd)";
    }

     typedef my-decimal {
         type decimal64 {
             fraction-digits 2;
         }
     }

    identity id;

    identity id1 {
        base id;
    }

    identity id2 {
        base id;
    }

    identity id3 {
        base id;
    }

    identity id2.1 {
        base id2;
    }

    identity id2.2 {
        base id2;
    }

    typedef id-type {
        type identityref {
            base id;
        }
        description "Basic type is identityref(id1 id2 id2.1 id2.2 id3)";
     }

     typedef thingnameref {
```


Copyright© 2022 Ciena® Corporation

```
        type leafref {
            path "/ciena-test1:things/ciena-test1:thing/ciena-test1:name";
        }
     }

     typedef security-model {
      type union {
        type enumeration {
          enum v1;
          enum v2c;
          enum usm;
          enum tsm;
        }
        type int32 {
          range "1..2147483647";
        }
      }
    }

    typedef union-type {
        type union {
            type int32;
            type boolean;
            type my-decimal;
            type minibits-type;
            type fish-type;
            type id-type;
            type binary;
            type string;
        }
        description "Basic type is union(int32 string)";
    }

    typedef union-str-type {
        type union {
            type int32;
            type boolean;
            type my-decimal;
            type minibits-type;
            type fish-type;
            type id-type;
            type string;
        }
        description "Basic type is union(int32 string)";
    }

    container c1 {
        choice chx {
            case on {
                container c3{

                    leaf int-leaf {
                        type uint32;
                    }

                    leaf bool-leaf {
                        type boolean;
                    }
                }
            }
```


Copyright© 2022 Ciena® Corporation

```
            case off {
                leaf empty-leaf {
                    type empty;
                }
            }
        }
        leaf x1 { type string; }
        leaf-list foo1 {
            type string;
            min-elements 1;
        }
        presence "presence is not mandatory";
    }

    deviation /c2/foo2 {
        deviate replace {
            min-elements 0;
        }
    }
    container c2 {
        leaf x2 { type string; }
        leaf-list foo2 {
            type string;
            min-elements 1;
        }
    }

    container things {

        list thing {
            key "name";

            leaf name {
                type string;
            }
            list place {
               key "placename placeid";
               leaf placename{
                  type string;
               }
               leaf e1{
                  type empty;
               }
             leaf placeid {
              type id-type;
             }
             leaf-list placeidlist {
              type id-type;
             }
             leaf-list placeleaflist {
              type thingnameref;
             }
               list fruit {
                  key "fruitref fruitname";
                  leaf fruitname{
                     type string;
                  }
                  leaf fruitseason{
                     type season-enum;
```


Copyright© 2022 Ciena® Corporation

```
            }
            leaf likethefruit{
               type boolean;
            }
            leaf fruitref{
               type thingnameref;
            }
         }
         list animal {
            key "animalenum animalunion animald64";
            leaf animalenum{
               type fish-type;
            }
            leaf animald64{
               type my-decimal;
            }
            leaf animalunion{
               type union-type;
            }
            container stuffx{
                list toy{
                    key "toystr toyint8 toybool toybit toybin";
                    leaf toystr{
                       type string;
                    }
                    leaf toyint8{
                       type int8;
                    }
                    leaf toybit{
                       type minibits-type;
                    }
                    leaf toybool{
                       type boolean;
                    }
                    leaf toydesc{
                       type string;
                    }
                    leaf toybin{
                       type binary;
                    }
                }
            }
            leaf food{
               type string;
            }
         }
      }
      leaf description {
          type string;
      }
      leaf number {
          type uint32;
      }
      leaf mybitval {
          type minibits-type;
      }
      leaf boolvar {
      type boolean;
      }
      leaf fishenum {
```


Copyright© 2022 Ciena® Corporation

```
            type fish-type;
          }
          leaf mybin {
          type binary;
          }
          leaf myd64 {
          type my-decimal;
          }
          leaf myunion {
              type union-type;
          }
        leaf-list thingstr {
            type string;
            description "List of thing strs";
        }
        leaf-list thingbit {
            type minibits-type;
            description "List of thing bits";
        }
        leaf-list thingbin {
            type binary;
            description "List of thing binary";
        }
        leaf-list thingenum {
            type fish-type;
            description "List of thing enums";
        }
        leaf-list thingint {
            type uint32;
            description "List of thing ints";
        }
        leaf-list thingbool {
            type boolean;
            description "List of thing booleans";
        }
        leaf-list thingdec {
            type my-decimal;
            description "List of thing dec";
        }
        leaf-list thingunion {
            type union-str-type;
            description "List of thing unions";
        }
      }
  }
  list artist{
      key "artistname";

      leaf artistname {
         type string;
      }
      list song {
         key "singername";
         leaf singername{
            type string;
         }
         list music {
            key "songid musicstyle";
            leaf songid{
               type uint32;
```


Copyright© 2022 Ciena® Corporation

```
                }
                leaf musicstyle{
                   type string;
                }
                leaf songname{
                   type fish-type;
                }
            }
            leaf musicschool{
                type season-enum;
            }
        }
        leaf movies{
           type boolean;
        }
    }

    rpc word-count {
        input {
            leaf word {
                type string;
            }
        }
        output {
            leaf count {
                type uint32;
            }
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-test-rollback.yang

```
module ciena-test-rollback {
   namespace "http://www.ciena.com/ns/yang/ciena-test-rollback";
    prefix "ciena-test-rollback";
    revision "2016-11-16";

    container test-rollback {

    leaf rollback-condition {
      type boolean;
    }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-test-types-deviation.yang

```
module ciena-test-types-deviation {
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:demo-dev";
    prefix "ciena-test-types-dev";

    import ciena-test-types {
        prefix "ctt";
    }

    revision "2016-07-20" {
        description
            "Deviating the nodes that are not currently supported by ciena";
    }

    deviation /ctt:config-nodes/ctt:boolean-leaflist {
      deviate not-supported;
   }

    deviation /ctt:oper-nodes/ctt:boolean-leaflist {
      deviate not-supported;
   }

}
```


Copyright© 2022 Ciena® Corporation

ciena-test-types.yang

```
module ciena-test-types {
    namespace "http://www.ciena.com/ns/yang/ciena-test-types";
    prefix "ciena-test-types";
    revision "2017-04-27";

    typedef season-enum {
        type enumeration {
            enum spring;
            enum summer;
            enum fall;
            enum winter;
        }
    }
   typedef bits-type {
      type bits {
          bit zero;
          bit one;
          bit two {
              position 2;
       }
          bit after_two;
            bit four;
            bit five;
            bit six;
            bit seven;
            bit eight;
            bit nine;
            bit ten;
            bit alpha;
            bit beta;
            bit charlie;
            bit delta;
            bit echo;
            bit foxtrot;
            bit golf;
            bit hotel;
            bit india;
            bit juliett;
            bit kilo;
            bit lima;
            bit mike;
            bit november;
            bit oscar;
            bit papa;
            bit quebec;
            bit romeo;
            bit sierra;
            bit tango;
            bit uniform;
            bit victor;
            bit whiskey;
            bit xray;
            bit yankee;
            bit zulu;
        }
        description "Basic type is bits(zero one two etc)";
    }
    identity id;
```


Copyright© 2022 Ciena® Corporation

```
identity id1 {
    base id;
}

identity id2 {
    base id;
}

identity id3 {
    base id;
}

identity id2.1 {
    base id2;
}

identity id2.2 {
    base id2;
}

typedef id-type {
    type identityref {
        base id;
    }
    description "Basic type is identityref(id1 id2 id2.1 id2.2 id3)";
}

grouping all-type-leaves {
    leaf int8-leaf {
        type int8;
    }
    leaf int16-leaf {
        type int16;
    }
    leaf int32-leaf {
        type int32;
    }
    leaf int64-leaf {
        type int64;
    }
    leaf uint8-leaf {
        type uint8;
    }
    leaf uint16-leaf {
        type uint16;
    }
    leaf uint32-leaf {
        type uint32;
    }
    leaf uint64-leaf {
        type uint64;
    }
    leaf decimal64-leaf {
        type decimal64 {
        fraction-digits 2;
     }
    }
    leaf string-leaf {
        type string;
```


Copyright© 2022 Ciena® Corporation

```
        }
        leaf boolean-leaf {
            type boolean;
        }
        leaf enumeration-leaf {
            type season-enum;
        }
        leaf bits-leaf {
            type bits-type;
        }
        leaf binary-leaf {
            type binary;
        }
        leaf identityref-leaf {
        type id-type;
        }
        leaf empty-leaf {
            type empty;
        }
    }

    grouping all-type-leaf-lists {
        leaf-list int8-leaflist {
            type int8;
        }
        leaf-list int16-leaflist {
            type int16;
        }
        leaf-list int32-leaflist {
            type int32;
        }
        leaf-list int64-leaflist{
        type int64;
        }
        leaf-list uint8-leaflist {
            type uint8;
        }
        leaf-list uint16-leaflist {
            type uint16;
        }
        leaf-list uint32-leaflist {
            type uint32;
        }
        leaf-list uint64-leaflist {
            type uint64;
        }
        leaf-list decimal64-leaflist {
            type decimal64 {
            fraction-digits 2;
         }
        }
        leaf-list string-leaflist {
            type string;
        }
        leaf-list boolean-leaflist {
            type boolean;
        }
        leaf-list enumeration-leaflist {
            type season-enum;
        }
```


Copyright© 2022 Ciena® Corporation

```
        leaf-list bits-leaflist {
            type bits-type;
        }
        leaf-list binary-leaflist {
            type binary;
        }
        leaf-list identityref-leaflist {
        type id-type;
        }

    }


    grouping all-type-union-leaf{

    leaf union-leaf-int8{
      type union{
          type int8;
          type string;
      }
    }

    leaf union-leaf-int16{
      type union{
          type int16;
          type string;
      }
    }

    leaf union-leaf-int32{
      type union{
          type int32;
          type string;
      }
    }

    leaf union-leaf-int64{
      type union{
          type int64;
          type string;
      }
    }

    leaf union-leaf-uint8{
      type union{
          type uint8;
          type string;
      }
    }

    leaf union-leaf-uint16{
      type union{
          type uint16;
          type string;
      }
    }

    leaf union-leaf-uint32{
      type union{
          type uint32;
```


Copyright© 2022 Ciena® Corporation

```
        type string;
    }
  }


  leaf union-leaf-uint64{
    type union{
        type uint64;
        type string;
    }
  }

  leaf union-leaf-decimal64{
    type union{
        type decimal64 {fraction-digits 8;}
        type string;
    }
  }

  leaf union-leaf-enumeration{
    type union{
        type enumeration{
           enum one;
           enum two;
           enum three;
        }
        type string;
    }
  }

  leaf union-leaf-bits{
    type union{
        type bits{
           bit alpha;
           bit beta;
           bit gamma;
        }
        type string;
    }
  }

  leaf union-leaf-identityref{
    type union{
        type id-type;
        type string;
    }
  }

  leaf union-leaf-binary{
    type union{
        type binary;
        type string;
    }
  }

  leaf union-leaf-boolean{
    type union{
        type boolean;
        type string;
    }
```


Copyright© 2022 Ciena® Corporation

```
  }

  leaf union-leaf-string{
    type union{
        type string;
        type decimal64 {fraction-digits 8;}

    }
  }
  }

grouping all-type-union-leaflist{

   leaf-list union-leaflist1{
          type union {
             type boolean;
             type uint64;
             type int64;
             type id-type;
             type decimal64 {fraction-digits 8;}
             type enumeration {
                   enum red;
                   enum blue;
                   enum voilet;
             }
           type bits {
             bit zero;
                bit one;
             }
             type string;
       }
   }

   leaf-list union-leaflist-integer8{
       type union{
          type uint8;
          type int8;
       }
   }

   leaf-list union-leaflist-integer16{
       type union{
          type uint16;
          type int16;
       }
   }

   leaf-list union-leaflist-integer32{
       type union{
          type uint32;
          type int32;
       }
   }

   leaf-list union-leaflist-binary{
       type union{
          type binary;
          type int32;
       }
   }
```


Copyright© 2022 Ciena® Corporation

```
   }

    container config-nodes {
    description "For testing Edit-ConfigData for basic types";
        uses all-type-leaves;
        uses all-type-leaf-lists;
        uses all-type-union-leaf;
        uses all-type-union-leaflist;
        leaf set-defaults {
            type boolean;
        }
    }

    container parent-container{
      description "For testing delete of a parent container";
      leaf uint8-leaf{ type uint8;}
      list child-list{
          key 'string-leaf';
          leaf string-leaf{type string;}
          leaf int8-leaf{type int8;}
      }
   }

    list parent-list{
      description "For testing delete of a parent list in nested list case";
      key "uint8-leaf";
      leaf uint8-leaf{ type uint8;
                       }
      leaf uint16-leaf{ type uint16;
                        }
      list child-list1{
          key 'string-leaf';
          leaf string-leaf{type string;}
          leaf int8-leaf{type int8;}
                    list child-list2 {
                        key "uint16-leaf";
                        leaf uint16-leaf{ type uint16;}
                        leaf string-leaf {type string;}
                    }

      }
   }



    list test-leaf-ref-union {

        key 'leaf-ref-key';
        leaf leaf-ref-key {
            type leafref {
           path "/ciena-test-types:config-nodes/ciena-test-types:union-leaf-
int8";
      }
    }
  }
    list test-leaf-ref-iref {
        key 'i-ref-key';
        leaf i-ref-key {
            type leafref {
```


Copyright© 2022 Ciena® Corporation

```
          path "/ciena-test-types:config-nodes/ciena-test-types:identityref-
leaf";
      }
    }
  }

    container toplevel-container{
      description "For testing delete of a parent container in nested
contianer case";
      leaf uint8-leaf{ type uint8;}
      container child-container{
          leaf string-leaf{type string;}
          leaf int8-leaf{type int8;}
      }
   }

   container container1{
      description "Testing Rollback of child node when parent node delete
causes an error";
      leaf uint8-leaf{ type uint8;}
      container container2{
          leaf string-leaf{type string;}
          leaf int8-leaf{type int8;}
      }
   }

    list toplevel-list{

      description "For testing delete of a parent list";
      key "uint8-leaf";
      leaf uint8-leaf{ type uint8;
                       }
      leaf uint16-leaf{ type uint16;
                       }
      container child-container1{
          leaf string-leaf{type string;}
          leaf int8-leaf{type int8;}
      }
   }

    container err {
        leaf string-leaf{type string;}
    }
    container err-obj {
        leaf uint16-leaf{type uint16;}
    }

    list order-list{
      description "For testing ordered by user for list";
      ordered-by user;
      key "uint8-leaf uint16-leaf";
      leaf uint16-leaf{ type uint16;
                       }
      leaf uint8-leaf{ type uint8;
                       }
   }

   container order-test{
      leaf-list order-leaflist{
          type uint8;
```


Copyright© 2022 Ciena® Corporation

```
          ordered-by user;
      }

   }

    container edit-choice-nested {
        choice test-nested-choice {
            case A {
                leaf a1 {type string;}
                leaf a2 {type string;}
            }
            container B {
                choice B0 {
                    case B01 {
                        leaf b01 {type string;}
                     }
                      case B02 {
                        leaf b02 {type string;}
                     }
                 }
             }
        }
   }

    container oper-nodes {
    description "For testing Get-OperData for basic types";
        config false;
        uses all-type-leaves;
        uses all-type-leaf-lists;
        uses all-type-union-leaf;
        uses all-type-union-leaflist;
    }

    list int8-list{
    config false;
    key "int8-key-leaf";
    leaf int8-key-leaf{ type int8;}
    }

    list int16-list{
    config false;
    key "int16-key-leaf";
    leaf int16-key-leaf{ type int16;}
    }

    list int32-list{
    config false;
    key "int32-key-leaf";
    leaf int32-key-leaf{ type int32;}
    }

    list int64-list{
    config false;
    key "int64-key-leaf";
    leaf int64-key-leaf{ type int64;}
    }

    list uint8-list{
    config false;
    key "uint8-key-leaf";
```


Copyright© 2022 Ciena® Corporation

```
leaf uint8-key-leaf{ type uint8;}
}

list uint16-list{
config false;
key "uint16-key-leaf";
leaf uint16-key-leaf{ type uint16;}
}

list uint32-list{
config false;
key "uint32-key-leaf";
leaf uint32-key-leaf{ type uint32;}
}

list uint64-list{
config false;
key "uint64-key-leaf";
leaf uint64-key-leaf{ type uint64;}
}

list decimal64-list{
config false;
key "decimal64-key-leaf";
leaf decimal64-key-leaf{ type decimal64 {fraction-digits 8;} }
}

list string-list{
config false;
key "string-key-leaf";
leaf string-key-leaf {type string;}
}

list boolean-list{
config false;
key "boolean-key-leaf";
leaf boolean-key-leaf {type boolean;}
}

list enumeration-list{
config false;
key "enumeration-key-leaf";
leaf enumeration-key-leaf {
  type enumeration{
         enum monday;
         enum tuesday;
         enum wednesday;
  }
}
}

list bits-list{
config false;
key "bits-key-leaf";
leaf bits-key-leaf{
  type bits{
      bit red;
      bit blue;
      bit green;
      bit yellow;
```


Copyright© 2022 Ciena® Corporation

```
      }
    }
    }

    list indentityref-list{
    config false;
    key "indentityref-key-leaf";
    leaf indentityref-key-leaf {type id-type;}
    }

    list union-int8-list{
    config false;
    key "union-int8-key-leaf";
    leaf union-int8-key-leaf{
      type union{
          type int8;
          type string;
      }
    }
    }

    list union-int16-list{
    config false;
    key "union-int16-key-leaf";
    leaf union-int16-key-leaf{
      type union{
          type int16;
          type string;
      }
    }
    }


   list union-int32-list{
    config false;
    key "union-int32-key-leaf";
    leaf union-int32-key-leaf{
      type union{
          type int32;
          type string;
      }
    }
    }

    list union-int64-list{
    config false;
    key "union-int64-key-leaf";
    leaf union-int64-key-leaf{
      type union{
          type int64;
          type string;
      }
    }
    }

    list union-uint8-list{
    config false;
    key "union-uint8-key-leaf";
    leaf union-uint8-key-leaf{
      type union{
```


Copyright© 2022 Ciena® Corporation

```
        type uint8;
        type string;
    }
  }
  }

  list union-uint16-list{
  config false;
  key "union-uint16-key-leaf";
  leaf union-uint16-key-leaf{
    type union{
        type uint16;
        type string;
    }
  }
  }

 list union-uint32-list{
  config false;
  key "union-uint32-key-leaf";
  leaf union-uint32-key-leaf{
    type union{
        type uint32;
        type string;
    }
  }
  }

  list union-uint64-list{
  config false;
  key "union-uint64-key-leaf";
  leaf union-uint64-key-leaf{
    type union{
        type uint64;
        type string;
    }
  }
  }

  list union-decimal64-list{
  config false;
  key "union-decimal64-key-leaf";
  leaf union-decimal64-key-leaf{
    type union{
        type decimal64 {fraction-digits 8;}
        type string;
    }
  }
  }

  list union-string-list{
  config false;
  key "union-string-key-leaf";
  leaf union-string-key-leaf{
    type union{
        type int8;
        type string;
    }
  }
  }
```


Copyright© 2022 Ciena® Corporation

```
list union-boolean-list{
config false;
key "union-boolean-key-leaf";
leaf union-boolean-key-leaf{
  type union{
      type boolean;
      type string;
  }
}
}

list union-enumeration-list{
config false;
key "union-enumeration-key-leaf";
leaf union-enumeration-key-leaf{
  type union{
      type enumeration{
         enum earth;
         enum mars;
         enum venus;
         enum jupiter;
      }
      type string;
  }
}
}

list union-bits-list{
config false;
key "union-bits-key-leaf";
leaf union-bits-key-leaf{
  type union{
      type bits{
            bit java;
            bit python;
            bit scala;
      }
      type string;
  }
}
}

container oper1 {
    config false;
    choice oper-choice {
        case A {
            leaf a1 {type string;}
            leaf a2 {type string;}
        }
        container B {
            leaf b1 {type uint8;}
        }
        case C {
            list choiceList {
            key "choicekey";
            leaf choicekey {type uint16;}
            leaf c1 {type uint32;}
            }
        }
```


Copyright© 2022 Ciena® Corporation

```
            case D {
                choice D0 {
                    case D01 {
                        leaf d01 {type string;}
                     }
                      case D02 {
                        leaf d02 {type string;}
                      }
                 }
             }
             leaf E {type string;}

             case F {
                 container F1 {
                 leaf f1 {type uint8;}
                 }
             }
             list G{
                 key "choicekeyG";
                 leaf choicekeyG {type uint16;}
                 leaf g1 {type uint32;}
             }
        }
    }

    container operTestnestedlist {
        leaf t1 {type uint16;}
        list l1 {
            config false;
            key "l1key";
            leaf l1key {type int16;}
            leaf l11 {type uint16;}

            list l2 {
                config false;
                key "l2key";
                leaf l2key {type int16;}
                leaf l22 {type uint16;}
            }
        }
    }


    list operTestContainer {
        config false;
        key "lkey";
        leaf lkey {type string;}
        container lC1 {
            leaf lc1 {type int16;}
            list lC1l2 {
                key "c1l2key";
                leaf c1l2key {type uint32;}
                leaf c1l2 {type uint16;}
            }
        }
    }

    container get-parent-container{
    config false;
```


Copyright© 2022 Ciena® Corporation

```
leaf p1{ type int16;}

list child1{
  key "cleaf1";
  leaf cleaf1 {type uint32;}

}

}

rpc rpc-test-types {
    input {
        leaf word {
            type string;
        }
        leaf rhyme {
        type enumeration {
        enum sing;
        enum ding;
        enum ping;
        }
        }
        leaf increment-u8 {
            type uint8;
        }
        leaf increment-u16 {
            type uint16;
        }
        leaf increment-u32 {
            type uint32;
        }
        leaf increment-u64 {
            type uint64;
        }
        leaf increment-i8 {
            type int8;
        }
        leaf increment-i16 {
            type int16;
        }
        leaf increment-i32 {
            type int32;
        }
        leaf increment-i64 {
            type int64;
        }
        leaf increment-idref {
            type id-type;
        }

        leaf increment-decimal64{
        type decimal64 {
           fraction-digits 2;
     }
        }

        leaf rpc-boolean-in{
        type boolean;
        }
```


Copyright© 2022 Ciena® Corporation

```
    leaf rpc-binary-in {
        type binary;
    }

    leaf rpc-bits-in {
     type bits-type;
}

     leaf u32-union {
         type union {
             type uint32;
             type string;
         }
     }
     leaf-list leaflist-u32 {
         type uint32;
     }
     leaf-list leaflist-union {
         type union {
             type uint32;
             type enumeration {
                 enum a;
                 enum b;
                 enum c;
             }
         }
     }
     container cntr {
         leaf enumeration-leaf {
             type enumeration {
                 enum hello;
                 enum how_are_you;
                 enum goodbye;
             }
         }
         leaf uint8-leaf {
             type uint8;
         }
     }
     list lst {
         leaf u32 {
             type uint32;
         }
         leaf str {
             type string;
         }
     }
     choice ch {
         leaf case1 {
             type string;
         }
         container case2 {
             list xxx {
                 leaf yyy {
                     type uint32;
                 }
             }
         }
     }
     uses all-type-leaf-lists;
```


Copyright© 2022 Ciena® Corporation

```
        uses all-type-union-leaf;
        uses all-type-union-leaflist;
    }
    output {
        leaf count {
            type uint32;
        }
        leaf rhyme {
        type string;
        }
        leaf response {
        type enumeration {
        enum song;
        enum dong;
        enum pong;
        }
        }
        leaf incremented-u8 {
            type uint8;
        }
        leaf incremented-u16 {
            type uint16;
        }
        leaf incremented-u32 {
            type uint32;
        }
        leaf incremented-u64 {
            type uint64;
        }
        leaf incremented-i8 {
            type int8;
        }
        leaf incremented-i16 {
            type int16;
        }
        leaf incremented-i32 {
            type int32;
        }
        leaf incremented-i64 {
            type int64;
        }
      leaf incremented-idref {
            type id-type;
        }
        leaf incremented-u32-union {
            type union {
                type uint32;
                type string;
            }
        }
        leaf-list incremented-leaflist-u32 {
            type uint32;
        }

        leaf incremented-decimal64{
        type decimal64 {
           fraction-digits 2;
     }
        }
        leaf rpc-boolean-out{
```


Copyright© 2022 Ciena® Corporation

```
            type boolean;
            }

           leaf rpc-binary-out {
               type binary;
           }

           leaf rpc-bits-out {
            type bits-type;
       }

            leaf-list incremented-leaflist-union {
                type union {
                    type uint32;
                    type enumeration {
                        enum a;
                        enum b;
                        enum c;
                    }
                }
            }
            uses all-type-leaf-lists;
            uses all-type-union-leaf;
            uses all-type-union-leaflist;
            container c2 {
                leaf result {
                    type string;
                }
            }
            list l2 {
                leaf u32 {
                    type uint32;
                }
                leaf str {
                    type string;
                }
            }
            choice ch {
                leaf case1-len {
                    type uint32;
                }
                container case2 {
                    list xxx {
                        leaf yyy {
                            type uint32;
                        }
                    }
                }
            }
        }
    }

    rpc rpc-inerr {
        input {
            leaf inerr {type string;}
        }
    }

    rpc rpc-outerr {
        output {
```


Copyright© 2022 Ciena® Corporation

```
            leaf outerr {type string;}
        }
    }


    choice edit-choice {
        case B {
            container Be {
                leaf uint8-leaf {type uint8;}
            }
        }
    }

}
```


Copyright© 2022 Ciena® Corporation

ciena-time.yang

```
module ciena-time {

  namespace "http://www.ciena.com/ns/yang/ciena-time";
  prefix time;

  import ietf-system {
    prefix sys;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module defines Ciena's global configuration for time
     settings.

     Copyright (c) 2016 Ciena Corporation.  All rights
     reserved.

     All information contained herein is, and remains
     the property of Ciena Corporation. Dissemination of this
     information or reproduction of this material is strictly
     forbidden unless prior written permission is obtained from
     Ciena Corporation.";

  revision 2017-06-06 {
    description
      "Initial version";
    reference
      "RFC 6020: YANG - A Data Modeling Language for
       the Network Configuration Protocol (NETCONF).
       No specific reference; standard not available.";
  }

  /*
   * typedefs
   */

  typedef time-offset-seconds {
    type int32 {
      range "-43200 .. 50400";
    }
    units "seconds";
    description
      "The system time-offset in seconds from UTC.";
  }

  typedef time-offset-source {
    type enumeration {
```


Copyright© 2022 Ciena® Corporation

```
      enum user {
        description
          "The time offset was set by a user.";
      }
      enum ntp {
        description
          "The time offset was set by NTP.";
      }
      enum dhcp {
        description
          "The time offset was set by DHCP.";
      }
    }
    description
      "The source of the time offset configuration.";
  }

  typedef timestamp-type {
    type enumeration {
      enum local {
        description
          "Local time will be used for system log timestamps.";
      }
      enum UTC {
        description
          "UTC will be used for system log timestamps.";
      }
    }
    description
      "The timezone used for timestamps in system logs.";
  }

  /*
   * Identities
   */

  /*
   * Features
   */

  /*
   * Configuration data nodes
   */
  augment /sys:system/sys:clock {
    description
      "SAOS time-related features not in the IETF system clock model.";

    leaf timestamp {
      type timestamp-type;
      description
        "The timezone used for timestamps in system logs.";
    }
  }


  /*
   * Operational state data nodes
   */
  augment /sys:system-state/sys:clock {
    description
```


Copyright© 2022 Ciena® Corporation

```
      "SAOS time-related features not in the IETF system clock model.";

    leaf time-configuration-source {
      type time-offset-source;
      description
        "The source of the time-offset configuration.";
    }
  }

  /*
   * RPCs
   */
}
```


Copyright© 2022 Ciena® Corporation

ciena-twamp.yang

```
module ciena-twamp
{
    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:ciena-twamp";
    prefix "ciena-twamp";

    import ietf-inet-types
    {
        prefix "inet";
    }

    import ciena-ietf-twamp
    {
        prefix "twamp";
    }

    import ietf-interfaces
    {
        prefix "if";
    }

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
                 Hanover, Maryland 21076
                 U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
        "This YANG module defines Ciena's TWAMP
         configuration for the NETCONF protocol";

    revision "2016-01-20"
    {
        description
            "Update import of ciena-ietf-twamp";
    }

    revision "2014-02-23"
    {
        description
            "Initial version";
    }

    /*
     * typedefs
     */
    typedef twamp-hw-assist-state
    {
        type enumeration
        {
            enum off;
            enum on;
        }
```


Copyright© 2022 Ciena® Corporation

```
}

typedef twamp-admin-state
{
    type enumeration
    {
        enum disable;
        enum enable;
    }
}

/*
 * Identities
 */

/*
 * Features
 */

/*
 * Configuration data nodes
 */
augment "/twamp:twamp"
{
    container ciena-twamp
    {
        leaf admin-state
        {
            type twamp-admin-state;
          description "Enable or disable TWAMP as a whole on this device";
        }

        list port
        {
            key "port-ref";

            leaf port-ref
            {
                type if:interface-ref;
                description "physical port";
            }

            leaf admin-state
            {
                type twamp-admin-state;
                description "Enable or disable TWAMP on a port";
            }
        }
    }
}

augment "/twamp:twamp/twamp:twampServer"
{
    container ciena-twamp-server
    {
        leaf server-hw-assist
        {
            type twamp-hw-assist-state;
        }
```


Copyright© 2022 Ciena® Corporation

```
            list session-deny
            {
                key "ip-addr";

                leaf ip-addr
                {
                    type inet:ip-address;
                description "IP address of the interface to deny sessions for";
                }

                leaf port
                {
                    type if:interface-ref;
                    description "Ingress port on which the incoming TWAMP
packets with the specified IP address should be rejected";
                }
            }
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ciena-types.yang

```
module ciena-types {
  namespace "http://www.ciena.com/ns/yang/ciena-types";
  prefix "ciena-types";

  import ietf-inet-types {
    prefix inet;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module defines Ciena standard types and groupings.

     Copyright (c) 2016 Ciena Corporation.  All rights
     reserved.

     All information contained herein is, and remains
     the property of Ciena Corporation. Dissemination of this
     information or reproduction of this material is strictly
     forbidden unless prior written permission is obtained from
     Ciena Corporation.";

  revision 2017-09-12 {
    description
      "Shortened lines and reordered elements of the
       current-secret leaves to be in canonical order to
       comply with yang RFC standards.";
    reference
      "RFC 6020, Ciena Functional Specification.";
  }


  revision 2017-04-25 {
    description
      "Added preferred source address support.
       Added duration, simple-date, simple-time types.";
    reference
      "Ciena Functional Specification.";
  }

  revision 2016-05-11 {
    description
      "Initial version";
    reference
      "RFC 6020: YANG - A Data Modeling Language for
       the Network Configuration Protocol (NETCONF).
       No specific reference; standard not available.";
  }
```


Copyright© 2022 Ciena® Corporation

```
/*
 * Typedefs
 */
typedef state {
  type enumeration {
    enum disabled {
      description
      "The node is disabled.";
    }
    enum enabled {
      description
      "The node is enabled.";
    }
  }
  description
    "The state of the node.";
}

typedef admin-state {
  type state;
  description
    "The admin state of the node.";
}

typedef oper-state {
  type state;
  description
    "The oper state of the node.";
}

typedef on-off-state {
  type enumeration {
    enum off {
      description
        "The off state.";
    }
    enum on {
      description
        "The on state.";
    }
  }
  description
    "The on / off state.";
}

typedef event-severity {
  type enumeration {
    enum critical {
      description
        "The critical severity.";
    }
    enum major {
      description
        "The major severity.";
    }
    enum minor {
      description
        "The minor severity.";
    }
```


Copyright© 2022 Ciena® Corporation

```
      enum warning {
        description
          "The warning severity.";
      }
      enum config {
        description
          "The config severity.";
      }
      enum info {
        description
          "The info severity.";
      }
      enum debug {
        description
          "The debug severity.";
      }
    }
    description
      "The system event severity.";
  }

  typedef slot-index {
    type enumeration {
      enum non-existent {
        value -1;
        description
          "Non-existent slot.";
      }
      enum zero {
        value 0;
        description
          "Reserved slot number.";
      }
      enum lm1 {
        value 1;
        description
          "Slot LM1.";
      }
      enum lm2 {
        value 2;
        description
          "Slot LM2.";
      }
      enum lm3 {
        value 3;
        description
          "Slot LM3.";
      }
      enum lm4 {
        value 4;
        description
          "Slot LM4.";
      }
      enum lm5 {
        value 5;
        description
          "Slot LM5.";
      }
      enum lm6 {
        value 6;
```


Copyright© 2022 Ciena® Corporation

```
      description
        "Slot LM6.";
    }
    enum lm7 {
      value 7;
      description
        "Slot LM7.";
    }
    enum lm8 {
      value 8;
      description
        "Slot LM8.";
    }
    enum lm9 {
      value 9;
      description
        "Slot LM9.";
    }
    enum lm10 {
      value 10;
      description
        "Slot LM10.";
    }
    enum lm11 {
      value 11;
      description
        "Slot LM11.";
    }
    enum lm12 {
      value 12;
      description
        "Slot LM12.";
    }
    enum lm13 {
      value 13;
      description
        "Slot LM13.";
    }
    enum lm14 {
      value 14;
      description
        "Slot LM14.";
    }
    enum lm15 {
      value 15;
      description
        "Slot LM15.";
    }
    enum lm16 {
      value 16;
      description
        "Slot LM16.";
    }
    enum lm17 {
      value 17;
      description
        "Slot LM17.";
    }
    enum lm18 {
      value 18;
```


Copyright© 2022 Ciena® Corporation

```
    description
      "Slot LM18.";
  }
  enum lm19 {
    value 19;
    description
      "Slot LM19.";
  }
  enum lm20 {
    value 20;
    description
      "Slot LM20.";
  }
  enum lm21 {
    value 21;
    description
      "Slot LM21.";
  }
  enum lm22 {
    value 22;
    description
      "Slot LM22.";
  }
  enum lm23 {
    value 23;
    description
      "Slot LM23.";
  }
  enum lm24 {
    value 24;
    description
      "Slot LM24.";
  }
  enum lm25 {
    value 25;
    description
      "Slot LM25.";
  }
  enum lm26 {
    value 26;
    description
      "Slot LM26.";
  }
  enum lm27 {
    value 27;
    description
      "Slot LM27.";
  }
  enum lm28 {
    value 28;
    description
      "Slot LM28.";
  }
  enum lm29 {
    value 29;
    description
      "Slot LM29.";
  }
  enum lm30 {
    value 30;
```


Copyright© 2022 Ciena® Corporation

```
    description
      "Slot LM30.";
  }
  enum ctm1 {
    value 31;
    description
      "Slot CTM1 (CTX1).";
  }
  enum ctm2 {
    value 32;
    description
      "Slot CTM2 (CTX2).";
  }
  enum ctx1 {
    value 33;
    description
      "Slot CTX1. Non-existent alias sometimes used for ctm1.";
  }
  enum ctx2 {
    value 34;
    description
      "Slot CTX2. Non-existent alias sometimes used for ctm1.";
  }
  enum sm1 {
    value 35;
    description
     "Slot SM1. May be an alias for CTX1.sm depending on the platform.";
  }
  enum sm2 {
    value 36;
    description
     "Slot SM2. May be an alias for CTX2.sm depending on the platform.";
  }
  enum sm3 {
    value 37;
    description
      "Slot SM3. May be an alias for SM depending on the platform.";
  }
  enum sm4 {
    value 38;
    description
      "Slot SM4.";
  }
  enum sm5 {
    value 39;
    description
      "Slot SM5.";
  }
  enum pdu1 {
    value 40;
    description
      "Slot PDU1.  May be PDU-A1 or PDU-A depending of the platform.";
  }
  enum pdu2 {
    value 41;
    description
      "Slot PDU2.  May be PDU-B1 or PDU-B depending of the platform.";
  }
  enum pdu3 {
    value 42;
```


Copyright© 2022 Ciena® Corporation

```
        description
          "Slot PDU3.";
      }
      enum pdu4 {
        value 43;
        description
          "Slot PDU3.";
      }
      enum pdu5 {
        value 44;
        description
          "Slot PDU3.";
      }
      enum pdu6 {
        value 45;
        description
          "Slot PDU3.";
      }
      enum pdu7 {
        value 46;
        description
          "Slot PDU3.";
      }
      enum pdu8 {
        value 47;
        description
          "Slot PDU3.";
      }
      enum cfu1 {
        value 48;
        description
          "Slot CFU1.";
      }
      enum cfu2 {
        value 49;
        description
          "Slot CFU2.";
      }
      enum io {
        value 50;
        description
          "Slot IO.";
      }
      enum chassis {
        value 58;
        description
          "Chassis.";
      }
    }
    description
      "Chassis slot indexing";
  }

  typedef preferred-source-address {
     type enumeration {
        enum default {
           description
              "The default address for the device.";
        }
        enum loopback {
```


Copyright© 2022 Ciena® Corporation

```
          description
             "The loopback address configured on the device.";
       }
    }
    description
      "The preferred source address.";
}

typedef duration {
  type string {
    pattern
      'P(\d+Y)?(\d+M)?(\d+W)?(\d+D)?T(\d+H)?(\d+M)?(\d+S)?';
  }
  description
    "A duration of time specified in ISO 8601 format.";
  reference
    "ISO 8601";
}

typedef simple-date {
  type string {
    pattern
      '\d{4}-\d{2}-\d{2}';
  }
  description
    "A date specified in ISO 8601 format, YYYY-MM-DD.";
  reference
    "ISO 8601";
}

typedef simple-time {
  type string {
    pattern
      '\d{2}:\d{2}:\d{2}';
  }
  description
    "A time specified in ISO 8601 format, hh:mm:ss.";
  reference
    "ISO 8601";
}

/*
 * Identities
 */

/*
 * Features
 */
feature saos-6x {
  description
    "This feature string is returned by the
     netconf server if running on a 6.x device.";
}

feature saos-8x {
  description
    "This feature string is returned by the
     netconf server if running on a 8.x device.";
}
```


Copyright© 2022 Ciena® Corporation

```
/////
// These may need to go into their feature specific YANG.
/////
feature pm {
  description
    "This feature string is returned by the netconf
     server if PM is supported.";
}

feature protocol-filters {
  description
    "This feature string is returned by the netconf
     server if protocol filters are supported.";
}

feature pwe-module {
  description
    "This feature string is returned by the netconf
     server if the PWE module is supported.";
}

feature vpls-fpga {
  description
    "This feature string is returned by the netconf
     server if the VPLS FPGA is supported.";
}

feature pbt {
  description
    "This feature string is returned by the netconf
     server if PBT is supported.";
}

feature mpls {
  description
    "This feature string is returned by the netconf
     server if MPLS is supported.";
}

feature max-vc-l2xforms {
  description
    "This feature string is returned by the netconf
     server if max VC L2 transforms are supported.";
}

feature max-queue-groups {
  description
    "This feature string is returned by the netconf
     server if max queue groups are supported.";
}

feature raps {
  description
    "This feature string is returned by the netconf
     server if ring protection is supported.";
}

feature network-sync {
  description
    "This feature string is returned by the netconf
```


Copyright© 2022 Ciena® Corporation

```
      server if network sync is supported.";
  }

  feature bits-timing {
    description
      "This feature string is returned by the netconf
       server if bits timing is supported.";
  }

  feature gps-timing {
    description
      "This feature string is returned by the netconf
       server if gps timing is supported.";
  }

  feature ptp-timing {
    description
      "This feature string is returned by the netconf
       server if ptp timing is supported.";
  }

  feature benchmark {
    description
      "This feature string is returned by the netconf
       server if the benchmark feature is supported.";
  }

  feature benchmark-reflector {
    description
      "This feature string is returned by the netconf
       server if the benchmark reflector feature is supported.";
  }

  feature benchmark-generator {
    description
      "This feature string is returned by the netconf
       server if the benchmark generator feature is supported.";
  }

  feature benchmark-non-fpga-reflector {
    description
      "This feature string is returned by the netconf
       server if the benchmark non-FPGA reflector feature is supported.";
  }

  feature benchmark-non-fpga-generator {
    description
      "This feature string is returned by the netconf
       server if the benchmark non-FPGA generator feature is supported.";
  }

  ////
  // End feature specific
  ////

  /*
   * Groupings
   */

  grouping security-password {
```


Copyright© 2022 Ciena® Corporation

```
    description
      "Reusable grouping for security passwords and secrets.";

    choice security {
      description
        "Leafs for password or secret input.  These leafs will
         update the current-secret and may not return any data.";

      leaf password {
        type string;
        description
          "The string used for the clear text password.  No data will
           be returned from this leaf. This leaf will update the contents
           of the current-secret leaf.";
      }
      leaf secret {
        type string;
        description
          "The string used for the cipher text which may
           be encrypted or hashed. This leaf will update
           the contents of the current-secret leaf.";
      }
    }

    leaf current-secret {
      type string;
      config false;
      description
        "The current string used for the cipher
         text which may be encrypted or hashed.";
    }
  }

  grouping security-key {
    description
      "Reusable grouping for security keys and secrets.";

    choice security {
      description
        "Leafs for key or secret input.  These leafs will
         update the current-secret and may not return any data.";

      leaf key {
        type string;
        description
          "The string used for the clear text key.  No data will
           be returned from this leaf.
           This leaf will update the contents of the current-secret leaf.";
      }
      leaf secret {
        type string;
        description
          "The string used for the cipher text which may
           be encrypted or hashed. This leaf will update
           the contents of the current-secret leaf.";
      }
    }

    leaf current-secret {
      type string;
```


Copyright© 2022 Ciena® Corporation

```
      config false;
      description
        "The current string used for the cipher text
         which may be encrypted or hashed.";
    }
  }

  // Hostname and IP resolve.
  grouping resolved-address {
    description
      ".";
    leaf hostname {
      type inet:host;
      description
        "Hostname of the device.";
    }
    leaf address {
      type inet:ip-address;
      description
        "Resolved IP address of the device.";
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-users.yang

```
module ciena-users {
   namespace "http://www.ciena.com/ns/yang/ciena-users";
   prefix "users";

   import ciena-types {
      prefix ciena;
   }

   import ietf-system {
      prefix sys;
   }

   organization
      "Ciena Corporation";

   contact
      "Web URL: http://www.ciena.com/
      E-mail:  yang@ciena.com
      Postal:  7035 Ridge Road
               Hanover, Maryland 21076
               U.S.A.
      Phone:   +1 800-921-1144
      Fax:     +1 410-694-5750";

   description
     "This YANG module defines Ciena's global configuration for user settings.

      Copyright (c) 2016 Ciena Corporation.  All rights
      reserved.

      All information contained herein is, and remains
      the property of Ciena Corporation. Dissemination of this
      information or reproduction of this material is strictly
      forbidden unless prior written permission is obtained from
      Ciena Corporation.";

   revision 2017-04-25 {
      description
         "Add user accounting leaf.";
   }

   revision 2016-06-14 {
      description
         "Initial version";
      reference
         "RFC 6020: YANG - A Data Modeling Language for
         the Network Configuration Protocol (NETCONF).
         No specific reference; standard not available.";
   }

   /*
    * typedefs
    */

   typedef user-access-level {
      type enumeration {
         enum limited {
            description
```


Copyright© 2022 Ciena® Corporation

```
                "User with only limited access to device commands.";
            }
            enum admin {
               description
                  "User with only admin access to device commands.";
            }
            enum super {
               description
                  "User with only super access to device commands.";
            }
            enum diag {
               description
                  "User with complete access to device commands.";
            }
         }

         description
            "Access level of the user.";

      }

      typedef user-authentication-method {
         type enumeration {
            enum none {
               description
                  "Use no method for user authentication.";
            }
            enum local {
               description
                  "Use local authorization for user authentication.";
            }
            enum radius {
               description
                  "Use RADIUS for user authentication.";
            }
            enum tacacs {
               description
                  "Use TACACS for authentication.";
            }
       enum radsec {
               description
                  "Use RADSEC for authentication.";
            }
         }

         description
            "Enumeration describing the methods for user authentication.";

      }

      typedef user-authentication-scope {

         type enumeration {
            enum all {
               description
                  "This applies to all interfaces.";
            }
            enum serial {
               description
                  "This only applies to serial interfaces.";
```


Copyright© 2022 Ciena® Corporation

```
        }
        enum remote {
           description
              "This only applies to remote interfaces.";
        }
     }

     description
        "Enumeration describing the scope of the authentication changes.";

  }

  typedef user-authentication-priority {
     type uint16 {
        range "1 .. 3";
     }
     description
        "Authentication priority of the user authorization providers.";
  }

  typedef user-accounting-method {
     type enumeration {
        enum default {
           description
              "Use the default method for user accounting.";
        }
        enum radius {
           description
              "Use RADIUS for user accounting.";
        }
        enum tacacs {
           description
              "Use TACACS for user accounting.";
        }
        enum radsec {
           description
              "Use Radsec for user accounting.";
        }
     }
     description
        "Enumeration describing the methods for user accounting.";
  }

  typedef username-string {
     type string {
        length "1 .. 32";
     }
     description
        "A valid system user name string.";
  }

  /*
   * Identities
   */

  /*
   * Features
   */

  /*
```


Copyright© 2022 Ciena® Corporation

```
    * Configuration data nodes
    */
   augment /sys:system {
      description
         "SAOS's concept of user authentication configuration.";

      container authentication {
         description
            "Configuration of the users and their account settings.";

         list user {
            key name;
            description
               "The list of users configured on this device.";

            leaf name {
               type username-string;
               description
                  "The user name identifying this entry.";
            }

            uses ciena:security-password;

            leaf access-level {
               type user-access-level;
               description
                  "Access level of the current user.";
            }

            leaf max-session-limit {
               type uint16 {
                  range "0 .. 12";
               }
               description
                  "Maximum simultaneous user logins.";
            }
         }

         list authentication {
            key priority;
            description
               "User authentication settings and their priorities.";

            leaf priority {
               type user-authentication-priority;
               description
                  "The Authorization priority.";
            }

            leaf method {
               type user-authentication-method;
               description
                  "Set the authentication method for this priority.";
            }

            leaf scope {
               type user-authentication-scope;
               description
                  "Set the scope for this authentication method/priority.";
            }
```


Copyright© 2022 Ciena® Corporation

```
         }

         leaf max-limited-users {
            type uint16 {
               range "0 .. 9";
            }
            description
               "Maximum number of limited users supported by this system.";
         }

         leaf max-admin-users {
            type uint16 {
               range "0 .. 10";
            }
            description
               "Maximum number of admin users supported by this system.";
         }

         leaf max-super-users {
            type uint16 {
               range "1 .. 10";
            }
            description
               "Maximum number of super users supported by this system.";
         }

         container password-policy {
            description
               "Rules about how users are allowed to create passwords.";

            leaf disallow-dict-words {
               type ciena:on-off-state;
               description
                  "Disallow passwords containing words in the dictionary.";
            }

            leaf disallow-username {
               type ciena:on-off-state;
               description
                "Disallow passwords containing the user name or its reverse.";
            }

            leaf max-repeated-chars {
               type uint16 {
                  range "0 .. 128";
               }
               description
               "Maximum run of the same character in user account passwords.";
            }

            leaf min-character-change {
               type uint16 {
                  range "0 .. 128";
               }
               description
                  "Minimum number of characters in the password that must
change.";
            }

            leaf min-length {
```


Copyright© 2022 Ciena® Corporation

```
               type uint16 {
                  range "0 .. 128";
               }
               description
                  "Minimum length of the user account passwords.";
            }

            leaf min-lowercase-chars {
               type uint16 {
                  range "0 .. 128";
               }
               description
                  "Minimum number of lower case letters in the user account
passwords.";
            }

            leaf min-numeric-chars {
               type uint16 {
                  range "0 .. 128";
               }
               description
                  "Minimum number of numeric characters in the user account
passwords.";
            }

            leaf min-special-chars {
               type uint16 {
                  range "0 .. 128";
               }
               description
                  "Minimum number of special characters in the user account
passwords, such as !@#$%^*().";
            }

            leaf min-uppercase-chars {
               type uint16 {
                  range "0 .. 128";
               }
               description
                  "Minimum number of upper case letters in the user account
passwords.";
            }
         }

         container username-policy {
            description
               "Rules about how users are allowed to create user names.";

            leaf min-length {
               type uint16 {
                  range "1 .. 32";
               }
               description
                  "Minimum length of a username.";
            }
         }

         leaf user-accounting-method {
            type user-accounting-method;
            description
```


Copyright© 2022 Ciena® Corporation

```
                "Set the accounting method for users.";
            }
        }
    }

    /*
     * Operational state data nodes
     */
    augment /sys:system-state {
        description
            "SAOS's concept of user authentication operational data.";

        container authentication {
            config false;

            description
                "Operational data for this systems user management.";

            list active-users {
                key name;
                description
                    "The list of users currently on this device.";

                leaf name {
                    type username-string;
                    description
                        "The user name identifying this entry.";
                }

                leaf pid {
                    type uint32;
                    description
                        "The process ID of the current session";
                }

                leaf terminal {
                    type string;
                    description
                       "The name of the terminal the current session is logged on
through.";
                }

                leaf idle-time {
                    type uint32;
                    description
                     "Amount of time in minutes that this session has been idle.";
                }
            }

            list user-accounting {
                key name;
                description
                    "List of all the active user's sessions";

                leaf name {
                    type username-string;
                    description
                        "The user name identifying this entry.";
                }
```


Copyright© 2022 Ciena® Corporation

```
            leaf privilege {
               type user-access-level;
               description
                  "Access level of this user.";
            }

            leaf current-sessions {
               type uint16;
               description
                  "The number of active sessions this user has currently.";
            }

            leaf telnet-sessions {
               type uint16;
               description
              "The number of active telnet sessions this user has currently.";
            }

            leaf ssh-sessions {
               type uint16;
               description
                "The number of active ssh sessions this user has currently.";
            }

            leaf session-limit {
               type uint16;
               description
                  "The number of active sessions this user is limited to.";
            }
         }


         list account-status {
            key privilege;
            description
               "List of account levels with their current use counts.";

            leaf privilege {
               type user-access-level;
               description
                  "User access level.";
            }

            leaf current-active-network {
               type uint16;
               description
                 "Users of this access level currently logged on through the
network.";
            }

            leaf current-active-serial {
               type uint16;
               description
                  "Users of this access level currently logged on through
serial.";
            }

         }

         list authorization-provider-stats {
```


Copyright© 2022 Ciena® Corporation

```
            key priority;
            description
               "Statistics based on authentication success/failure sorted by
authentication priority.";

            leaf priority {
               type user-authentication-priority;
               description
                  "Priority of the authorization method.";
            }

            leaf method {
               type user-authentication-method;
               description
                  "Method of authentication.";
            }

            leaf called {
               type uint16;
               description
                  "Number of times this authentication method was invoked.";
            }

            leaf success {
               type uint16;
               description
                "Number of times this authentication method was successful.";
            }

            leaf failure {
               type uint16;
               description
                  "Number of times this authentication method failed.";
            }

            leaf skipped {
               type uint16;
               description
                  "Number of times this authentication method was skipped.";
            }

            leaf scope {
               type user-authentication-scope;
               description
                  "Scope of this authentication method.";
            }
         }

         leaf total-active-users {
            type uint16;
            description
               "Total number of current active users.";
         }
      }
   }
   /*
    * RPCs
    */

   rpc write {
```


Copyright© 2022 Ciena® Corporation

```
      description
         "Send a message to another user's CLI session.";

      input {
         leaf username {
            type username-string;
            description
               "User name of the message recipient.";
         }
         leaf message {
            type string;
            description
               "Message content to send to the user's CLI session.";
         }
      }
      output {
         leaf status {
            type string;
            description
               "Status of the user-kill operation.";
         }
      }
   }

   rpc kill {
      description
         "Kills all sessions linked to the current user.";

      input {
         leaf pid {
            type uint16;
            description
               "Process ID (pid) of the user to kill.";
         }
      }
      output {
         leaf status {
            type string;
            description
               "Status of the user-kill operation.";
         }
      }
   }

   rpc kill-all-but-me {
      description
         "Kills all sessions except for the current session.";
      output {
         leaf status {
            type string;
            description
               "Status of the user-kill operation.";
         }
      }
   }
}
```


Copyright© 2022 Ciena® Corporation

ciena-vlli.yang

```
module ciena-vlli {
  namespace "http://www.ciena.com/ns/yang/vlli";
  prefix "vlli";

  import ietf-interfaces {
    prefix "if";
  }

  import mef-cfm {
    prefix "cfm";
    revision-date 2012-04-16;
  }

  import ciena-types {
    prefix ciena;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This module contains the YANG definition of Virtual
     Link Loss Indication (VLLI) parameters as per
     ciena-vlli-functional-specification.docx.

     Copyright (c) 2016 Ciena Corporation.  All rights
     reserved.

     All information contained herein is, and remains
     the property of Ciena Corporation. Dissemination of this
     information or reproduction of this material is strictly
     forbidden unless prior written permission is obtained from
     Ciena Corporation.";

  revision "2018-08-09" {
    description
      "Mac Flush action added.";
    reference
      "ciena-vlli-functional-specification.docx";
  }

  revision "2017-04-10" {
    description
      "Updated the following parameters :
        - Removed the container config and global-state.
        - Updated naming convention for VLLI global state admin-status to
global-admin-state.
        - Updated naming convention for instance group leaf state to admin-
state.
```


Copyright© 2022 Ciena® Corporation

```
        - Updated naming convention for operational data";
    reference
      "ciena-vlli-functional-specification.docx";
  }

  revision "2016-02-18" {
    description
      "Initial Version.";
    reference
      "ciena-vlli-functional-specification.docx";
  }

  /*
   * Type definitions
   */

  typedef last-event {
    type enumeration {
      enum unknown {
        description "No defined previous event.";
      }
      enum external-fault {
        description "The fault event external to VLLI.";
      }
      enum external-recovery {
        description "The recovery event external to VLLI.";
      }
      enum admin-fault {
        description "This is a VLLI ADMIN fault event.";
      }
      enum admin-recovery {
        description "This is a VLLI ADMIN recovery event.";
      }
    }
    description
      "The last event for each Instance.";
  }

  identity action-type {
    description
      "Base identity for VLLI action types.";
  }

  identity fault {
    base action-type;
    description
      "Defines FAULT action.";
  }

  identity recovery {
    base action-type;
    description
      "Defines RECOVERY action.";
  }

  identity action-name {
    description
      "Base identity for set of VLLI actions.";
  }
```


Copyright© 2022 Ciena® Corporation

```
identity  none {
  base action-name;
  description
    "The action is none.";
}

identity port-shut {
  base action-name;
  description
    "The action is interface down.";
}

identity port-unshut {
  base action-name;
  description
    "The action is interface up.";
}

identity ccm-stop {
  base action-name;
  description
    "The action is Continuity Check Message Stop.";
}

identity ccm-resume {
  base action-name;
  description
    "The action is Continuity Check Message Resume.";
}

identity mac-flush {
  base action-name;
  description
    "The action is Mac Flush on a port.";
}

grouping action-group-reference {
  description
    "This grouping uniquely identifies an action group.";

  leaf action-group {
    type leafref {
      path "/vlli:vlli/vlli:action-group/vlli:name";
    }
    description
      "A reference to a specific Action group.";
  }
}

grouping instance-group-reference {
  description
    "This grouping uniquely identifies an instance group.";

  leaf instance-group {
    type leafref {
      path "/vlli:vlli/vlli:instance-group/vlli:name";
    }
    description
      "A reference to a specific Instance group.";
  }
```


Copyright© 2022 Ciena® Corporation

```
}

/*
 * Data node definitions
 */

container vlli {
  description
    "Top level node which has both configuration and
    state nodes.";

  leaf global-admin-state {
    type ciena:admin-state;
     default disabled;
    description
      "Specifies if VLLI is enabled or disabled globally.
       The VLLI admin-state always starts in the disabled
       state until an explicit management action or
       configuration information is retained by the system
       and causes a transition to the enabled state";
  }

  list action-group {
    key "name";
    description
      "The list of VLLI action groups.";

    leaf name {
      type string {
        length "1..15";
      }
      description
        "The unique identifier of action group entry.";
    }

    list action {
      key "type precedence";
      description
        "The list of actions that an action-group can have.";

      leaf type {
        type identityref {
          base action-type;
        }
        description
          "Defines the type of VLLI action.";
      }

      leaf precedence {
        type uint8 {
          range 1..8;
        }
        description
          "The precedence number associated with each action.";
      }

      leaf action {
        type identityref {
          base action-name;
        }
```


Copyright© 2022 Ciena® Corporation

```
        description
          "Defines the actual action.";
      }
    }
  }

  list instance-group {
    key "name";
    description
      "The list of VLLI instance groups.";

    leaf name {
      type string {
        length "1..45";
      }
      description
        "A name is associated with group.";
    }

    leaf direction {
      type enumeration {
        enum undirectional {
          description
            "The action will take place only from source to
             destination direction.";
        }
        enum bidirectional {
          description
            "The action will take place in either direction
             from source to destination or destination to
             source.";
        }
      }
      default undirectional;
      description
        "Defines the direction of fault/recovery action between
         source and destination.";
    }

    leaf trigger {
      type enumeration {
        enum any {
          description
            "Any action that happens on either side of source
             or destination instance will trigger the same action
             on the other instance.";
        }
        enum all {
          description
            "The instance will get triggered only when
             all actions get triggered on other side.";
        }
      }
      default any;
      description
        "any -single source can trigger action.
         all - all sources should be in the same state
               to trigger action.";
    }
```


Copyright© 2022 Ciena® Corporation

```
leaf force {
  type enumeration {
    enum none {
      description
        "No force status.";
    }
    enum fault {
      description
        "It indicates whether it is VLLI force fault or
         the actual fault.";
    }
    enum recovery {
      description
        "It indicates whether it is VLLI force recovery or
         the actual recovery.";
    }
  }
  default none;
  description
    "It indicates about VLLI action or external.";
}

leaf admin-state {
  type ciena:admin-state;
   default disabled;
  description
    "Specifies if instance group is enabled or disabled.
     The VLLI admin-state always starts in the disabled
     state until an explicit management action or
     configuration information is retained by the system
     and causes a transition to the enabled state";
}

list vlli-instance {
  key "mode";
  description
    "The list of instances in an instance group.";

  leaf mode {
    type enumeration {
      enum destination {
        description
          "Defines the instance to be destination instance.";
      }
      enum source {
        description
          "Defines the instace to be source instance.";
      }
    }
    description
      "The mode of instance whether it is applicable at
      source or at destination.";
  }

  list port-instance {
    key "name";
    description
      "The list of port instances in an instance group.";

    leaf name {
```


Copyright© 2022 Ciena® Corporation

```
            type if:interface-ref;
            description
              "The name of the port that is used as an unique
              identifier.";
          }

          uses action-group-reference;

        }

        list cfm-instance {
           key "maintenance-domain maintenance-association";
           description
              "The list of cfm instances in an instance group.";

          uses cfm:maintenance-association-reference;

           uses action-group-reference;

        }
      }
    }
  }

  /*
   * Operational state data nodes
   */

  container vlli-state {
    config false;
    description
      "Data nodes for the operational state of instance groups.";

    list instance-group {
      key "name";
      description
        "The list of VLLI instance groups.";

      leaf name {
        type string {
          length "1..45";
        }
        description
          "Name of the group.";
      }

      leaf recovery-destination-counter {
        type uint32;
        description
          "The number of recoveries on destination side of
          an instance group.";
      }

      leaf fault-destination-counter {
        type uint32;
        description
          "The number of faults on destination side of
          an instance group.";
      }
```


Copyright© 2022 Ciena® Corporation

```
leaf recovery-source-counter {
  type uint32;
  description
    "The number of recoveries on source side of
    an instance group.";
}

leaf fault-source-counter {
  type uint32;
  description
    "The number of faults on source side of
    an instance group.";
}

list vlli-instance {
  key "mode";
  description
    "The list of instances in an instance group.";

  leaf mode {
    type enumeration {
      enum destination {
        description
          "Defines the instance to be destination instance.";
      }
      enum source {
        description
          "Defines the instace to be source instance.";
      }
    }
    description
      "The mode of instance whether it is applicable at
      source or at destination.";
  }

  list port-instance {
    key "name";
    description
      "The list of port instances in an instance group.";

    leaf name {
      type if:interface-ref;
      description
        "The name of the port.";
    }

    leaf status {
      type last-event;
      description
        "The last event occured on this instance.";
    }
  }

  list cfm-instance {
     key "maintenance-domain maintenance-association";
     description
        "The list of cfm instances in an instance group.";

     uses cfm:maintenance-association-reference;
```


Copyright© 2022 Ciena® Corporation

```
          leaf status {
             type last-event;
             description
                "The last event occured on this instance.";
          }
        }
      }
    }
  }

  //
  // RPCs
  //

  rpc clear-counter {
    description
      "Administratively clears the counters of an instance group.";

    input {
      uses instance-group-reference;
    }
  }

  //
  // Notifications
  //

  notification vlli-event {
    description
      "A notification of a specific intance about fault/recovery
       action.";

    container event {
      description
        "Data definitions related to event on a specific instance.";

      leaf group-name {
        type string {
          length "1..45";
        }
        description
          "The instance group name.";
      }

      leaf instance-name {
        type string {
          length "1..45";
        }
        description
          "This is a port or cfm instance of the instance group.";
      }

      leaf action-category {
        type identityref {
          base action-type;
        }
        description
          "The VLLI action type i.e., either fault or recovery.";
      }
```


Copyright© 2022 Ciena® Corporation

```
      leaf action {
        type identityref {
          base action-name;
        }
        description
          "The actual VLLI action that occured on the instance.";
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ciena-xftp.yang

```
module ciena-xftp {

  namespace "http://www.ciena.com/ns/yang/ciena-xftp";
  prefix xftp;

  import ietf-inet-types {
    prefix inet;
  }

  import ciena-types {
    prefix ciena;
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module defines the configuration data, state data and
     RPCs (remote procedure calls) for Ciena system file-transfers.

     Copyright (c) 2016-2017 Ciena Corporation.  All rights reserved.

     All information contained herein is, and remains the property of
     Ciena Corporation.  Dissemination of this information or
     reproduction of this material is strictly forbidden unless prior
     written permission is obtained from Ciena Corporation.";

  revision 2017-08-30 {
    description
      "Removed the sftp-client algorithm-settings container.";
    reference
      "";
  }

  revision 2017-08-03 {
    description
      "Cleaned up nomenclature and inconsistencies.";
    reference
      "No specific reference; standard not available.";
  }

  revision 2016-05-12 {
    description
      "Initial version";
    reference
      "RFC 6020: YANG - A Data Modeling Language for
       the Network Configuration Protocol (NETCONF).
       No specific reference; standard not available.";
  }
```

```
/*
 * typedefs
 */

typedef protocol {
  type enumeration {
    enum tftp {
      description
        "TFTP (Trivial File Transfer Protocol)";
    }
    enum ftp {
      description
        "FTP (File Transfer Protocol)";
    }
    enum sftp {
      description
        "SFTP (SSH File Transfer Protocol or Secure File Transfer
         Protocol)";
    }
    enum xftp {
      description
        "xFTP, the system's default file transfer protocol";
    }
    enum http {
      description
        "HTTP (HTTP (Hypertext Transfer Protocol)";
    }
  }
  description
    "The file transfer protocol in use.";
}

typedef compression-algorithm {
  type enumeration {
    enum none {
      description
        "Placeholder for algorithms.";
    }
  }
  description
    "Compression algorithms.";
}

/*
 * groupings
 */

grouping server-config {
  leaf protocol {
    type protocol;
    description
      "The file transfer protocol used by the server.";
  }

  leaf address {
    type string {
      length 0..253;
    }
    description
```


Copyright© 2022 Ciena® Corporation

```
        "The server IP address or hostname to be used for the selected
         server.";
    }

    leaf port {
      type inet:port-number;
      description
        "The server port number to be used for the selected server.";
    }

    leaf login-id {
      when "../protocol != 'tftp'" {
        description
          "Login ID does not apply to TFTP.";
      }
      type string {
        length 0..32;
      }
      description
        "Login ID for the specified server.";
    }

    leaf password {
      when "../protocol != 'tftp'" {
        description
          "Password does not apply to TFTP.";
      }
      type string;
      description
        "The string used for the clear text password.  No data will be
         returned from this leaf.  This leaf will update the contents
         of the current-secret leaf.";
    }

    leaf secret {
      when "../protocol != 'tftp'" {
        description
          "Secret does not apply to TFTP.";
      }
      type string;
      description
        "The string used for the cipher text which may be encrypted or
         hashed.  This leaf will update the contents of the
         current-secret leaf.";
    }

    description
      "XFTP server configuration options.";
  }

  grouping server-settings {
    choice destination-server{

      description
        "A selection between the default server options already
         configured and the specific server along with the server's IP
         address or hostname.";

      case default-server {
        description
```


Copyright© 2022 Ciena® Corporation

```
          "Choose server type for file transfer.  This server will use
           a pre-configured IP address or hostname.";
        leaf default-server-type {
          type protocol;
          mandatory true;
          description
            "Select the server type and the pre-configured IP address or
             hostname will be used.";
        }
      }

      case server {
        description
          "Choose a server, but the IP address or hostname must be
           specified.";

        uses server-config;
      }

      case url {
        description
          "Select the URL for the file transfer operation.";
        leaf url-address {
          type string;
          description
            "The URL specified for the file transfer.";
        }
      }
    }
    description
      "Select the server options for the specified file transfer.";
  }

  /*
   * Identities
   */

  /*
   * Features
   */

  /*
   * Configuration data nodes
   */

  container xftp-config {
    description
      "Configuration of the system file transfer properties.";

    list client {
      key protocol;
      description
        "File transfer client settings.";

      leaf protocol {
        type protocol;
        description
          "File transfer protocol used by this client.";
      }
```


Copyright© 2022 Ciena® Corporation

```
    leaf admin-state {
      type ciena:admin-state;
      description
        "Administrative state of this client.";
    }
  }

  list server {
    key protocol;
    description
      "The configuration properties for the system's servers.";

    uses server-config;
  }
}

/*
 * Operational state data nodes
 */

container xftp-state {
  config false;

  description
    "Operational data for the system's file transfer settings.";

  list server {
    key protocol;
    description
      "The operational properties for the system's servers.";

    leaf protocol {
      type protocol;
      description
        "Settings for the selected server type.";
    }

    leaf resolved-address {
      type inet:ip-address;
      description
        "The resolved xftp server address.";
    }

    leaf current-secret {
      type string;
      config false;
      description
        "The current string used for the cipher text which may be
         encrypted or hashed.";
    }
  }
}

/*
 * RPCs
 */

rpc xftp-transfer {
  description
    "Transfer a file to or from a remote system.";
```


Copyright© 2022 Ciena® Corporation

```
    input {
      leaf action {
        type enumeration {
          enum download {
            description
              "Specify the download action.  The server retrieves
               the file identified by the remote-file-path
               to the location specified by the local-file-path.";
          }
          enum upload {
            description
              "Specify the upload action.  The server sends
               the file identified by the local-file-path
               to the URI specified by the remote-file-path.";
          }
        }
        mandatory true;
        description
          "Transfer action to be performed:  download or upload.";
      }

      leaf remote-file-uri {
        type inet:uri;
        mandatory true;
        description
          "URI (Universal Resource Identifier) of the remote file.";
      }

      leaf local-file-path {
        type string;
        description
          "Name of the file on the local system.";
      }

      leaf login-id {
        type string {
          length 0..32;
        }
        description
          "Login ID for the specified server.  Not needed for TFTP.";
      }

      leaf password {
        type string;
        description
          "The string used for the clear text password.  Not needed for
           TFTP.";
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

Standard YANG model schemas

This chapter provides the Standard schemas for YANG models used with 39XX/51XX switches.

iana-crypt-hash@2014-08-06.yang

```
module iana-crypt-hash {
  namespace "urn:ietf:params:xml:ns:yang:iana-crypt-hash";
  prefix ianach;

  organization "IANA";
  contact
    "         Internet Assigned Numbers Authority

     Postal: ICANN
             12025 Waterfront Drive, Suite 300
             Los Angeles, CA  90094-2536
             United States

     Tel:    +1 310 301 5800
     E-Mail: iana@iana.org>";
  description
    "This YANG module defines a type for storing passwords
     using a hash function and features to indicate which hash
     functions are supported by an implementation.

     The latest revision of this YANG module can be obtained from
     the IANA web site.

     Requests for new values should be made to IANA via
     email (iana@iana.org).

     Copyright (c) 2014 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject
     to the license terms contained in, the Simplified BSD License
     set forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     The initial version of this YANG module is part of RFC 7317;
     see the RFC itself for full legal notices.";

  revision 2014-08-06 {
```


Copyright© 2022 Ciena® Corporation

```
    description
      "Initial revision.";
    reference
      "RFC 7317: A YANG Data Model for System Management";
  }

  typedef crypt-hash {
    type string {
      pattern
        '$0$.*'
      + '|$1$[a-zA-Z0-9./]{1,8}$[a-zA-Z0-9./]{22}'
      + '|$5$(rounds=\d+$)?[a-zA-Z0-9./]{1,16}$[a-zA-Z0-9./]{43}'
      + '|$6$(rounds=\d+$)?[a-zA-Z0-9./]{1,16}$[a-zA-Z0-9./]{86}';
    }
    description
      "The crypt-hash type is used to store passwords using
       a hash function.  The algorithms for applying the hash
       function and encoding the result are implemented in
       various UNIX systems as the function crypt(3).

       A value of this type matches one of the forms:

         $0$<clear text password>
         $<id>$<salt>$<password hash>
         $<id>$<parameter>$<salt>$<password hash>

       The '$0$' prefix signals that the value is clear text.  When
       such a value is received by the server, a hash value is
       calculated, and the string '$<id>$<salt>$' or
       $<id>$<parameter>$<salt>$ is prepended to the result.  This
       value is stored in the configuration data store.
       If a value starting with '$<id>$', where <id> is not '0', is
       received, the server knows that the value already represents a
       hashed value and stores it 'as is' in the data store.

       When a server needs to verify a password given by a user, it
       finds the stored password hash string for that user, extracts
       the salt, and calculates the hash with the salt and given
       password as input.  If the calculated hash value is the same
       as the stored value, the password given by the client is
       accepted.

       This type defines the following hash functions:

         id | hash function | feature
         ---+---------------+-------------------
          1 | MD5           | crypt-hash-md5
          5 | SHA-256       | crypt-hash-sha-256
          6 | SHA-512       | crypt-hash-sha-512

       The server indicates support for the different hash functions
       by advertising the corresponding feature.";
    reference
      "IEEE Std 1003.1-2008 - crypt() function
       RFC 1321: The MD5 Message-Digest Algorithm
       FIPS.180-4.2012: Secure Hash Standard (SHS)";
  }

  feature crypt-hash-md5 {
    description
```


Copyright© 2022 Ciena® Corporation

```
      "Indicates that the device supports the MD5
       hash function in 'crypt-hash' values.";
    reference "RFC 1321: The MD5 Message-Digest Algorithm";
  }

  feature crypt-hash-sha-256 {
    description
      "Indicates that the device supports the SHA-256
       hash function in 'crypt-hash' values.";
    reference "FIPS.180-4.2012: Secure Hash Standard (SHS)";
  }

  feature crypt-hash-sha-512 {
    description
      "Indicates that the device supports the SHA-512
       hash function in 'crypt-hash' values.";
    reference "FIPS.180-4.2012: Secure Hash Standard (SHS)";
  }

}
```


Copyright© 2022 Ciena® Corporation

iana-if-type.yang

```
module iana-if-type {
  namespace "urn:ietf:params:xml:ns:yang:iana-if-type";
  prefix ianaift;

  import ietf-interfaces {
    prefix if;
  }

  organization "IANA";
  contact
    "        Internet Assigned Numbers Authority

     Postal: ICANN
             4676 Admiralty Way, Suite 330
             Marina del Rey, CA 90292

     Tel:    +1 310 823 9358
     <mailto:iana@iana.org>";
  description
    "This YANG module defines YANG identities for IANA-registered
     interface types.

     This YANG module is maintained by IANA and reflects the
     'ifType definitions' registry.

     The latest revision of this YANG module can be obtained from
     the IANA web site.

     Requests for new values should be made to IANA via
     email (iana@iana.org).

     Copyright (c) 2014 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject
     to the license terms contained in, the Simplified BSD License
     set forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     The initial version of this YANG module is part of RFC 7224;
     see the RFC itself for full legal notices.";
    reference
      "IANA 'ifType definitions' registry.
       <http://www.iana.org/assignments/smi-numbers>";

  revision 2014-05-08 {
    description
      "Initial revision.";
    reference
      "RFC 7224: IANA Interface Type YANG Module";
  }

  identity iana-interface-type {
    base if:interface-type;
    description
      "This identity is used as a base for all interface types
```


Copyright© 2022 Ciena® Corporation

```
      defined in the 'ifType definitions' registry.";
  }




  identity other {
    base iana-interface-type;
  }
  identity regular1822 {
    base iana-interface-type;
  }
  identity hdh1822 {
    base iana-interface-type;
  }
  identity ddnX25 {
    base iana-interface-type;
  }
  identity rfc877x25 {
    base iana-interface-type;
    reference
      "RFC 1382 - SNMP MIB Extension for the X.25 Packet Layer";
  }
  identity ethernetCsmacd {
    base iana-interface-type;
    description
      "For all Ethernet-like interfaces, regardless of speed,
       as per RFC 3635.";
    reference
      "RFC 3635 - Definitions of Managed Objects for the
                  Ethernet-like Interface Types";
  }
  identity iso88023Csmacd {
    base iana-interface-type;
    status deprecated;
    description
      "Deprecated via RFC 3635.
       Use ethernetCsmacd(6) instead.";
    reference
      "RFC 3635 - Definitions of Managed Objects for the
                  Ethernet-like Interface Types";
  }
  identity iso88024TokenBus {
    base iana-interface-type;
  }
  identity iso88025TokenRing {
    base iana-interface-type;
  }
  identity iso88026Man {
    base iana-interface-type;
  }
  identity starLan {
    base iana-interface-type;
    status deprecated;
    description
      "Deprecated via RFC 3635.
       Use ethernetCsmacd(6) instead.";
    reference
```


Copyright© 2022 Ciena® Corporation

```
      "RFC 3635 - Definitions of Managed Objects for the
                  Ethernet-like Interface Types";
  }
  identity proteon10Mbit {
    base iana-interface-type;
  }
  identity proteon80Mbit {
    base iana-interface-type;
  }
  identity hyperchannel {
    base iana-interface-type;
  }
  identity fddi {
    base iana-interface-type;
    reference
      "RFC 1512 - FDDI Management Information Base";
  }
  identity lapb {
    base iana-interface-type;
    reference
      "RFC 1381 - SNMP MIB Extension for X.25 LAPB";
  }
  identity sdlc {
    base iana-interface-type;
  }
  identity ds1 {
    base iana-interface-type;
    description
      "DS1-MIB.";
    reference
      "RFC 4805 - Definitions of Managed Objects for the
                  DS1, J1, E1, DS2, and E2 Interface Types";
  }
  identity e1 {
    base iana-interface-type;
    status obsolete;
    description
      "Obsolete; see DS1-MIB.";
    reference
      "RFC 4805 - Definitions of Managed Objects for the
                  DS1, J1, E1, DS2, and E2 Interface Types";
  }


  identity basicISDN {
    base iana-interface-type;
    description
      "No longer used.  See also RFC 2127.";
  }
  identity primaryISDN {
    base iana-interface-type;
    description
      "No longer used.  See also RFC 2127.";
  }
  identity propPointToPointSerial {
    base iana-interface-type;
    description
      "Proprietary serial.";
  }
  identity ppp {
```


Copyright© 2022 Ciena® Corporation

```
    base iana-interface-type;
  }
  identity softwareLoopback {
    base iana-interface-type;
  }
  identity eon {
    base iana-interface-type;
    description
      "CLNP over IP.";
  }
  identity ethernet3Mbit {
    base iana-interface-type;
  }
  identity nsip {
    base iana-interface-type;
    description
      "XNS over IP.";
  }
  identity slip {
    base iana-interface-type;
    description
      "Generic SLIP.";
  }
  identity ultra {
    base iana-interface-type;
    description
      "Ultra Technologies.";
  }
  identity ds3 {
    base iana-interface-type;
    description
      "DS3-MIB.";
    reference
      "RFC 3896 - Definitions of Managed Objects for the
                  DS3/E3 Interface Type";
  }
  identity sip {
    base iana-interface-type;
    description
      "SMDS, coffee.";
    reference
      "RFC 1694 - Definitions of Managed Objects for SMDS
                  Interfaces using SMIv2";
  }
  identity frameRelay {
    base iana-interface-type;
    description
      "DTE only.";
    reference
      "RFC 2115 - Management Information Base for Frame Relay
                  DTEs Using SMIv2";
  }
  identity rs232 {
    base iana-interface-type;
    reference
      "RFC 1659 - Definitions of Managed Objects for RS-232-like
                  Hardware Devices using SMIv2";
  }
  identity para {
    base iana-interface-type;
```


Copyright© 2022 Ciena® Corporation

```
    description
      "Parallel-port.";
    reference
      "RFC 1660 - Definitions of Managed Objects for
                  Parallel-printer-like Hardware Devices using
                  SMIv2";
  }
  identity arcnet {
    base iana-interface-type;
    description
      "ARCnet.";
  }
  identity arcnetPlus {
    base iana-interface-type;
    description
      "ARCnet Plus.";
  }



  identity atm {
    base iana-interface-type;
    description
      "ATM cells.";
  }
  identity miox25 {
    base iana-interface-type;
    reference
      "RFC 1461 - SNMP MIB extension for Multiprotocol
                  Interconnect over X.25";
  }
  identity sonet {
    base iana-interface-type;
    description
      "SONET or SDH.";
  }
  identity x25ple {
    base iana-interface-type;
    reference
      "RFC 2127 - ISDN Management Information Base using SMIv2";
  }
  identity iso88022llc {
    base iana-interface-type;
  }
  identity localTalk {
    base iana-interface-type;
  }
  identity smdsDxi {
    base iana-interface-type;
  }
  identity frameRelayService {
    base iana-interface-type;
    description
      "FRNETSERV-MIB.";
    reference
      "RFC 2954 - Definitions of Managed Objects for Frame
                  Relay Service";
  }
  identity v35 {
    base iana-interface-type;
```


Copyright© 2022 Ciena® Corporation

```
}
identity hssi {
  base iana-interface-type;
}
identity hippi {
  base iana-interface-type;
}

identity modem {
  base iana-interface-type;
  description
    "Generic modem.";
}
identity aal5 {
  base iana-interface-type;
  description
    "AAL5 over ATM.";
}
identity sonetPath {
  base iana-interface-type;
}
identity sonetVT {
  base iana-interface-type;
}
identity smdsIcip {
  base iana-interface-type;
  description
    "SMDS InterCarrier Interface.";
}
identity propVirtual {
  base iana-interface-type;
  description
    "Proprietary virtual/internal.";
  reference
    "RFC 2863 - The Interfaces Group MIB";
}
identity propMultiplexor {
  base iana-interface-type;
  description
    "Proprietary multiplexing.";
  reference
    "RFC 2863 - The Interfaces Group MIB";
}
identity ieee80212 {
  base iana-interface-type;
  description
    "100BaseVG.";
}
identity fibreChannel {
  base iana-interface-type;
  description
    "Fibre Channel.";
}



identity hippiInterface {
  base iana-interface-type;
  description
    "HIPPI interfaces.";
```


Copyright© 2022 Ciena® Corporation

```
  }
  identity frameRelayInterconnect {
    base iana-interface-type;
    status obsolete;
    description
      "Obsolete; use either
       frameRelay(32) or frameRelayService(44).";
  }
  identity aflane8023 {
    base iana-interface-type;
    description
      "ATM Emulated LAN for 802.3.";
  }
  identity aflane8025 {
    base iana-interface-type;
    description
      "ATM Emulated LAN for 802.5.";
  }
  identity cctEmul {
    base iana-interface-type;
    description
      "ATM Emulated circuit.";
  }
  identity fastEther {
    base iana-interface-type;
    status deprecated;
    description
      "Obsoleted via RFC 3635.
       ethernetCsmacd(6) should be used instead.";
    reference
      "RFC 3635 - Definitions of Managed Objects for the
                  Ethernet-like Interface Types";
  }
  identity isdn {
    base iana-interface-type;
    description
      "ISDN and X.25.";
    reference
      "RFC 1356 - Multiprotocol Interconnect on X.25 and ISDN
                  in the Packet Mode";
  }



  identity v11 {
    base iana-interface-type;
    description
      "CCITT V.11/X.21.";
  }
  identity v36 {
    base iana-interface-type;
    description
      "CCITT V.36.";
  }
  identity g703at64k {
    base iana-interface-type;
    description
      "CCITT G703 at 64Kbps.";
  }
  identity g703at2mb {
```


Copyright© 2022 Ciena® Corporation

```
    base iana-interface-type;
    status obsolete;
    description
      "Obsolete; see DS1-MIB.";
  }
  identity qllc {
    base iana-interface-type;
    description
      "SNA QLLC.";
  }
  identity fastEtherFX {
    base iana-interface-type;
    status deprecated;
    description
      "Obsoleted via RFC 3635.
       ethernetCsmacd(6) should be used instead.";
    reference
      "RFC 3635 - Definitions of Managed Objects for the
                  Ethernet-like Interface Types";
  }
  identity channel {
    base iana-interface-type;
    description
      "Channel.";
  }
  identity ieee80211 {
    base iana-interface-type;
    description
      "Radio spread spectrum.";
  }
  identity ibm370parChan {
    base iana-interface-type;
    description
      "IBM System 360/370 OEMI Channel.";
  }
  identity escon {
    base iana-interface-type;
    description
      "IBM Enterprise Systems Connection.";
  }
  identity dlsw {
    base iana-interface-type;
    description
      "Data Link Switching.";
  }
  identity isdns {
    base iana-interface-type;
    description
      "ISDN S/T interface.";
  }
  identity isdnu {
    base iana-interface-type;
    description
      "ISDN U interface.";
  }
  identity lapd {
    base iana-interface-type;
    description
      "Link Access Protocol D.";
  }
```


Copyright© 2022 Ciena® Corporation

```
  identity ipSwitch {
    base iana-interface-type;
    description
      "IP Switching Objects.";
  }
  identity rsrb {
    base iana-interface-type;
    description
      "Remote Source Route Bridging.";
  }
  identity atmLogical {
    base iana-interface-type;
    description
      "ATM Logical Port.";
    reference
      "RFC 3606 - Definitions of Supplemental Managed Objects
                  for ATM Interface";
  }
  identity ds0 {
    base iana-interface-type;
    description
      "Digital Signal Level 0.";
    reference
      "RFC 2494 - Definitions of Managed Objects for the DS0
                  and DS0 Bundle Interface Type";
  }
  identity ds0Bundle {
    base iana-interface-type;
    description
      "Group of ds0s on the same ds1.";
    reference
      "RFC 2494 - Definitions of Managed Objects for the DS0
                  and DS0 Bundle Interface Type";
  }
  identity bsc {
    base iana-interface-type;
    description
      "Bisynchronous Protocol.";
  }
  identity async {
    base iana-interface-type;
    description
      "Asynchronous Protocol.";
  }
  identity cnr {
    base iana-interface-type;
    description
      "Combat Net Radio.";
  }
  identity iso88025Dtr {
    base iana-interface-type;
    description
      "ISO 802.5r DTR.";
  }
  identity eplrs {
    base iana-interface-type;
    description
      "Ext Pos Loc Report Sys.";
  }
  identity arap {
```


Copyright© 2022 Ciena® Corporation

```
    base iana-interface-type;
    description
      "Appletalk Remote Access Protocol.";
  }
  identity propCnls {
    base iana-interface-type;
    description
      "Proprietary Connectionless Protocol.";
  }
  identity hostPad {
    base iana-interface-type;
    description
      "CCITT-ITU X.29 PAD Protocol.";
  }
  identity termPad {
    base iana-interface-type;
    description
      "CCITT-ITU X.3 PAD Facility.";
  }
  identity frameRelayMPI {
    base iana-interface-type;
    description
      "Multiproto Interconnect over FR.";
  }
  identity x213 {
    base iana-interface-type;
    description
      "CCITT-ITU X213.";
  }
  identity adsl {
    base iana-interface-type;
    description
      "Asymmetric Digital Subscriber Loop.";
  }
  identity radsl {
    base iana-interface-type;
    description
      "Rate-Adapt. Digital Subscriber Loop.";
  }
  identity sdsl {
    base iana-interface-type;
    description
      "Symmetric Digital Subscriber Loop.";
  }
  identity vdsl {
    base iana-interface-type;
    description
      "Very H-Speed Digital Subscrib. Loop.";
  }
  identity iso88025CRFPInt {
    base iana-interface-type;
    description
      "ISO 802.5 CRFP.";
  }
  identity myrinet {
    base iana-interface-type;
    description
      "Myricom Myrinet.";
  }
  identity voiceEM {
```


Copyright© 2022 Ciena® Corporation

```
    base iana-interface-type;
    description
      "Voice recEive and transMit.";
  }
  identity voiceFXO {
    base iana-interface-type;
    description
      "Voice Foreign Exchange Office.";
  }
  identity voiceFXS {
    base iana-interface-type;
    description
      "Voice Foreign Exchange Station.";
  }
  identity voiceEncap {
    base iana-interface-type;
    description
      "Voice encapsulation.";
  }
  identity voiceOverIp {
    base iana-interface-type;
    description
      "Voice over IP encapsulation.";
  }
  identity atmDxi {
    base iana-interface-type;
    description
      "ATM DXI.";
  }
  identity atmFuni {
    base iana-interface-type;
    description
      "ATM FUNI.";
  }
  identity atmIma {
    base iana-interface-type;
    description
      "ATM IMA.";
  }
  identity pppMultilinkBundle {
    base iana-interface-type;
    description
      "PPP Multilink Bundle.";
  }
  identity ipOverCdlc {
    base iana-interface-type;
    description
      "IBM ipOverCdlc.";
  }
  identity ipOverClaw {
    base iana-interface-type;
    description
      "IBM Common Link Access to Workstn.";
  }
  identity stackToStack {
    base iana-interface-type;
    description
      "IBM stackToStack.";
  }
  identity virtualIpAddress {
```


Copyright© 2022 Ciena® Corporation

```
    base iana-interface-type;
    description
      "IBM VIPA.";
  }
  identity mpc {
    base iana-interface-type;
    description
      "IBM multi-protocol channel support.";
  }
  identity ipOverAtm {
    base iana-interface-type;
    description
      "IBM ipOverAtm.";
    reference
      "RFC 2320 - Definitions of Managed Objects for Classical IP
                  and ARP Over ATM Using SMIv2 (IPOA-MIB)";
  }
  identity iso88025Fiber {
    base iana-interface-type;
    description
      "ISO 802.5j Fiber Token Ring.";
  }
  identity tdlc {
    base iana-interface-type;
    description
      "IBM twinaxial data link control.";
  }
  identity gigabitEthernet {
    base iana-interface-type;
    status deprecated;


    description
      "Obsoleted via RFC 3635.
       ethernetCsmacd(6) should be used instead.";
    reference
      "RFC 3635 - Definitions of Managed Objects for the
                  Ethernet-like Interface Types";
  }
  identity hdlc {
    base iana-interface-type;
    description
      "HDLC.";
  }
  identity lapf {
    base iana-interface-type;
    description
      "LAP F.";
  }
  identity v37 {
    base iana-interface-type;
    description
      "V.37.";
  }
  identity x25mlp {
    base iana-interface-type;
    description
      "Multi-Link Protocol.";
  }
  identity x25huntGroup {
```


Copyright© 2022 Ciena® Corporation

```
    base iana-interface-type;
    description
      "X25 Hunt Group.";
  }
  identity transpHdlc {
    base iana-interface-type;
    description
      "Transp HDLC.";
  }
  identity interleave {
    base iana-interface-type;
    description
      "Interleave channel.";
  }
  identity fast {
    base iana-interface-type;
    description
      "Fast channel.";
  }

  identity ip {
    base iana-interface-type;
    description
      "IP (for APPN HPR in IP networks).";
  }
  identity docsCableMaclayer {
    base iana-interface-type;
    description
      "CATV Mac Layer.";
  }
  identity docsCableDownstream {
    base iana-interface-type;
    description
      "CATV Downstream interface.";
  }
  identity docsCableUpstream {
    base iana-interface-type;
    description
      "CATV Upstream interface.";
  }
  identity a12MppSwitch {
    base iana-interface-type;
    description
      "Avalon Parallel Processor.";
  }
  identity tunnel {
    base iana-interface-type;
    description
      "Encapsulation interface.";
  }
  identity coffee {
    base iana-interface-type;
    description
      "Coffee pot.";
    reference
      "RFC 2325 - Coffee MIB";
  }
  identity ces {
    base iana-interface-type;
    description
```


Copyright© 2022 Ciena® Corporation

```
      "Circuit Emulation Service.";
  }
  identity atmSubInterface {
    base iana-interface-type;
    description
      "ATM Sub Interface.";
  }

  identity l2vlan {
    base iana-interface-type;
    description
      "Layer 2 Virtual LAN using 802.1Q.";
  }
  identity l3ipvlan {
    base iana-interface-type;
    description
      "Layer 3 Virtual LAN using IP.";
  }
  identity l3ipxvlan {
    base iana-interface-type;
    description
      "Layer 3 Virtual LAN using IPX.";
  }
  identity digitalPowerline {
    base iana-interface-type;
    description
      "IP over Power Lines.";
  }
  identity mediaMailOverIp {
    base iana-interface-type;
    description
      "Multimedia Mail over IP.";
  }
  identity dtm {
    base iana-interface-type;
    description
      "Dynamic synchronous Transfer Mode.";
  }
  identity dcn {
    base iana-interface-type;
    description
      "Data Communications Network.";
  }
  identity ipForward {
    base iana-interface-type;
    description
      "IP Forwarding Interface.";
  }
  identity msdsl {
    base iana-interface-type;
    description
      "Multi-rate Symmetric DSL.";
  }
  identity ieee1394 {
    base iana-interface-type;

    description
      "IEEE1394 High Performance Serial Bus.";
  }
  identity if-gsn {
```


Copyright© 2022 Ciena® Corporation

```
    base iana-interface-type;
    description
      "HIPPI-6400.";
  }
  identity dvbRccMacLayer {
    base iana-interface-type;
    description
      "DVB-RCC MAC Layer.";
  }
  identity dvbRccDownstream {
    base iana-interface-type;
    description
      "DVB-RCC Downstream Channel.";
  }
  identity dvbRccUpstream {
    base iana-interface-type;
    description
      "DVB-RCC Upstream Channel.";
  }
  identity atmVirtual {
    base iana-interface-type;
    description
      "ATM Virtual Interface.";
  }
  identity mplsTunnel {
    base iana-interface-type;
    description
      "MPLS Tunnel Virtual Interface.";
  }
  identity srp {
    base iana-interface-type;
    description
      "Spatial Reuse Protocol.";
  }
  identity voiceOverAtm {
    base iana-interface-type;
    description
      "Voice over ATM.";
  }
  identity voiceOverFrameRelay {
    base iana-interface-type;
    description
      "Voice Over Frame Relay.";
  }
  identity idsl {
    base iana-interface-type;
    description
      "Digital Subscriber Loop over ISDN.";
  }
  identity compositeLink {
    base iana-interface-type;
    description
      "Avici Composite Link Interface.";
  }
  identity ss7SigLink {
    base iana-interface-type;
    description
      "SS7 Signaling Link.";
  }
  identity propWirelessP2P {
```


Copyright© 2022 Ciena® Corporation

```
    base iana-interface-type;
    description
      "Prop. P2P wireless interface.";
  }
  identity frForward {
    base iana-interface-type;
    description
      "Frame Forward Interface.";
  }
  identity rfc1483 {
    base iana-interface-type;
    description
      "Multiprotocol over ATM AAL5.";
    reference
      "RFC 1483 - Multiprotocol Encapsulation over ATM
                  Adaptation Layer 5";
  }
  identity usb {
    base iana-interface-type;
    description
      "USB Interface.";
  }
  identity ieee8023adLag {
    base iana-interface-type;
    description
      "IEEE 802.3ad Link Aggregate.";
  }
  identity bgppolicyaccounting {
    base iana-interface-type;
    description
      "BGP Policy Accounting.";
  }
  identity frf16MfrBundle {
    base iana-interface-type;
    description
      "FRF.16 Multilink Frame Relay.";
  }
  identity h323Gatekeeper {
    base iana-interface-type;
    description
      "H323 Gatekeeper.";
  }
  identity h323Proxy {
    base iana-interface-type;
    description
      "H323 Voice and Video Proxy.";
  }
  identity mpls {
    base iana-interface-type;
    description
      "MPLS.";
  }
  identity mfSigLink {
    base iana-interface-type;
    description
      "Multi-frequency signaling link.";
  }
  identity hdsl2 {
    base iana-interface-type;
    description
```


Copyright© 2022 Ciena® Corporation

```
      "High Bit-Rate DSL - 2nd generation.";
  }
  identity shdsl {
    base iana-interface-type;
    description
      "Multirate HDSL2.";
  }
  identity ds1FDL {
    base iana-interface-type;
    description
      "Facility Data Link (4Kbps) on a DS1.";
  }
  identity pos {
    base iana-interface-type;
    description
      "Packet over SONET/SDH Interface.";
  }



  identity dvbAsiIn {
    base iana-interface-type;
    description
      "DVB-ASI Input.";
  }
  identity dvbAsiOut {
    base iana-interface-type;
    description
      "DVB-ASI Output.";
  }
  identity plc {
    base iana-interface-type;
    description
      "Power Line Communications.";
  }
  identity nfas {
    base iana-interface-type;
    description
      "Non-Facility Associated Signaling.";
  }
  identity tr008 {
    base iana-interface-type;
    description
      "TR008.";
  }
  identity gr303RDT {
    base iana-interface-type;
    description
      "Remote Digital Terminal.";
  }
  identity gr303IDT {
    base iana-interface-type;
    description
      "Integrated Digital Terminal.";
  }
  identity isup {
    base iana-interface-type;
    description
      "ISUP.";
  }
```


Copyright© 2022 Ciena® Corporation

```
identity propDocsWirelessMaclayer {
  base iana-interface-type;
  description
    "Cisco proprietary Maclayer.";
}



identity propDocsWirelessDownstream {
  base iana-interface-type;
  description
    "Cisco proprietary Downstream.";
}
identity propDocsWirelessUpstream {
  base iana-interface-type;
  description
    "Cisco proprietary Upstream.";
}
identity hiperlan2 {
  base iana-interface-type;
  description
    "HIPERLAN Type 2 Radio Interface.";
}
identity propBWAp2Mp {
  base iana-interface-type;
  description
    "PropBroadbandWirelessAccesspt2Multipt (use of this value
     for IEEE 802.16 WMAN interfaces as per IEEE Std 802.16f
     is deprecated, and ieee80216WMAN(237) should be used
     instead).";
}
identity sonetOverheadChannel {
  base iana-interface-type;
  description
    "SONET Overhead Channel.";
}
identity digitalWrapperOverheadChannel {
  base iana-interface-type;
  description
    "Digital Wrapper.";
}
identity aal2 {
  base iana-interface-type;
  description
    "ATM adaptation layer 2.";
}
identity radioMAC {
  base iana-interface-type;
  description
    "MAC layer over radio links.";
}
identity atmRadio {
  base iana-interface-type;
  description
    "ATM over radio links.";
}
identity imt {
  base iana-interface-type;
  description
    "Inter-Machine Trunks.";
```


Copyright© 2022 Ciena® Corporation

```
  }
  identity mvl {
    base iana-interface-type;
    description
      "Multiple Virtual Lines DSL.";
  }
  identity reachDSL {
    base iana-interface-type;
    description
      "Long Reach DSL.";
  }
  identity frDlciEndPt {
    base iana-interface-type;
    description
      "Frame Relay DLCI End Point.";
  }
  identity atmVciEndPt {
    base iana-interface-type;
    description
      "ATM VCI End Point.";
  }
  identity opticalChannel {
    base iana-interface-type;
    description
      "Optical Channel.";
  }
  identity opticalTransport {
    base iana-interface-type;
    description
      "Optical Transport.";
  }
  identity propAtm {
    base iana-interface-type;
    description
      "Proprietary ATM.";
  }
  identity voiceOverCable {
    base iana-interface-type;
    description
      "Voice Over Cable Interface.";
  }



  identity infiniband {
    base iana-interface-type;
    description
      "Infiniband.";
  }
  identity teLink {
    base iana-interface-type;
    description
      "TE Link.";
  }
  identity q2931 {
    base iana-interface-type;
    description
      "Q.2931.";
  }
  identity virtualTg {
```


Copyright© 2022 Ciena® Corporation

```
    base iana-interface-type;
    description
      "Virtual Trunk Group.";
  }
  identity sipTg {
    base iana-interface-type;
    description
      "SIP Trunk Group.";
  }
  identity sipSig {
    base iana-interface-type;
    description
      "SIP Signaling.";
  }
  identity docsCableUpstreamChannel {
    base iana-interface-type;
    description
      "CATV Upstream Channel.";
  }
  identity econet {
    base iana-interface-type;
    description
      "Acorn Econet.";
  }
  identity pon155 {
    base iana-interface-type;
    description
      "FSAN 155Mb Symetrical PON interface.";
  }



  identity pon622 {
    base iana-interface-type;
    description
      "FSAN 622Mb Symetrical PON interface.";
  }
  identity bridge {
    base iana-interface-type;
    description
      "Transparent bridge interface.";
  }
  identity linegroup {
    base iana-interface-type;
    description
      "Interface common to multiple lines.";
  }
  identity voiceEMFGD {
    base iana-interface-type;
    description
      "Voice E&M Feature Group D.";
  }
  identity voiceFGDEANA {
    base iana-interface-type;
    description
      "Voice FGD Exchange Access North American.";
  }
  identity voiceDID {
    base iana-interface-type;
    description
```


Copyright© 2022 Ciena® Corporation

```
      "Voice Direct Inward Dialing.";
  }
  identity mpegTransport {
    base iana-interface-type;
    description
      "MPEG transport interface.";
  }
  identity sixToFour {
    base iana-interface-type;
    status deprecated;
    description
      "6to4 interface (DEPRECATED).";
    reference
      "RFC 4087 - IP Tunnel MIB";
  }
  identity gtp {
    base iana-interface-type;
    description
      "GTP (GPRS Tunneling Protocol).";
  }
  identity pdnEtherLoop1 {
    base iana-interface-type;
    description
      "Paradyne EtherLoop 1.";
  }
  identity pdnEtherLoop2 {
    base iana-interface-type;
    description
      "Paradyne EtherLoop 2.";
  }
  identity opticalChannelGroup {
    base iana-interface-type;
    description
      "Optical Channel Group.";
  }
  identity homepna {
    base iana-interface-type;
    description
      "HomePNA ITU-T G.989.";
  }
  identity gfp {
    base iana-interface-type;
    description
      "Generic Framing Procedure (GFP).";
  }
  identity ciscoISLvlan {
    base iana-interface-type;
    description
      "Layer 2 Virtual LAN using Cisco ISL.";
  }
  identity actelisMetaLOOP {
    base iana-interface-type;
    description
      "Acteleis proprietary MetaLOOP High Speed Link.";
  }
  identity fcipLink {
    base iana-interface-type;
    description
      "FCIP Link.";
  }
```


Copyright© 2022 Ciena® Corporation

```
  identity rpr {
    base iana-interface-type;
    description
      "Resilient Packet Ring Interface Type.";
  }



  identity qam {
    base iana-interface-type;
    description
      "RF Qam Interface.";
  }
  identity lmp {
    base iana-interface-type;
    description
      "Link Management Protocol.";
    reference
      "RFC 4327 - Link Management Protocol (LMP) Management
                  Information Base (MIB)";
  }
  identity cblVectaStar {
    base iana-interface-type;
    description
      "Cambridge Broadband Networks Limited VectaStar.";
  }
  identity docsCableMCmtsDownstream {
    base iana-interface-type;
    description
      "CATV Modular CMTS Downstream Interface.";
  }
  identity adsl2 {
    base iana-interface-type;
    status deprecated;
    description
      "Asymmetric Digital Subscriber Loop Version 2
       (DEPRECATED/OBSOLETED - please use adsl2plus(238)
       instead).";
    reference
      "RFC 4706 - Definitions of Managed Objects for Asymmetric
                  Digital Subscriber Line 2 (ADSL2)";
  }
  identity macSecControlledIF {
    base iana-interface-type;
    description
      "MACSecControlled.";
  }
  identity macSecUncontrolledIF {
    base iana-interface-type;
    description
      "MACSecUncontrolled.";
  }
  identity aviciOpticalEther {
    base iana-interface-type;
    description
      "Avici Optical Ethernet Aggregate.";
  }
  identity atmbond {
    base iana-interface-type;
    description
```


Copyright© 2022 Ciena® Corporation

```
      "atmbond.";
  }
  identity voiceFGDOS {
    base iana-interface-type;
    description
      "Voice FGD Operator Services.";
  }
  identity mocaVersion1 {
    base iana-interface-type;
    description
      "MultiMedia over Coax Alliance (MoCA) Interface
       as documented in information provided privately to IANA.";
  }
  identity ieee80216WMAN {
    base iana-interface-type;
    description
      "IEEE 802.16 WMAN interface.";
  }
  identity adsl2plus {
    base iana-interface-type;
    description
      "Asymmetric Digital Subscriber Loop Version 2 -
       Version 2 Plus and all variants.";
  }
  identity dvbRcsMacLayer {
    base iana-interface-type;
    description
      "DVB-RCS MAC Layer.";
    reference
      "RFC 5728 - The SatLabs Group DVB-RCS MIB";
  }
  identity dvbTdm {
    base iana-interface-type;
    description
      "DVB Satellite TDM.";
    reference
      "RFC 5728 - The SatLabs Group DVB-RCS MIB";
  }
  identity dvbRcsTdma {
    base iana-interface-type;
    description
      "DVB-RCS TDMA.";
    reference
      "RFC 5728 - The SatLabs Group DVB-RCS MIB";
  }
  identity x86Laps {
    base iana-interface-type;
    description
      "LAPS based on ITU-T X.86/Y.1323.";
  }
  identity wwanPP {
    base iana-interface-type;
    description
      "3GPP WWAN.";
  }
  identity wwanPP2 {
    base iana-interface-type;
    description
      "3GPP2 WWAN.";
  }
```


Copyright© 2022 Ciena® Corporation

```
  identity voiceEBS {
    base iana-interface-type;
    description
      "Voice P-phone EBS physical interface.";
  }
  identity ifPwType {
    base iana-interface-type;
    description
      "Pseudowire interface type.";
    reference
      "RFC 5601 - Pseudowire (PW) Management Information Base (MIB)";
  }
  identity ilan {
    base iana-interface-type;
    description
      "Internal LAN on a bridge per IEEE 802.1ap.";
  }
  identity pip {
    base iana-interface-type;
    description
      "Provider Instance Port on a bridge per IEEE 802.1ah PBB.";
  }
  identity aluELP {
    base iana-interface-type;
    description
      "Alcatel-Lucent Ethernet Link Protection.";
  }
  identity gpon {
    base iana-interface-type;
    description
      "Gigabit-capable passive optical networks (G-PON) as per
       ITU-T G.948.";
  }
  identity vdsl2 {
    base iana-interface-type;
    description
      "Very high speed digital subscriber line Version 2
       (as per ITU-T Recommendation G.993.2).";
    reference
      "RFC 5650 - Definitions of Managed Objects for Very High
                  Speed Digital Subscriber Line 2 (VDSL2)";
  }
  identity capwapDot11Profile {
    base iana-interface-type;
    description
      "WLAN Profile Interface.";
    reference
      "RFC 5834 - Control and Provisioning of Wireless Access
                  Points (CAPWAP) Protocol Binding MIB for
                  IEEE 802.11";
  }
  identity capwapDot11Bss {
    base iana-interface-type;
    description
      "WLAN BSS Interface.";
    reference
      "RFC 5834 - Control and Provisioning of Wireless Access
                  Points (CAPWAP) Protocol Binding MIB for
                  IEEE 802.11";
  }
```


Copyright© 2022 Ciena® Corporation

```
  identity capwapWtpVirtualRadio {
    base iana-interface-type;
    description
      "WTP Virtual Radio Interface.";
    reference
      "RFC 5833 - Control and Provisioning of Wireless Access
                  Points (CAPWAP) Protocol Base MIB";
  }
  identity bits {
    base iana-interface-type;
    description
      "bitsport.";
  }
  identity docsCableUpstreamRfPort {
    base iana-interface-type;
    description
      "DOCSIS CATV Upstream RF Port.";
  }


  identity cableDownstreamRfPort {
    base iana-interface-type;
    description
      "CATV downstream RF Port.";
  }
  identity vmwareVirtualNic {
    base iana-interface-type;
    description
      "VMware Virtual Network Interface.";
  }
  identity ieee802154 {
    base iana-interface-type;
    description
      "IEEE 802.15.4 WPAN interface.";
    reference
      "IEEE 802.15.4-2006";
  }
  identity otnOdu {
    base iana-interface-type;
    description
      "OTN Optical Data Unit.";
  }
  identity otnOtu {
    base iana-interface-type;
    description
      "OTN Optical channel Transport Unit.";
  }
  identity ifVfiType {
    base iana-interface-type;
    description
      "VPLS Forwarding Instance Interface Type.";
  }
  identity g9981 {
    base iana-interface-type;
    description
      "G.998.1 bonded interface.";
  }
  identity g9982 {
    base iana-interface-type;
    description
```


Copyright© 2022 Ciena® Corporation

```
      "G.998.2 bonded interface.";
  }
  identity g9983 {
    base iana-interface-type;
    description
      "G.998.3 bonded interface.";
  }

  identity aluEpon {
    base iana-interface-type;
    description
      "Ethernet Passive Optical Networks (E-PON).";
  }
  identity aluEponOnu {
    base iana-interface-type;
    description
      "EPON Optical Network Unit.";
  }
  identity aluEponPhysicalUni {
    base iana-interface-type;
    description
      "EPON physical User to Network interface.";
  }
  identity aluEponLogicalLink {
    base iana-interface-type;
    description
      "The emulation of a point-to-point link over the EPON
       layer.";
  }
  identity aluGponOnu {
    base iana-interface-type;
    description
      "GPON Optical Network Unit.";
    reference
      "ITU-T G.984.2";
  }
  identity aluGponPhysicalUni {
    base iana-interface-type;
    description
      "GPON physical User to Network interface.";
    reference
      "ITU-T G.984.2";
  }
  identity vmwareNicTeam {
    base iana-interface-type;
    description
      "VMware NIC Team.";
  }
}
```


Copyright© 2022 Ciena® Corporation

ieee-dot1q-types.yang

```
module ieee-dot1q-types {

   namespace "urn:ieee:params:xml:ns:yang:ieee-dot1q-types";
    prefix "dot1q-types";

    import ietf-interfaces {
      prefix "if";
   }

    organization
        "Institute of Electrical and Electronics Engineers";

    contact
    "Web URL: http://www.ieee802.org/1/

        Working Group Chair:
       Glenn Parsons
       glenn.parsons@ericsson.com

        Working Group Vice-chair:
       John Messenger
       J.L.Messenger@ieee.org

        Chair, Time Sensitive Networking Task Group:
       Michael Johas Teener
       mikejt@broadcom.com

    Vice-Chair, Time Sensitive Networking Task Group:
       Janos Farkas
         janos.farkas@ericsson.com

        Editor:
       Marc Holness
        mholness@ciena.com";

    description
    "Common types used within dot1Q-bridge modules.";

    revision "2015-09-10" {
        description
        "Initial revision.";
        reference
            "IEEE 802.1Q-2014, Media Access Control (MAC) Bridges
        and Virtual Bridged Local Area Networks.";
    }

   /*
     * IEEE 802.1Q Type Definitions.
     */

    typedef component-ref {
    type string;
    description
          "Unique identifier for a particular Bridge component
      within the system.";
    }

   typedef name-type {
```

```
      type string {
          length "0..32";
      }
      description
          "A text string of up to 32 characters, of locally
          determined significance.";
   }

   typedef port-number-type {
      type uint16 {
          range "1..4095";
      }
      description
          "An Unsigned value assigned to a Port as part of
          a Port Identifier. Valid Port Numbers are in the range 1
          through 4095.";
   }

   typedef priority-type {
      type uint8 {
          range "0..7";
      }
      description
          "A range of priorities from 0 to 7 (inclusive).";
   }

   typedef vid-range {
      /*
       * Defines the type used to represent ranges of VLAN Ids.
       *
       * Ideally we would model that as a list of VLAN Ids in YANG,
       * but the model is easier to use if this is just represented
       * as a string.
       *
       * This type is used to match an ordered list of VLAN Ids, or
       * contiguous ranges of VLAN Ids. Valid VLAN Ids must be in
       * the range 1 to 4094, and included in the list in non
       * overlapping ascending order.
       *
       * For example, "1, 10-100, 50, 500-1000"
       *
       */
      type string {
          pattern "([0-9]{1,4}(-[0-9]{1,4})?(,[0-9]{1,4}" +
                  "(-[0-9]{1,4})?)*)";
      }
      description
          "A list of VLAN Ids, or non overlapping VLAN ranges,
            in ascending order, between 1 and 4094";
   }

   typedef pcp-selection-type {
      type enumeration {
          enum 8P0D {
             description
                "8 priorities, 0 discard";
          }
          enum 7P1D {
             description
                "7 priorities, 1 discard";
```


Copyright© 2022 Ciena® Corporation

```
            }
            enum 6P2D {
               description
                  "6 priorities, 2 discards";
            }
            enum 5P3D {
               description
                  "5 priorities, 3 discards";
            }
         }
         description
            "Priority Code Point selection types.";
         reference
            "IEEE 802.1Q-2014 Clause 12.6.2.5.3, 6.9.3";
      }

      typedef protocol-template-format {
         type enumeration {
            enum Ethernet {
               description
                  "Ethernet frame format";
            }
            enum rfc1042 {
               description
                  "RFC 1042 frame format";
            }
            enum snap8021H {
               description
                  "SNAP 802.1H frame format";
            }
            enum snapOther {
               description
                  "Other SNAP frame format";
            }
            enum llcOther {
               description
                  "Other LLC frame format";
            }
         }
         description
            "A value representing the frame format to be matched.";
         reference
            "IEEE 802.1Q-2014 Clause 12.10.1.7.1";
      }

      typedef etherType {
         type uint16;
         description
            "The Ethernet Type (or Length) value.";
         reference
            "IEEE 802.3-2012 Clause 3.2.6";
      }

      /*
        * IEEE 802.1Q Bridge Group definitions.
        */

      grouping priority-regeneration-table {
         description
            "The priority regeneration table.";
```


Copyright© 2022 Ciena® Corporation

```
        reference
            "IEEE 802.1Q-2014 Clause 6.9.4";
        leaf priority0 {
            type priority-type;
            description
               "Priority 0";
            reference
               "IEEE 802.1Q-2014 Clause 6.9.4, 12.6.2.3";
        }
        leaf priority1 {
            type priority-type;
            description
               "Priority 1";
            reference
               "IEEE 802.1Q-2014 Clause 6.9.4, 12.6.2.3";
        }
        leaf priority2 {
            type priority-type;
            description
               "Priority 2";
            reference
               "IEEE 802.1Q-2014 Clause 6.9.4, 12.6.2.3";
        }
        leaf priority3 {
            type priority-type;
            description
               "Priority 3";
            reference
               "IEEE 802.1Q-2014 Clause 6.9.4, 12.6.2.3";
        }
        leaf priority4 {
            type priority-type;
            description
               "Priority 4";
            reference
               "IEEE 802.1Q-2014 Clause 6.9.4, 12.6.2.3";
        }
        leaf priority5 {
            type priority-type;
            description
               "Priority 5";
            reference
               "IEEE 802.1Q-2014 Clause 6.9.4, 12.6.2.3";
        }
        leaf priority6 {
            type priority-type;
            description
               "Priority 6";
            reference
               "IEEE 802.1Q-2014 Clause 6.9.4, 12.6.2.3";
        }
        leaf priority7 {
            type priority-type;
            description
               "Priority 7";
            reference
               "IEEE 802.1Q-2014 Clause 6.9.4, 12.6.2.3";
        }
    } // priority-regeneration-table
```


Copyright© 2022 Ciena® Corporation

```
grouping pcpDecodingTable {
   description
       "The Priority Code Point decoding table.";
   reference
       "IEEE 802.1Q-2014 Clause 6.9.3";
   container pcp0 {
       description
          "Priority code point 0 value";
       leaf priority {
          type priority-type;
          description
             "Priority associated with pcp";
          reference
             "IEEE 802.1Q-2014 Clause 12.6.2.7, 6.9.3";
       }
       leaf drop-eligible {
          type boolean;
          description
             "Drop eligible value for pcp0";
          reference
             "IEEE 802.1Q-2014 Clause 12.6.2.7, 6.9.3";
       }
   }
   container pcp1 {
       description
          "Priority code point 1 value";
       leaf priority {
          type priority-type;
          description
             "Priority associated with pcp";
          reference
             "IEEE 802.1Q-2014 Clause 12.6.2.7, 6.9.3";
       }
       leaf drop-eligible {
          type boolean;
          description
             "Drop eligible value for pcp1";
          reference
             "IEEE 802.1Q-2014 Clause 12.6.2.7, 6.9.3";
       }
   }
   container pcp2 {
       description
          "Priority code point 2 value";
       leaf priority {
          type priority-type;
          description
             "Priority associated with pcp";
          reference
             "IEEE 802.1Q-2014 Clause 12.6.2.7, 6.9.3";
       }
       leaf drop-eligible {
          type boolean;
          description
             "Drop eligible value for pcp2";
          reference
             "IEEE 802.1Q-2014 Clause 12.6.2.7, 6.9.3";
       }
   }
   container pcp3 {
```


Copyright© 2022 Ciena® Corporation

```
        description
           "Priority code point 3 value";
        leaf priority {
           type priority-type;
           description
              "Priority associated with pcp";
           reference
              "IEEE 802.1Q-2014 Clause 12.6.2.7, 6.9.3";
        }
        leaf drop-eligible {
           type boolean;
           description
              "Drop eligible value for pcp3";
           reference
              "IEEE 802.1Q-2014 Clause 12.6.2.7, 6.9.3";
        }
     }
     container pcp4 {
        description
           "Priority code point 4 value";
        leaf priority {
           type priority-type;
           description
              "Priority associated with pcp";
           reference
              "IEEE 802.1Q-2014 Clause 12.6.2.7, 6.9.3";
        }
        leaf drop-eligible {
           type boolean;
           description
              "Drop eligible value for pcp4";
           reference
              "IEEE 802.1Q-2014 Clause 12.6.2.7, 6.9.3";
        }
     }
     container pcp5 {
        description
           "Priority code point 5 value";
        leaf priority {
           type priority-type;
           description
              "Priority associated with pcp";
           reference
              "IEEE 802.1Q-2014 Clause 12.6.2.7, 6.9.3";
        }
        leaf drop-eligible {
           type boolean;
           description
              "Drop eligible value for pcp5";
           reference
              "IEEE 802.1Q-2014 Clause 12.6.2.7, 6.9.3";
        }
     }
     container pcp6 {
        description
           "Priority code point 6 value";
        leaf priority {
           type priority-type;
           description
              "Priority associated with pcp";
```


Copyright© 2022 Ciena® Corporation

```
            reference
               "IEEE 802.1Q-2014 Clause 12.6.2.7, 6.9.3";
         }
         leaf drop-eligible {
            type boolean;
            description
               "Drop eligible value for pcp6";
            reference
               "IEEE 802.1Q-2014 Clause 12.6.2.7, 6.9.3";
         }
      }
      container pcp7 {
         description
            "Priority code point 7 value";
         leaf priority {
            type priority-type;
            description
               "Priority associated with pcp";
            reference
               "IEEE 802.1Q-2014 Clause 12.6.2.7, 6.9.3";
         }
         leaf drop-eligible {
            type boolean;
            description
               "Drop eligible value for pcp7";
            reference
               "IEEE 802.1Q-2014 Clause 12.6.2.7, 6.9.3";
         }
      }
   } // pcpDecodingTable

   grouping pcp-encoding-table {
      description
          "The Priority Code Point encoding table.";
      reference
          "IEEE 802.1Q-2014 Clause 12.6.2.9, 6.9.3";
      container priority0 {
          description
             "Priority 0 PCP encoding table entry";
          leaf pcp-dei-false {
             type priority-type;
             description
                "PCP value for priority 0 when DEI false";
             reference
                "IEEE 802.1Q-2014 Clause 12.6.2.9, 6.9.3";
          }
          leaf pcp-dei-true {
             type priority-type;
             description
                "PCP value for priority 0 when DEI true";
             reference
                "IEEE 802.1Q-2014 Clause 12.6.2.9, 6.9.3";
          }
      }
      container priority1 {
          description
             "Priority 1 PCP encoding table entry";
          leaf pcp-dei-false {
             type priority-type;
             description
```


Copyright© 2022 Ciena® Corporation

```
                "PCP value for priority 1 when DEI false";
            reference
                "IEEE 802.1Q-2014 Clause 12.6.2.9, 6.9.3";
        }
        leaf pcp-dei-true {
            type priority-type;
            description
                "PCP value for priority 1 when DEI true";
            reference
                "IEEE 802.1Q-2014 Clause 12.6.2.9, 6.9.3";
        }
    }
    container priority2 {
        description
            "Priority 2 PCP encoding table entry";
        leaf pcp-dei-false {
            type priority-type;
            description
                "PCP value for priority 2 when DEI false";
            reference
                "IEEE 802.1Q-2014 Clause 12.6.2.9, 6.9.3";
        }
        leaf pcp-dei-true {
            type priority-type;
            description
                "PCP value for priority 2 when DEI true";
            reference
                "IEEE 802.1Q-2014 Clause 12.6.2.9, 6.9.3";
        }
    }
    container priority3 {
        description
            "Priority 3 PCP encoding table entry";
        leaf pcp-dei-false {
            type priority-type;
            description
                "PCP value for priority 3 when DEI false";
            reference
                "IEEE 802.1Q-2014 Clause 12.6.2.9, 6.9.3";
        }
        leaf pcp-dei-true {
            type priority-type;
            description
                "PCP value for priority 3 when DEI true";
            reference
                "IEEE 802.1Q-2014 Clause 12.6.2.9, 6.9.3";
        }
    }
    container priority4 {
        description
            "Priority 4 PCP encoding table entry";
        leaf pcp-dei-false {
            type priority-type;
            description
                "PCP value for priority 4 when DEI false";
            reference
                "IEEE 802.1Q-2014 Clause 12.6.2.9, 6.9.3";
        }
        leaf pcp-dei-true {
            type priority-type;
```


Copyright© 2022 Ciena® Corporation

```
            description
               "PCP value for priority 4 when DEI true";
            reference
               "IEEE 802.1Q-2014 Clause 12.6.2.9, 6.9.3";
        }
    }
    container priority5 {
        description
           "Priority 5 PCP encoding table entry";
        leaf pcp-dei-false {
           type priority-type;
           description
              "PCP value for priority 5 when DEI false";
           reference
              "IEEE 802.1Q-2014 Clause 12.6.2.9, 6.9.3";
        }
        leaf pcp-dei-true {
           type priority-type;
           description
              "PCP value for priority 5 when DEI true";
           reference
              "IEEE 802.1Q-2014 Clause 12.6.2.9, 6.9.3";
        }
    }
    container priority6 {
        description
           "Priority 6 PCP encoding table entry";
        leaf pcp-dei-false {
           type priority-type;
           description
              "PCP value for priority 6 when DEI false";
           reference
              "IEEE 802.1Q-2014 Clause 12.6.2.9, 6.9.3";
        }
        leaf pcp-dei-true {
           type priority-type;
           description
              "PCP value for priority 6 when DEI true";
           reference
              "IEEE 802.1Q-2014 Clause 12.6.2.9, 6.9.3";
        }
    }
    container priority7 {
        description
           "Priority 7 PCP encoding table entry";
        leaf pcp-dei-false {
           type priority-type;
           description
              "PCP value for priority 7 when DEI false";
           reference
              "IEEE 802.1Q-2014 Clause 12.6.2.9, 6.9.3";
        }
        leaf pcp-dei-true {
           type priority-type;
           description
              "PCP value for priority 7 when DEI true";
           reference
              "IEEE 802.1Q-2014 Clause 12.6.2.9, 6.9.3";
        }
    }
```


Copyright© 2022 Ciena® Corporation

```
} // pcp-encoding-table

grouping service-access-priority-table {
   description
       "The Service Access Priority Table.";
   reference
       "IEEE 802.1Q-2014 Clause 6.13.1, 12.6.2.17";
   leaf priority0 {
       type priority-type;
       description
          "Service access priority value for priority 0";
       reference
          "IEEE 802.1Q-2014 Clause 6.13.1, 12.6.2.17";
   }
   leaf priority1 {
       type priority-type;
       description
          "Service access priority value for priority 1";
       reference
          "IEEE 802.1Q-2014 Clause 6.13.1, 12.6.2.17";
   }
   leaf priority2 {
       type priority-type;
       description
          "Service access priority value for priority 2";
       reference
          "IEEE 802.1Q-2014 Clause 6.13.1, 12.6.2.17";
   }
   leaf priority3 {
       type priority-type;
       description
          "Service access priority value for priority 3";
       reference
          "IEEE 802.1Q-2014 Clause 6.13.1, 12.6.2.17";
   }
   leaf priority4 {
       type priority-type;
       description
          "Service access priority value for priority 4";
       reference
          "IEEE 802.1Q-2014 Clause 6.13.1, 12.6.2.17";
   }
   leaf priority5 {
       type priority-type;
       description
          "Service access priority value for priority 5";
       reference
          "IEEE 802.1Q-2014 Clause 6.13.1, 12.6.2.17";
   }
   leaf priority6 {
       type priority-type;
       description
          "Service access priority value for priority 6";
       reference
          "IEEE 802.1Q-2014 Clause 6.13.1, 12.6.2.17";
   }
   leaf priority7 {
       type priority-type;
       description
          "Service access priority value for priority 7";
```


Copyright© 2022 Ciena® Corporation

```
        reference
           "IEEE 802.1Q-2014 Clause 6.13.1, 12.6.2.17";
    }
  } // service-access-priority-table

  grouping traffic-class-table {
    description
        "The Traffic Class Table models the operations that can
        be performed on, or inquire about, the current contents
        of the Traffic Class Table (8.6.6) for a given Port.";
    reference
        "IEEE 802.1Q-2014 Clause 12.6.3, 8.6.6";
    // To be confirmed
    list tc-entries {
        key "priority";
        description
           "The priority index into the traffic class table.";
        leaf priority {
           type priority-type;
           description
              "The priority of the traffic class entry.";
           reference
              "IEEE 802.1Q-2014 Clause 8.6.6";
        }
        list available-TC {
           key "traffic-class";
           description
              "The traffic class index associated with a given
              priority within the traffic class table.";
           reference
              "IEEE 802.1Q-2014 Clause 8.6.6";
           leaf traffic-class {
              type uint8 {
                    range "1..8";
              }
              description
                    "The traffic class index associated with a
                    given traffic class entry.";
              reference
                    "IEEE 802.1Q-2014 Clause 8.6.6";
           }
           leaf num-tc {
              type priority-type;
              description
                    "The available number of traffic classes.";
              reference
                    "IEEE 802.1Q-2014 Clause 8.6.6";
           }
        }
    }
  } // traffic-class-table

  grouping port-map {
    description
        "A set of control indicators, one for each Port.";
    reference
        "IEEE 802.1Q-2014 Clause 8.8.1, 8.8.2";
    list port-map {
        key "port-number";
        description
```


Copyright© 2022 Ciena® Corporation

```
            "The list of entries composing the port map.";
         leaf port-number {
            //type port-number-type;
            type if:interface-ref;
            description
               "The interface port reference associated with
               this map.";
            reference
               "IEEE 802.1Q-2014 Clause 8.8.1";
         }
         leaf map {
            // Need to confirm this
            type enumeration {
               enum forward {
                     description
                         "Forwarded, independently of any dynamic
                         filtering information held by the FDB.";
               }
               enum filter {
                     description
                         "Filtered, independently of any dynamic
                         filtering information.";
               }
               enum forward-filter {
                     description
                         "Forwarded or filtered on the basis of
                         dynamic filtering information, or on the
                         basis of the default Group filtering
                         behavior for the outbound Port (8.8.6)
                         if no dynamic filtering information is
                         present specifically for the MAC address.";
               }
            }
            description
               "containing a control element for each outbound
               Port, specifying that a frame with a destination
               MAC address, and in the case of VLAN Bridge
               components, VID that meets this specification.";
            reference
               "IEEE 802.1Q-2014 Clause 8.8.1";
         }
      }
   } // port-map
}
```


Copyright© 2022 Ciena® Corporation

ieee-port-lldp.yang

```
module ieee-port-lldp {
    namespace "ieee:ns:yang:ieee-port-lldp";
    prefix "port-lldp";

    import ietf-interfaces {
        prefix "if";
    }

    import ietf-yang-types {
        prefix "yang";
    }

    organization
        "Ciena Corporation";

    contact
        "Web URL: http://www.ciena.com/
        E-mail:  yang@ciena.com
        Postal:  7035 Ridge Road
                 Hanover, Maryland 21076
                 U.S.A.
        Phone:   +1 800-921-1144
        Fax:     +1 410-694-5750";

    description
      "This YANG module contains the schema definition for the port portion
        of Link Layer Discovery Protocol (LLDP).";

    revision "2014-12-30" {
      description
        "Initial Version.";
      reference
        "Add the reference here.";
    }

    augment "/if:interfaces/if:interface" {
        when "if:type = 'ianaift:ethernetCsmacd'";

        leaf admin-status {
            type enumeration {
                enum tx-only {
                    description
                       "Enables or disables the transmission of LLDP packets
                         on an interface.";
                }
                enum rx-only {
                    description
                        "Enables or disables the reception of LLDP packets
                         on an interface.";
                }
                enum tx-and-rx {
                    description
                        "Enables or disables the reception and transimission
                         of LLDP packets on an interface. After you globally
                         enable LLDP, tx and rx is enabled on all interfaces
                         by default.";
                }
                enum disabled {
```


Copyright© 2022 Ciena® Corporation

```
                description
                    "Disable transmission and reception of LLDP packets
                     on this interface.";
            }
        }
        default "tx-and-rx";
    }

    leaf notification {
        type boolean;
        description
            "Enable or disable the sending of notification.";
    }

    leaf tlv-select {
        type bits {
            bit management-address {
                description
                    "Transmit the management address TLV. The list of
                     possible management addresses can be selected
                     from the list defined below.";
            }
            bit port-description {
                description
                    "";
            }
            bit system-capabilities {
                description
                    "";
            }
            bit  system-description {
                description
                    "";
            }
            bit system-name {
                description
                    "";
            }
        }
    }

    leaf management-address-type {
        type bits {
            bit management-address-local;
            bit management-address-local-ipv6;
            bit management-address-remote;
            bit management-address-remote-ipv6;
        }
    }

    leaf dot3-tlv {
        type bits {
            bit mac-phy-config-status {
                description
                   "Transmit 'MAC/PHY configuration/status TLV. Indicates
                     the auto-negotiation capability and the duplex/speed
                      status of IEEE 802.3 MAC/PHYs.";
            }
            bit power-via-mdi {
                description
```


Copyright© 2022 Ciena® Corporation

```
                    "Transmit Power via MDI TLV. Indicates the capabilities
                       and current status of IEEE 802.3 PMDs that either
                       require or are able to provide power over twisted
                       pair copper wire.";
                }
                bit link-aggregation {
                    description
                        "Transmit link aggregation TLV. Indicates the link
                         aggregation status of IEEE 802.3 MACs.";
                }
                bit max-frame-size {
                    description
                        "Transmit maximum frame size TLV.";
                }
            }
        }

        leaf dot1-tlv {
            type bits {
                bit port-vlan-id {
                    description
                        "Transmit the Port VLAN ID TLV.";
                }
                bit port-protocol {
                    description
                        "Transmit Port and VLAN ID TLV.";
                }
                bit vlan-name {
                    description
                        "Transmit the VLAN name TLV.";
                }
                bit protocol-identity {
                    description
                        "Transmit the protocol TLV (protocol list defined
                         below.";
                }
                bit vid-usage-digest {
                    description
                        "Transmit the VID usage digest TLV.";
                }
                bit management-vid {
                    description
                        "Transmit the Management VID TLV.";
                }
                bit link-aggregation {
                    description
                        "Transmit the Link Aggregation TLV.";
                }
            }
        }

        leaf protocol-list {
            type bits {
                bit dot1x;
                bit lacp;
                bit oam;
                bit stp;
            }
        }
    }
```


Copyright© 2022 Ciena® Corporation

```
    augment "/if:interfaces-state/if:interface" {
        when "if:type = 'ianaift:ethernetCsmacd'";

        leaf tx-frames-total {
            type yang:counter32;
            description
                "Total number of LLDP frames transmitted out of this port.";
        }

        leaf rx-frames-discarded {
            type yang:counter32;
            description
                "Total number of LLDP frames discarded on receive for any
                 reason.";
        }

        leaf rx-frames-errors {
            type yang:counter32;
            description
                "Total number of LLDP frames that were received with errors
                 in them.";
        }

        leaf rx-frames-total {
            type yang:counter32;
            description
                "Total number of LLDP frames received on this port.";
        }

        leaf rx-tlv-discarded {
            type yang:counter32;
            description
                "Total number of LLDP TLV frames that were discarded on
                 receive for any reason.";
        }

        leaf rx-tlv-unrecognized {
            type yang:counter32;
            description
                "Total number of LLDP TLV frames that were unrecognized and
                 discarded (?)";
        }

        leaf rx-stats-ageout {
            type yang:zero-based-counter32;
            description
                "Total number of times the LLDP stats on the receive were
                 aged out.";
        }
    }
}
```


Copyright© 2022 Ciena® Corporation

ieee-types.yang

```
module ieee-types {

  namespace "urn:ieee:params:xml:ns:yang:ieee-types";
  prefix "ieee";

  organization
      "Institute of Electrical and Electronics Engineers";

    contact
      "Web URL: http://www.ieee802.org

       Editor:
       Marc Holness
       mholness@ciena.com";

  description
     "This module contains a collection of generally useful derived
     data types for IEEE YANG models.";

  revision 2015-09-10 {
     description
        "Initial revision.";
      reference
          "IEEE 802";
  }

  /*
   * Collection of IEEE address type definitions
   */

  typedef mac-address {
     type string {
        pattern '[0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){5}';
      }
     description
        "The mac-address type represents an 802 MAC address
        represented in the canonical order defined by
        IEEE 802.";
     reference
        "IEEE 802.3-2012, Clause 3.2.3";
  }

  /*
   * Collection of IEEE 802 related identifier types
   */

  typedef bridgeid {
     type string {
        pattern '[0-9a-fA-F]{4}(:[0-9a-fA-F]{2}){6}';
      }
     description
        "The bridgeid type represents identifiers that uniquely
        identify a bridge.  Its first four hexadecimal digits
        contain a priority value followed by a colon. The
        remaining characters contain the MAC address used to
        refer to a bridge in a unique fashion (typically, the
        numerically smallest MAC address of all ports on the
        bridge).
```


Copyright© 2022 Ciena® Corporation

```
          This type is in the value set and its semantics equivalent
          to the BridgeId textual convention of the SMIv2. However,
          since the BridgeId textual convention does not prescribe
          a lexical representation, the appearance might be different.";
      reference
      "TBD";
  }

  typedef vlanid {
     type uint16 {
        range "1..4094";
     }
     description
        "The vlanid type uniquely identifies a VLAN. This is the
        12-bit VLAN-ID used in the VLAN Tag header. The range is
        defined by the referenced specification.

        This type is in the value set and its semantics equivalent
        to the VlanId textual convention of the SMIv2.";
     reference
        "IEEE Std 802.1Q-2014: Virtual Bridged Local Area Networks.";
  }

}
```


Copyright© 2022 Ciena® Corporation

ietf-inet-types.yang

```
module ietf-inet-types {
  namespace "urn:ietf:params:xml:ns:yang:ietf-inet-types";
  prefix "inet";
  organization
   "IETF NETMOD (NETCONF Data Modeling Language) Working Group";
  contact
   "WG Web:   <http://tools.ietf.org/wg/netmod/>
    WG List:  <mailto:netmod@ietf.org>
    WG Chair: David Kessens
              <mailto:david.kessens@nsn.com>
    WG Chair: Juergen Schoenwaelder
              <mailto:j.schoenwaelder@jacobs-university.de>
    Editor:   Juergen Schoenwaelder
              <mailto:j.schoenwaelder@jacobs-university.de>";
  description
   "This module contains a collection of generally useful derived
    YANG data types for Internet addresses and related things.
    Copyright (c) 2013 IETF Trust and the persons identified as
    authors of the code.  All rights reserved.
    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject
    to the license terms contained in, the Simplified BSD License
    set forth in Section 4.c of the IETF Trust's Legal Provisions
    Relating to IETF Documents
    (http://trustee.ietf.org/license-info).
    This version of this YANG module is part of RFC 6991; see
    the RFC itself for full legal notices.";
  revision 2013-07-15 {
    description
     "This revision adds the following new data types:
      - ip-address-no-zone
      - ipv4-address-no-zone
      - ipv6-address-no-zone";
    reference
     "RFC 6991: Common YANG Data Types";
  }
  revision 2010-09-24 {
    description
     "Initial revision.";
    reference
     "RFC 6021: Common YANG Data Types";
  }
  /*** collection of types related to protocol fields ***/
  typedef ip-version {
    type enumeration {
      enum unknown {
        value "0";
        description
         "An unknown or unspecified version of the Internet
          protocol.";
      }
      enum ipv4 {
        value "1";
        description
         "The IPv4 protocol as defined in RFC 791.";
      }
      enum ipv6 {
        value "2";
```


Copyright© 2022 Ciena® Corporation

```
        description
          "The IPv6 protocol as defined in RFC 2460.";
      }
    }
    description
     "This value represents the version of the IP protocol.
      In the value set and its semantics, this type is equivalent
      to the InetVersion textual convention of the SMIv2.";
    reference
     "RFC  791: Internet Protocol
      RFC 2460: Internet Protocol, Version 6 (IPv6) Specification
      RFC 4001: Textual Conventions for Internet Network Addresses";
  }
  typedef dscp {
    type uint8 {
      range "0..63";
    }
    description
     "The dscp type represents a Differentiated Services Code Point
      that may be used for marking packets in a traffic stream.
      In the value set and its semantics, this type is equivalent
      to the Dscp textual convention of the SMIv2.";
    reference
     "RFC 3289: Management Information Base for the Differentiated
                Services Architecture
      RFC 2474: Definition of the Differentiated Services Field
                (DS Field) in the IPv4 and IPv6 Headers
      RFC 2780: IANA Allocation Guidelines For Values In
                the Internet Protocol and Related Headers";
  }
  typedef ipv6-flow-label {
    type uint32 {
      range "0..1048575";
    }
    description
     "The ipv6-flow-label type represents the flow identifier or Flow
      Label in an IPv6 packet header that may be used to
      discriminate traffic flows.
      In the value set and its semantics, this type is equivalent
      to the IPv6FlowLabel textual convention of the SMIv2.";
    reference
     "RFC 3595: Textual Conventions for IPv6 Flow Label
      RFC 2460: Internet Protocol, Version 6 (IPv6) Specification";
  }
  typedef port-number {
    type uint16 {
      range "0..65535";
    }
    description
     "The port-number type represents a 16-bit port number of an
      Internet transport-layer protocol such as UDP, TCP, DCCP, or
      SCTP.  Port numbers are assigned by IANA.  A current list of
      all assignments is available from <http://www.iana.org/>.
      Note that the port number value zero is reserved by IANA.  In
      situations where the value zero does not make sense, it can
      be excluded by subtyping the port-number type.
      In the value set and its semantics, this type is equivalent
      to the InetPortNumber textual convention of the SMIv2.";
    reference
     "RFC  768: User Datagram Protocol
```


Copyright© 2022 Ciena® Corporation

```
      RFC  793: Transmission Control Protocol
      RFC 4960: Stream Control Transmission Protocol
      RFC 4340: Datagram Congestion Control Protocol (DCCP)
      RFC 4001: Textual Conventions for Internet Network Addresses";
  }
  /*** collection of types related to autonomous systems ***/
  typedef as-number {
    type uint32;
    description
     "The as-number type represents autonomous system numbers
      which identify an Autonomous System (AS).  An AS is a set
      of routers under a single technical administration, using
      an interior gateway protocol and common metrics to route
      packets within the AS, and using an exterior gateway
      protocol to route packets to other ASes.  IANA maintains
      the AS number space and has delegated large parts to the
      regional registries.
      Autonomous system numbers were originally limited to 16
      bits.  BGP extensions have enlarged the autonomous system
      number space to 32 bits.  This type therefore uses an uint32
      base type without a range restriction in order to support
      a larger autonomous system number space.
      In the value set and its semantics, this type is equivalent
      to the InetAutonomousSystemNumber textual convention of
      the SMIv2.";
    reference
     "RFC 1930: Guidelines for creation, selection, and registration
                of an Autonomous System (AS)
      RFC 4271: A Border Gateway Protocol 4 (BGP-4)
      RFC 4001: Textual Conventions for Internet Network Addresses
      RFC 6793: BGP Support for Four-Octet Autonomous System (AS)
                Number Space";
  }
  /*** collection of types related to IP addresses and hostnames ***/
  typedef ip-address {
    type union {
      type inet:ipv4-address;
      type inet:ipv6-address;
    }
    description
     "The ip-address type represents an IP address and is IP
      version neutral.  The format of the textual representation
      implies the IP version.  This type supports scoped addresses
      by allowing zone identifiers in the address format.";
    reference
     "RFC 4007: IPv6 Scoped Address Architecture";
  }
  typedef ipv4-address {
    type string {
      pattern
        '(([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.){3}'
      +  '([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])'
      + '(%[\p{N}\p{L}]+)?';
    }
    description
      "The ipv4-address type represents an IPv4 address in
       dotted-quad notation.  The IPv4 address may include a zone
       index, separated by a % sign.
       The zone index is used to disambiguate identical address
       values.  For link-local addresses, the zone index will
```


Copyright© 2022 Ciena® Corporation

```
      typically be the interface index number or the name of an
      interface.  If the zone index is not present, the default
      zone of the device will be used.
      The canonical format for the zone index is the numerical
      format";
  }
  typedef ipv6-address {
    type string {
      pattern '((:|[0-9a-fA-F]{0,4}):)([0-9a-fA-F]{0,4}:){0,5}'
            + '((([0-9a-fA-F]{0,4}:)?(:|[0-9a-fA-F]{0,4}))|'
            + '(((25[0-5]|2[0-4][0-9]|[01]?[0-9]?[0-9])\.){3}'
            + '(25[0-5]|2[0-4][0-9]|[01]?[0-9]?[0-9])))'
            + '(%[\p{N}\p{L}]+)?';
      pattern '(([^:]+:){6}(([^:]+:[^:]+)|(.*\..*))|'
            + '((([^:]+:)*[^:]+)?::(([^:]+:)*[^:]+)?)'
            + '(%.+)?';
    }
    description
     "The ipv6-address type represents an IPv6 address in full,
      mixed, shortened, and shortened-mixed notation.  The IPv6
      address may include a zone index, separated by a % sign.
      The zone index is used to disambiguate identical address
      values.  For link-local addresses, the zone index will
      typically be the interface index number or the name of an
      interface.  If the zone index is not present, the default
      zone of the device will be used.
      The canonical format of IPv6 addresses uses the textual
      representation defined in Section 4 of RFC 5952.  The
      canonical format for the zone index is the numerical
      format as described in Section 11.2 of RFC 4007.";
    reference
     "RFC 4291: IP Version 6 Addressing Architecture
      RFC 4007: IPv6 Scoped Address Architecture
      RFC 5952: A Recommendation for IPv6 Address Text
                Representation";
  }
  typedef ip-address-no-zone {
    type union {
      type inet:ipv4-address-no-zone;
      type inet:ipv6-address-no-zone;
    }
    description
     "The ip-address-no-zone type represents an IP address and is
      IP version neutral.  The format of the textual representation
      implies the IP version.  This type does not support scoped
      addresses since it does not allow zone identifiers in the
      address format.";
    reference
     "RFC 4007: IPv6 Scoped Address Architecture";
  }
  typedef ipv4-address-no-zone {
    type inet:ipv4-address {
      pattern '[0-9\.]*';
    }
    description
      "An IPv4 address without a zone index.  This type, derived from
       ipv4-address, may be used in situations where the zone is
       known from the context and hence no zone index is needed.";
  }
  typedef ipv6-address-no-zone {
```


Copyright© 2022 Ciena® Corporation

```
    type inet:ipv6-address {
      pattern '[0-9a-fA-F:\.]*';
    }
    description
      "An IPv6 address without a zone index.  This type, derived from
       ipv6-address, may be used in situations where the zone is
       known from the context and hence no zone index is needed.";
    reference
     "RFC 4291: IP Version 6 Addressing Architecture
      RFC 4007: IPv6 Scoped Address Architecture
      RFC 5952: A Recommendation for IPv6 Address Text
                Representation";
  }
  typedef ip-prefix {
    type union {
      type inet:ipv4-prefix;
      type inet:ipv6-prefix;
    }
    description
     "The ip-prefix type represents an IP prefix and is IP
      version neutral.  The format of the textual representations
      implies the IP version.";
  }
  typedef ipv4-prefix {
    type string {
      pattern
         '(([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.){3}'
       +  '([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])'
       + '/(([0-9])|([1-2][0-9])|(3[0-2]))';
    }
    description
     "The ipv4-prefix type represents an IPv4 address prefix.
      The prefix length is given by the number following the
      slash character and must be less than or equal to 32.
      A prefix length value of n corresponds to an IP address
      mask that has n contiguous 1-bits from the most
      significant bit (MSB) and all other bits set to 0.
      The canonical format of an IPv4 prefix has all bits of
      the IPv4 address set to zero that are not part of the
      IPv4 prefix.";
  }
  typedef ipv6-prefix {
    type string {
      pattern '((:|[0-9a-fA-F]{0,4}):)([0-9a-fA-F]{0,4}:){0,5}'
            + '((([0-9a-fA-F]{0,4}:)?(:|[0-9a-fA-F]{0,4}))|'
            + '(((25[0-5]|2[0-4][0-9]|[01]?[0-9]?[0-9])\.){3}'
            + '(25[0-5]|2[0-4][0-9]|[01]?[0-9]?[0-9])))'
            + '(/(([0-9])|([0-9]{2})|(1[0-1][0-9])|(12[0-8])))';
      pattern '(([^:]+:){6}(([^:]+:[^:]+)|(.*\..*))|'
            + '((([^:]+:)*[^:]+)?::(([^:]+:)*[^:]+)?)'
            + '(/.+)';
    }
    description
     "The ipv6-prefix type represents an IPv6 address prefix.
      The prefix length is given by the number following the
      slash character and must be less than or equal to 128.
      A prefix length value of n corresponds to an IP address
      mask that has n contiguous 1-bits from the most
      significant bit (MSB) and all other bits set to 0.
      The IPv6 address should have all bits that do not belong
```


Copyright© 2022 Ciena® Corporation

```
      to the prefix set to zero.
      The canonical format of an IPv6 prefix has all bits of
      the IPv6 address set to zero that are not part of the
      IPv6 prefix.  Furthermore, the IPv6 address is represented
      as defined in Section 4 of RFC 5952.";
    reference
     "RFC 5952: A Recommendation for IPv6 Address Text
                Representation";
  }
  /*** collection of domain name and URI types ***/
  typedef domain-name {
    type string {
      pattern
        '((([a-zA-Z0-9_]([a-zA-Z0-9\-_]){0,61})?[a-zA-Z0-9]\.)*'
      + '([a-zA-Z0-9_]([a-zA-Z0-9\-_]){0,61})?[a-zA-Z0-9]\.?)'
      + '|\.';
      length "1..253";
    }
    description
     "The domain-name type represents a DNS domain name.  The
      name SHOULD be fully qualified whenever possible.
      Internet domain names are only loosely specified.  Section
      3.5 of RFC 1034 recommends a syntax (modified in Section
      2.1 of RFC 1123).  The pattern above is intended to allow
      for current practice in domain name use, and some possible
      future expansion.  It is designed to hold various types of
      domain names, including names used for A or AAAA records
      (host names) and other records, such as SRV records.  Note
      that Internet host names have a stricter syntax (described
      in RFC 952) than the DNS recommendations in RFCs 1034 and
      1123, and that systems that want to store host names in
      schema nodes using the domain-name type are recommended to
      adhere to this stricter standard to ensure interoperability.
      The encoding of DNS names in the DNS protocol is limited
      to 255 characters.  Since the encoding consists of labels
      prefixed by a length bytes and there is a trailing NULL
      byte, only 253 characters can appear in the textual dotted
      notation.
      The description clause of schema nodes using the domain-name
      type MUST describe when and how these names are resolved to
      IP addresses.  Note that the resolution of a domain-name value
      may require to query multiple DNS records (e.g., A for IPv4
      and AAAA for IPv6).  The order of the resolution process and
      which DNS record takes precedence can either be defined
      explicitly or may depend on the configuration of the
      resolver.
      Domain-name values use the US-ASCII encoding.  Their canonical
      format uses lowercase US-ASCII characters.  Internationalized
      domain names MUST be A-labels as per RFC 5890.";
    reference
     "RFC  952: DoD Internet Host Table Specification
      RFC 1034: Domain Names - Concepts and Facilities
      RFC 1123: Requirements for Internet Hosts -- Application
                and Support
      RFC 2782: A DNS RR for specifying the location of services
                (DNS SRV)
      RFC 5890: Internationalized Domain Names in Applications
                (IDNA): Definitions and Document Framework";
  }
  typedef host {
```


Copyright© 2022 Ciena® Corporation

```
    type union {
      type inet:ip-address;
      type inet:domain-name;
    }
    description
     "The host type represents either an IP address or a DNS
      domain name.";
  }
  typedef uri {
    type string;
    description
     "The uri type represents a Uniform Resource Identifier
      (URI) as defined by STD 66.
      Objects using the uri type MUST be in US-ASCII encoding,
      and MUST be normalized as described by RFC 3986 Sections
      6.2.1, 6.2.2.1, and 6.2.2.2.  All unnecessary
      percent-encoding is removed, and all case-insensitive
      characters are set to lowercase except for hexadecimal
      digits, which are normalized to uppercase as described in
      Section 6.2.2.1.
      The purpose of this normalization is to help provide
      unique URIs.  Note that this normalization is not
      sufficient to provide uniqueness.  Two URIs that are
      textually distinct after this normalization may still be
      equivalent.
      Objects using the uri type may restrict the schemes that
      they permit.  For example, 'data:' and 'urn:' schemes
      might not be appropriate.
      A zero-length URI is not a valid URI.  This can be used to
      express 'URI absent' where required.
      In the value set and its semantics, this type is equivalent
      to the Uri SMIv2 textual convention defined in RFC 5017.";
    reference
     "RFC 3986: Uniform Resource Identifier (URI): Generic Syntax
      RFC 3305: Report from the Joint W3C/IETF URI Planning Interest
                Group: Uniform Resource Identifiers (URIs), URLs,
                and Uniform Resource Names (URNs): Clarifications
                and Recommendations
      RFC 5017: MIB Textual Conventions for Uniform Resource
                Identifiers (URIs)";
  }
}
```


Copyright© 2022 Ciena® Corporation

ietf-inet-types@2010-09-24.yang

```
module ietf-inet-types {

  namespace "urn:ietf:params:xml:ns:yang:ietf-inet-types";
  prefix "inet";

  organization
   "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

  contact
   "WG Web:   <http://tools.ietf.org/wg/netmod/>
    WG List:  <mailto:netmod@ietf.org>

    WG Chair: David Partain
              <mailto:david.partain@ericsson.com>

    WG Chair: David Kessens
              <mailto:david.kessens@nsn.com>

    Editor:   Juergen Schoenwaelder
              <mailto:j.schoenwaelder@jacobs-university.de>";

  description
   "This module contains a collection of generally useful derived
    YANG data types for Internet addresses and related things.

    Copyright (c) 2010 IETF Trust and the persons identified as
    authors of the code.  All rights reserved.



    Redistribution and use in source and binary forms, with or without
    modification, is permitted pursuant to, and subject to the license
    terms contained in, the Simplified BSD License set forth in Section
    4.c of the IETF Trust's Legal Provisions Relating to IETF Documents
    (http://trustee.ietf.org/license-info).

    This version of this YANG module is part of RFC 6021; see
    the RFC itself for full legal notices.";

  revision 2010-09-24 {
    description
     "Initial revision.";
    reference
     "RFC 6021: Common YANG Data Types";
  }

  /*** collection of protocol field related types ***/

  typedef ip-version {
    type enumeration {
      enum unknown {
        value "0";
        description
         "An unknown or unspecified version of the Internet protocol.";
      }
      enum ipv4 {
        value "1";
        description
```


Copyright© 2022 Ciena® Corporation

```
        "The IPv4 protocol as defined in RFC 791.";
      }
      enum ipv6 {
        value "2";
        description
         "The IPv6 protocol as defined in RFC 2460.";
      }
    }
    description
     "This value represents the version of the IP protocol.

      In the value set and its semantics, this type is equivalent
      to the InetVersion textual convention of the SMIv2.";
    reference
     "RFC  791: Internet Protocol
      RFC 2460: Internet Protocol, Version 6 (IPv6) Specification
      RFC 4001: Textual Conventions for Internet Network Addresses";
  }

  typedef dscp {
    type uint8 {
      range "0..63";
    }
    description
     "The dscp type represents a Differentiated Services Code-Point
      that may be used for marking packets in a traffic stream.

      In the value set and its semantics, this type is equivalent
      to the Dscp textual convention of the SMIv2.";
    reference
     "RFC 3289: Management Information Base for the Differentiated
                Services Architecture
      RFC 2474: Definition of the Differentiated Services Field
                (DS Field) in the IPv4 and IPv6 Headers
      RFC 2780: IANA Allocation Guidelines For Values In
                the Internet Protocol and Related Headers";
  }

  typedef ipv6-flow-label {
    type uint32 {
      range "0..1048575";
    }
    description
     "The flow-label type represents flow identifier or Flow Label
      in an IPv6 packet header that may be used to discriminate
      traffic flows.

      In the value set and its semantics, this type is equivalent
      to the IPv6FlowLabel textual convention of the SMIv2.";
    reference
     "RFC 3595: Textual Conventions for IPv6 Flow Label
      RFC 2460: Internet Protocol, Version 6 (IPv6) Specification";
  }

  typedef port-number {
    type uint16 {
      range "0..65535";
    }
    description
     "The port-number type represents a 16-bit port number of an
```


Copyright© 2022 Ciena® Corporation

```
      Internet transport layer protocol such as UDP, TCP, DCCP, or
      SCTP.  Port numbers are assigned by IANA.  A current list of
      all assignments is available from <http://www.iana.org/>.

      Note that the port number value zero is reserved by IANA.  In
      situations where the value zero does not make sense, it can
      be excluded by subtyping the port-number type.

      In the value set and its semantics, this type is equivalent
      to the InetPortNumber textual convention of the SMIv2.";
    reference
     "RFC  768: User Datagram Protocol
      RFC  793: Transmission Control Protocol
      RFC 4960: Stream Control Transmission Protocol
      RFC 4340: Datagram Congestion Control Protocol (DCCP)
      RFC 4001: Textual Conventions for Internet Network Addresses";
  }

  /*** collection of autonomous system related types ***/

  typedef as-number {
    type uint32;
    description
     "The as-number type represents autonomous system numbers
      which identify an Autonomous System (AS).  An AS is a set
      of routers under a single technical administration, using
      an interior gateway protocol and common metrics to route
      packets within the AS, and using an exterior gateway
      protocol to route packets to other ASs'.  IANA maintains
      the AS number space and has delegated large parts to the
      regional registries.

      Autonomous system numbers were originally limited to 16
      bits.  BGP extensions have enlarged the autonomous system
      number space to 32 bits.  This type therefore uses an uint32
      base type without a range restriction in order to support
      a larger autonomous system number space.

      In the value set and its semantics, this type is equivalent
      to the InetAutonomousSystemNumber textual convention of
      the SMIv2.";
    reference
     "RFC 1930: Guidelines for creation, selection, and registration
                of an Autonomous System (AS)
      RFC 4271: A Border Gateway Protocol 4 (BGP-4)
      RFC 4893: BGP Support for Four-octet AS Number Space
      RFC 4001: Textual Conventions for Internet Network Addresses";
  }

  /*** collection of IP address and hostname related types ***/

  typedef ip-address {
    type union {
      type inet:ipv4-address;
      type inet:ipv6-address;
    }
    description
     "The ip-address type represents an IP address and is IP
      version neutral.  The format of the textual representations
      implies the IP version.";
```


Copyright© 2022 Ciena® Corporation

```
}

typedef ipv4-address {
  type string {
    pattern
      '(([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.){3}'
    +  '([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])'
    + '(%[\p{N}\p{L}]+)?';
  }
  description
    "The ipv4-address type represents an IPv4 address in
     dotted-quad notation.  The IPv4 address may include a zone
     index, separated by a % sign.

     The zone index is used to disambiguate identical address
     values.  For link-local addresses, the zone index will
     typically be the interface index number or the name of an
     interface.  If the zone index is not present, the default
     zone of the device will be used.

     The canonical format for the zone index is the numerical
     format";
}

typedef ipv6-address {
  type string {
    pattern '((:|[0-9a-fA-F]{0,4}):)([0-9a-fA-F]{0,4}:){0,5}'
          + '((([0-9a-fA-F]{0,4}:)?(:|[0-9a-fA-F]{0,4}))|'
          + '((25[0-5]|2[0-4][0-9]|[01]?[0-9]?[0-9])\.){3}'
          + '(25[0-5]|2[0-4][0-9]|[01]?[0-9]?[0-9])))'
          + '(%[\p{N}\p{L}]+)?';
    pattern '(([^:]+:){6}(([^:]+:[^:]+)|(.*\..*))|'
          + '((([^:]+:)*[^:]+)?::(([^:]+:)*[^:]+)?)'
          + '(%.+)?';
  }
  description
   "The ipv6-address type represents an IPv6 address in full,
    mixed, shortened, and shortened-mixed notation.  The IPv6
    address may include a zone index, separated by a % sign.




    The zone index is used to disambiguate identical address
    values.  For link-local addresses, the zone index will
    typically be the interface index number or the name of an
    interface.  If the zone index is not present, the default
    zone of the device will be used.

    The canonical format of IPv6 addresses uses the compressed
    format described in RFC 4291, Section 2.2, item 2 with the
    following additional rules: the :: substitution must be
    applied to the longest sequence of all-zero 16-bit chunks
    in an IPv6 address.  If there is a tie, the first sequence
    of all-zero 16-bit chunks is replaced by ::.  Single
    all-zero 16-bit chunks are not compressed.  The canonical
    format uses lowercase characters and leading zeros are
    not allowed.  The canonical format for the zone index is
    the numerical format as described in RFC 4007, Section
```


Copyright© 2022 Ciena® Corporation

```
    11.2.";
  reference
   "RFC 4291: IP Version 6 Addressing Architecture
    RFC 4007: IPv6 Scoped Address Architecture
    RFC 5952: A Recommendation for IPv6 Address Text Representation";
}

typedef ip-prefix {
  type union {
    type inet:ipv4-prefix;
    type inet:ipv6-prefix;
  }
  description
   "The ip-prefix type represents an IP prefix and is IP
    version neutral.  The format of the textual representations
    implies the IP version.";
}

typedef ipv4-prefix {
  type string {
    pattern
       '(([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.){3}'
     +  '([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])'
     + '/(([0-9])|([1-2][0-9])|(3[0-2]))';
  }
  description
   "The ipv4-prefix type represents an IPv4 address prefix.
    The prefix length is given by the number following the
    slash character and must be less than or equal to 32.



    A prefix length value of n corresponds to an IP address
    mask that has n contiguous 1-bits from the most
    significant bit (MSB) and all other bits set to 0.

    The canonical format of an IPv4 prefix has all bits of
    the IPv4 address set to zero that are not part of the
    IPv4 prefix.";
}

typedef ipv6-prefix {
  type string {
    pattern '((:|[0-9a-fA-F]{0,4}):)([0-9a-fA-F]{0,4}:){0,5}'
          + '((([0-9a-fA-F]{0,4}:)?(:|[0-9a-fA-F]{0,4}))|'
          + '(((25[0-5]|2[0-4][0-9]|[01]?[0-9]?[0-9])\.){3}'
          + '(25[0-5]|2[0-4][0-9]|[01]?[0-9]?[0-9])))'
          + '(/(([0-9])|([0-9]{2})|(1[0-1][0-9])|(12[0-8])))';
    pattern '(([^:]+:){6}(([^:]+:[^:]+)|(.*\..*)))|'
          + '((([^:]+:)*[^:]+)?::(([^:]+:)*[^:]+)?)'
          + '(/.+)';
  }
  description
   "The ipv6-prefix type represents an IPv6 address prefix.
    The prefix length is given by the number following the
    slash character and must be less than or equal 128.

    A prefix length value of n corresponds to an IP address
    mask that has n contiguous 1-bits from the most
    significant bit (MSB) and all other bits set to 0.
```


Copyright© 2022 Ciena® Corporation

```
      The IPv6 address should have all bits that do not belong
      to the prefix set to zero.

      The canonical format of an IPv6 prefix has all bits of
      the IPv6 address set to zero that are not part of the
      IPv6 prefix.  Furthermore, IPv6 address is represented
      in the compressed format described in RFC 4291, Section
      2.2, item 2 with the following additional rules: the ::
      substitution must be applied to the longest sequence of
      all-zero 16-bit chunks in an IPv6 address.  If there is
      a tie, the first sequence of all-zero 16-bit chunks is
      replaced by ::.  Single all-zero 16-bit chunks are not
      compressed.  The canonical format uses lowercase
      characters and leading zeros are not allowed.";
    reference
     "RFC 4291: IP Version 6 Addressing Architecture";
  }


  /*** collection of domain name and URI types ***/

  typedef domain-name {
    type string {
      pattern '((([a-zA-Z0-9_]([a-zA-Z0-9\-_]){0,61})?[a-zA-Z0-9]\.)*'
            +  '([a-zA-Z0-9_]([a-zA-Z0-9\-_]){0,61})?[a-zA-Z0-9]\.?)'
            +  '|\.';
      length "1..253";
    }
    description
     "The domain-name type represents a DNS domain name.  The
      name SHOULD be fully qualified whenever possible.

      Internet domain names are only loosely specified.  Section
      3.5 of RFC 1034 recommends a syntax (modified in Section
      2.1 of RFC 1123).  The pattern above is intended to allow
      for current practice in domain name use, and some possible
      future expansion.  It is designed to hold various types of
      domain names, including names used for A or AAAA records
      (host names) and other records, such as SRV records.  Note
      that Internet host names have a stricter syntax (described
      in RFC 952) than the DNS recommendations in RFCs 1034 and
      1123, and that systems that want to store host names in
      schema nodes using the domain-name type are recommended to
      adhere to this stricter standard to ensure interoperability.

      The encoding of DNS names in the DNS protocol is limited
      to 255 characters.  Since the encoding consists of labels
      prefixed by a length bytes and there is a trailing NULL
      byte, only 253 characters can appear in the textual dotted
      notation.

      The description clause of schema nodes using the domain-name
      type MUST describe when and how these names are resolved to
      IP addresses.  Note that the resolution of a domain-name value
      may require to query multiple DNS records (e.g., A for IPv4
      and AAAA for IPv6).  The order of the resolution process and
      which DNS record takes precedence can either be defined
      explicitely or it may depend on the configuration of the
      resolver.
```


Copyright© 2022 Ciena® Corporation

```
      Domain-name values use the US-ASCII encoding.  Their canonical
      format uses lowercase US-ASCII characters.  Internationalized
      domain names MUST be encoded in punycode as described in RFC
      3492";
    reference
     "RFC  952: DoD Internet Host Table Specification
      RFC 1034: Domain Names - Concepts and Facilities
      RFC 1123: Requirements for Internet Hosts -- Application
                and Support
      RFC 2782: A DNS RR for specifying the location of services
                (DNS SRV)
      RFC 3492: Punycode: A Bootstring encoding of Unicode for
                Internationalized Domain Names in Applications
                (IDNA)
      RFC 5891: Internationalizing Domain Names in Applications
                (IDNA): Protocol";
  }

  typedef host {
    type union {
      type inet:ip-address;
      type inet:domain-name;
    }
    description
     "The host type represents either an IP address or a DNS
      domain name.";
  }

  typedef uri {
    type string;
    description
     "The uri type represents a Uniform Resource Identifier
      (URI) as defined by STD 66.

      Objects using the uri type MUST be in US-ASCII encoding,
      and MUST be normalized as described by RFC 3986 Sections
      6.2.1, 6.2.2.1, and 6.2.2.2.  All unnecessary
      percent-encoding is removed, and all case-insensitive
      characters are set to lowercase except for hexadecimal
      digits, which are normalized to uppercase as described in
      Section 6.2.2.1.

      The purpose of this normalization is to help provide
      unique URIs.  Note that this normalization is not
      sufficient to provide uniqueness.  Two URIs that are
      textually distinct after this normalization may still be
      equivalent.

      Objects using the uri type may restrict the schemes that
      they permit.  For example, 'data:' and 'urn:' schemes
      might not be appropriate.

      A zero-length URI is not a valid URI.  This can be used to
      express 'URI absent' where required.

      In the value set and its semantics, this type is equivalent
      to the Uri SMIv2 textual convention defined in RFC 5017.";
    reference
     "RFC 3986: Uniform Resource Identifier (URI): Generic Syntax
```


Copyright© 2022 Ciena® Corporation

```
      RFC 3305: Report from the Joint W3C/IETF URI Planning Interest
                Group: Uniform Resource Identifiers (URIs), URLs,
                and Uniform Resource Names (URNs): Clarifications
                and Recommendations
      RFC 5017: MIB Textual Conventions for Uniform Resource
                Identifiers (URIs)";
  }

}
```


Copyright© 2022 Ciena® Corporation

ietf-interfaces.yang

```
module ietf-interfaces {

  namespace "urn:ietf:params:xml:ns:yang:ietf-interfaces";
  prefix if;

  import ietf-yang-types {
    prefix yang;
  }

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

  contact
    "WG Web:   http://tools.ietf.org/wg/netmod/;
     WG List:  mailto:netmod@ietf.org;

     WG Chair: Thomas Nadeau
               mailto:tnadeau@lucidvision.com;

     WG Chair: Juergen Schoenwaelder
               mailto:j.schoenwaelder@jacobs-university.de;

     Editor:   Martin Bjorklund
               mailto:mbj@tail-f.com;";

  description
    "This module contains a collection of YANG definitions for
     managing network interfaces.

     Copyright (c) 2014 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject
     to the license terms contained in, the Simplified BSD License
     set forth in Section 4 of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     http://trustee.ietf.org/license-info.

     This version of this YANG module is part of RFC 7223 see
     the RFC itself for full legal notices.";


  revision 2014-05-08 {
    description
      "Initial revision.";
    reference
      "RFC 7223: A YANG Data Model for Interface Management";
  }

  /*
   * Typedefs
   */

  typedef interface-ref {
    type leafref {
      path "/if:interfaces/if:interface/if:name";
    }
```


Copyright© 2022 Ciena® Corporation

```
    description
      "This type is used by data models that need to reference
       configured interfaces.";
  }

  typedef interface-state-ref {
    type leafref {
      path "/if:interfaces-state/if:interface/if:name";
    }
    description
      "This type is used by data models that need to reference
       the operationally present interfaces.";
  }

  /*
   * Identities
   */

  identity interface-type {
    description
      "Base identity from which specific interface types are
       derived.";
  }

  /*
   * Features
   */

  feature arbitrary-names {
    description
      "This feature indicates that the device allows user-controlled
       interfaces to be named arbitrarily.";
  }

  feature pre-provisioning {
    description
      "This feature indicates that the device supports
       pre-provisioning of interface configuration, i.e., it is
       possible to configure an interface whose physical interface
       hardware is not present on the device.";
  }

  feature if-mib {
    description
      "This feature indicates that the device implements
       the IF-MIB.";
    reference
      "RFC 2863 The Interfaces Group MIB";
  }

  /*
   * Configuration data nodes
   */

  container interfaces {
    description
      "Interface configuration parameters.";

    list interface {
      key "name";
```


Copyright© 2022 Ciena® Corporation

```
 description
   "The list of configured interfaces on the device.

    The operational state of an interface is available in the
    /interfaces-state/interface list.  If the configuration of a
    system-controlled interface cannot be used by the system
    (e.g., the interface hardware present does not match the
    interface type), then the configuration is not applied to
    the system-controlled interface shown in the
    /interfaces-state/interface list.  If the configuration
    of a user-controlled interface cannot be used by the system,
    the configured interface is not instantiated in the
    /interfaces-state/interface list.";

leaf name {
   type string;
   description
     "The name of the interface.

      A device MAY restrict the allowed values for this leaf,
      possibly depending on the type of the interface.

      For system-controlled interfaces, this leaf is the
      device-specific name of the interface.  The 'config false'
      list /interfaces-state/interface contains the currently
      existing interfaces on the device.

      If a client tries to create configuration for a
      system-controlled interface that is not present in the
      /interfaces-state/interface list, the server MAY reject
      the request if the implementation does not support
      pre-provisioning of interfaces or if the name refers to
      an interface that can never exist in the system.  A
      NETCONF server MUST reply with an rpc-error with the
      error-tag 'invalid-value' in this case.

      If the device supports pre-provisioning of interface
      configuration, the 'pre-provisioning' feature is
      advertised.

      If the device allows arbitrarily named user-controlled
      interfaces, the 'arbitrary-names' feature is advertised.

      When a configured user-controlled interface is created by
      the system, it is instantiated with the same name in the
      /interface-state/interface list.";
 }

 leaf description {
   type string;
   description
     "A textual description of the interface.

      A server implementation MAY map this leaf to the ifAlias
      MIB object.  Such an implementation needs to use some
      mechanism to handle the differences in size and characters
      allowed between this leaf and ifAlias.  The definition of
      such a mechanism is outside the scope of this document.
```


Copyright© 2022 Ciena® Corporation

```
      Since ifAlias is defined to be stored in non-volatile
      storage, the MIB implementation MUST map ifAlias to the
      value of 'description' in the persistently stored
      datastore.

      Specifically, if the device supports ':startup', when
      ifAlias is read the device MUST return the value of
      'description' in the 'startup' datastore, and when it is
      written, it MUST be written to the 'running' and 'startup'
      datastores.  Note that it is up to the implementation to


      decide whether to modify this single leaf in 'startup' or
      perform an implicit copy-config from 'running' to
      'startup'.

      If the device does not support ':startup', ifAlias MUST
      be mapped to the 'description' leaf in the 'running'
      datastore.";
    reference
      "RFC 2863: The Interfaces Group MIB - ifAlias";
  }

  leaf type {
    type identityref {
      base interface-type;
    }
    mandatory true;
    description
      "The type of the interface.

      When an interface entry is created, a server MAY
      initialize the type leaf with a valid value, e.g., if it
      is possible to derive the type from the name of the
      interface.

      If a client tries to set the type of an interface to a
      value that can never be used by the system, e.g., if the
      type is not supported or if the type does not match the
      name of the interface, the server MUST reject the request.
      A NETCONF server MUST reply with an rpc-error with the
      error-tag 'invalid-value' in this case.";
    reference
      "RFC 2863: The Interfaces Group MIB - ifType";
  }

  leaf enabled {
    type boolean;
    default "true";
    description
      "This leaf contains the configured, desired state of the
      interface.

      Systems that implement the IF-MIB use the value of this
      leaf in the 'running' datastore to set
      IF-MIB.ifAdminStatus to 'up' or 'down' after an ifEntry
      has been initialized, as described in RFC 2863.
```


Copyright© 2022 Ciena® Corporation

```
          Changes in this leaf in the 'running' datastore are
          reflected in ifAdminStatus, but if ifAdminStatus is
          changed over SNMP, this leaf is not affected.";
        reference
          "RFC 2863: The Interfaces Group MIB - ifAdminStatus";
      }

      leaf link-up-down-trap-enable {
        if-feature if-mib;
        type enumeration {
          enum enabled {
            value 1;
          }
          enum disabled {
            value 2;
          }
        }
        description
          "Controls whether linkUp/linkDown SNMP notifications
          should be generated for this interface.

          If this node is not configured, the value 'enabled' is
          operationally used by the server for interfaces that do
          not operate on top of any other interface (i.e., there are
          no 'lower-layer-if' entries), and 'disabled' otherwise.";
        reference
          "RFC 2863: The Interfaces Group MIB -
                     ifLinkUpDownTrapEnable";
      }
    }
  }

  /*
   * Operational state data nodes
   */

  container interfaces-state {
    config false;
    description
      "Data nodes for the operational state of interfaces.";

    list interface {
      key "name";

      description
        "The list of interfaces on the device.

         System-controlled interfaces created by the system are
         always present in this list, whether they are configured or
         not.";

      leaf name {
        type string;
        description
          "The name of the interface.

           A server implementation MAY map this leaf to the ifName
           MIB object.  Such an implementation needs to use some
           mechanism to handle the differences in size and characters
```


Copyright© 2022 Ciena® Corporation

```
      allowed between this leaf and ifName.  The definition of
      such a mechanism is outside the scope of this document.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifName";
}

leaf type {
  type identityref {
    base interface-type;
  }
  mandatory true;
  description
    "The type of the interface.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifType";
}

leaf admin-status {
  if-feature if-mib;
  type enumeration {
    enum up {
      value 1;
      description
        "Ready to pass packets.";
    }
    enum down {
      value 2;
      description
        "Not ready to pass packets and not in some test mode.";
    }

    enum testing {
      value 3;
      description
        "In some test mode.";
    }
  }
  mandatory true;
  description
    "The desired state of the interface.

     This leaf has the same read semantics as ifAdminStatus.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifAdminStatus";
}

leaf oper-status {
  type enumeration {
    enum up {
      value 1;
      description
        "Ready to pass packets.";
    }
    enum down {
      value 2;
      description
        "The interface does not pass any packets.";
    }
    enum testing {
      value 3;
```


Copyright© 2022 Ciena® Corporation

```
        description
          "In some test mode.  No operational packets can
           be passed.";
      }
      enum unknown {
        value 4;
        description
          "Status cannot be determined for some reason.";
      }
      enum dormant {
        value 5;
        description
          "Waiting for some external event.";
      }
      enum not-present {
        value 6;
        description
          "Some component (typically hardware) is missing.";
      }

      enum lower-layer-down {
        value 7;
        description
          "Down due to state of lower-layer interface(s).";
      }
    }
    mandatory true;
    description
      "The current operational state of the interface.

       This leaf has the same semantics as ifOperStatus.";
    reference
      "RFC 2863: The Interfaces Group MIB - ifOperStatus";
  }

  leaf last-change {
    type yang:date-and-time;
    description
      "The time the interface entered its current operational
       state.  If the current state was entered prior to the
       last re-initialization of the local network management
       subsystem, then this node is not present.";
    reference
      "RFC 2863: The Interfaces Group MIB - ifLastChange";
  }

  leaf if-index {
    if-feature if-mib;
    type int32 {
      range "1..2147483647";
    }
    mandatory true;
    description
      "The ifIndex value for the ifEntry represented by this
       interface.";
    reference
      "RFC 2863: The Interfaces Group MIB - ifIndex";
  }

  leaf phys-address {
```


Copyright© 2022 Ciena® Corporation

```
    type yang:phys-address;
    description
      "The interface's address at its protocol sub-layer.  For
       example, for an 802.x interface, this object normally
       contains a Media Access Control (MAC) address.  The
       interface's media-specific modules must define the bit



       and byte ordering and the format of the value of this
       object.  For interfaces that do not have such an address
       (e.g., a serial line), this node is not present.";
    reference
      "RFC 2863: The Interfaces Group MIB - ifPhysAddress";
  }

  leaf-list higher-layer-if {
    type interface-state-ref;
    description
      "A list of references to interfaces layered on top of this
       interface.";
    reference
      "RFC 2863: The Interfaces Group MIB - ifStackTable";
  }

  leaf-list lower-layer-if {
    type interface-state-ref;
    description
      "A list of references to interfaces layered underneath this
       interface.";
    reference
      "RFC 2863: The Interfaces Group MIB - ifStackTable";
  }

  leaf speed {
    type yang:gauge64;
    units "bits/second";
    description
        "An estimate of the interface's current bandwidth in bits
         per second.  For interfaces that do not vary in
         bandwidth or for those where no accurate estimation can
         be made, this node should contain the nominal bandwidth.
         For interfaces that have no concept of bandwidth, this
         node is not present.";
    reference
      "RFC 2863: The Interfaces Group MIB -
                 ifSpeed, ifHighSpeed";
  }

  container statistics {
    description
      "A collection of interface-related statistics objects.";

    leaf discontinuity-time {
      type yang:date-and-time;
      mandatory true;
      description
        "The time on the most recent occasion at which any one or
         more of this interface's counters suffered a
         discontinuity.  If no such discontinuities have occurred
```


Copyright© 2022 Ciena® Corporation

```
      since the last re-initialization of the local management
      subsystem, then this node contains the time the local
      management subsystem re-initialized itself.";
  }

  leaf in-octets {
    type yang:counter64;
    description
      "The total number of octets received on the interface,
       including framing characters.

       Discontinuities in the value of this counter can occur
       at re-initialization of the management system, and at
       other times as indicated by the value of
       'discontinuity-time'.";
    reference
      "RFC 2863: The Interfaces Group MIB - ifHCInOctets";
  }

  leaf in-unicast-pkts {
    type yang:counter64;
    description
      "The number of packets, delivered by this sub-layer to a
       higher (sub-)layer, that were not addressed to a
       multicast or broadcast address at this sub-layer.

       Discontinuities in the value of this counter can occur
       at re-initialization of the management system, and at
       other times as indicated by the value of
       'discontinuity-time'.";
    reference
      "RFC 2863: The Interfaces Group MIB - ifHCInUcastPkts";
  }




  leaf in-broadcast-pkts {
    type yang:counter64;
    description
      "The number of packets, delivered by this sub-layer to a
       higher (sub-)layer, that were addressed to a broadcast
       address at this sub-layer.

       Discontinuities in the value of this counter can occur
       at re-initialization of the management system, and at
       other times as indicated by the value of
       'discontinuity-time'.";
    reference
      "RFC 2863: The Interfaces Group MIB -
                 ifHCInBroadcastPkts";
  }

  leaf in-multicast-pkts {
    type yang:counter64;
    description
      "The number of packets, delivered by this sub-layer to a
       higher (sub-)layer, that were addressed to a multicast
       address at this sub-layer.  For a MAC-layer protocol,
```


Copyright© 2022 Ciena® Corporation

```
        this includes both Group and Functional addresses.

        Discontinuities in the value of this counter can occur
        at re-initialization of the management system, and at
        other times as indicated by the value of
        'discontinuity-time'.";
    reference
      "RFC 2863: The Interfaces Group MIB -
                 ifHCInMulticastPkts";
  }

  leaf in-discards {
    type yang:counter32;
    description
      "The number of inbound packets that were chosen to be
       discarded even though no errors had been detected to
       prevent their being deliverable to a higher-layer
       protocol.  One possible reason for discarding such a
       packet could be to free up buffer space.

       Discontinuities in the value of this counter can occur
       at re-initialization of the management system, and at
       other times as indicated by the value of
       'discontinuity-time'.";



    reference
      "RFC 2863: The Interfaces Group MIB - ifInDiscards";
  }

  leaf in-errors {
    type yang:counter32;
    description
      "For packet-oriented interfaces, the number of inbound
       packets that contained errors preventing them from being
       deliverable to a higher-layer protocol.  For character-
       oriented or fixed-length interfaces, the number of
       inbound transmission units that contained errors
       preventing them from being deliverable to a higher-layer
       protocol.

       Discontinuities in the value of this counter can occur
       at re-initialization of the management system, and at
       other times as indicated by the value of
       'discontinuity-time'.";
    reference
      "RFC 2863: The Interfaces Group MIB - ifInErrors";
  }

  leaf in-unknown-protos {
    type yang:counter32;
    description
      "For packet-oriented interfaces, the number of packets
       received via the interface that were discarded because
       of an unknown or unsupported protocol.  For
       character-oriented or fixed-length interfaces that
       support protocol multiplexing, the number of
       transmission units received via the interface that were
       discarded because of an unknown or unsupported protocol.
```


Copyright© 2022 Ciena® Corporation

```
        For any interface that does not support protocol
        multiplexing, this counter is not present.

        Discontinuities in the value of this counter can occur
        at re-initialization of the management system, and at
        other times as indicated by the value of
        'discontinuity-time'.";
    reference
      "RFC 2863: The Interfaces Group MIB - ifInUnknownProtos";
  }




  leaf out-octets {
    type yang:counter64;
    description
      "The total number of octets transmitted out of the
       interface, including framing characters.

       Discontinuities in the value of this counter can occur
       at re-initialization of the management system, and at
       other times as indicated by the value of
       'discontinuity-time'.";
    reference
      "RFC 2863: The Interfaces Group MIB - ifHCOutOctets";
  }

  leaf out-unicast-pkts {
    type yang:counter64;
    description
      "The total number of packets that higher-level protocols
       requested be transmitted, and that were not addressed
       to a multicast or broadcast address at this sub-layer,
       including those that were discarded or not sent.

       Discontinuities in the value of this counter can occur
       at re-initialization of the management system, and at
       other times as indicated by the value of
       'discontinuity-time'.";
    reference
      "RFC 2863: The Interfaces Group MIB - ifHCOutUcastPkts";
  }

  leaf out-broadcast-pkts {
    type yang:counter64;
    description
      "The total number of packets that higher-level protocols
       requested be transmitted, and that were addressed to a
       broadcast address at this sub-layer, including those
       that were discarded or not sent.

       Discontinuities in the value of this counter can occur
       at re-initialization of the management system, and at
       other times as indicated by the value of
       'discontinuity-time'.";
    reference
      "RFC 2863: The Interfaces Group MIB -
```


Copyright© 2022 Ciena® Corporation

```
                ifHCOutBroadcastPkts";
  }


  leaf out-multicast-pkts {
    type yang:counter64;
    description
      "The total number of packets that higher-level protocols
       requested be transmitted, and that were addressed to a
       multicast address at this sub-layer, including those
       that were discarded or not sent.  For a MAC-layer
       protocol, this includes both Group and Functional
       addresses.

       Discontinuities in the value of this counter can occur
       at re-initialization of the management system, and at
       other times as indicated by the value of
       'discontinuity-time'.";
    reference
      "RFC 2863: The Interfaces Group MIB -
                ifHCOutMulticastPkts";
  }

  leaf out-discards {
    type yang:counter32;
    description
      "The number of outbound packets that were chosen to be
       discarded even though no errors had been detected to
       prevent their being transmitted.  One possible reason
       for discarding such a packet could be to free up buffer
       space.

       Discontinuities in the value of this counter can occur
       at re-initialization of the management system, and at
       other times as indicated by the value of
       'discontinuity-time'.";
    reference
      "RFC 2863: The Interfaces Group MIB - ifOutDiscards";
  }

  leaf out-errors {
    type yang:counter32;
    description
      "For packet-oriented interfaces, the number of outbound
       packets that could not be transmitted because of errors.
       For character-oriented or fixed-length interfaces, the
       number of outbound transmission units that could not be
       transmitted because of errors.




       Discontinuities in the value of this counter can occur
       at re-initialization of the management system, and at
       other times as indicated by the value of
       'discontinuity-time'.";
    reference
      "RFC 2863: The Interfaces Group MIB - ifOutErrors";
```


Copyright© 2022 Ciena® Corporation

```
            }
          }
        }
      }
    }
}
```


Copyright© 2022 Ciena® Corporation

ietf-ip.yang

```
module ietf-ip {

 namespace "urn:ietf:params:xml:ns:yang:ietf-ip";
 prefix ip;

 import ietf-interfaces {
   prefix if;
 }
 import ietf-inet-types {
   prefix inet;
 }
 import ietf-yang-types {
   prefix yang;
 }

 organization
   "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

 contact
   "WG Web:   <http://tools.ietf.org/wg/netmod/>
    WG List:  <mailto:netmod@ietf.org>

    WG Chair: Thomas Nadeau
              <mailto:tnadeau@lucidvision.com>

    WG Chair: Juergen Schoenwaelder
              <mailto:j.schoenwaelder@jacobs-university.de>

    Editor:   Martin Bjorklund
              <mailto:mbj@tail-f.com>";









 description
   "This module contains a collection of YANG definitions for
    configuring IP implementations.

    Copyright (c) 2014 IETF Trust and the persons identified as
    authors of the code.  All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject
    to the license terms contained in, the Simplified BSD License
    set forth in Section 4.c of the IETF Trust's Legal Provisions
    Relating to IETF Documents
    (http://trustee.ietf.org/license-info).

    This version of this YANG module is part of RFC 7277; see
    the RFC itself for full legal notices.";

 revision 2014-06-16 {
```


Copyright© 2022 Ciena® Corporation

```
  description
    "Initial revision.";
  reference
    "RFC 7277: A YANG Data Model for IP Management";
}

/*

 * Features
 */

feature ipv4-non-contiguous-netmasks {
  description
    "Indicates support for configuring non-contiguous
     subnet masks.";
}

feature ipv6-privacy-autoconf {
  description
    "Indicates support for Privacy Extensions for Stateless Address
     Autoconfiguration in IPv6.";
  reference
    "RFC 4941: Privacy Extensions for Stateless Address
                Autoconfiguration in IPv6";
}




/*
 * Typedefs
 */

typedef ip-address-origin {
  type enumeration {
    enum other {
      description
        "None of the following.";
    }
    enum static {
      description
        "Indicates that the address has been statically
         configured - for example, using NETCONF or a Command Line
         Interface.";
    }
    enum dhcp {
      description
        "Indicates an address that has been assigned to this
         system by a DHCP server.";
    }
    enum link-layer {
      description
        "Indicates an address created by IPv6 stateless
         autoconfiguration that embeds a link-layer address in its
         interface identifier.";
    }
    enum random {
      description
        "Indicates an address chosen by the system at
```


Copyright© 2022 Ciena® Corporation

```
          random, e.g., an IPv4 address within 169.254/16, an
          RFC 4941 temporary address, or an RFC 7217 semantically
          opaque address.";
      reference
        "RFC 4941: Privacy Extensions for Stateless Address
                   Autoconfiguration in IPv6
         RFC 7217: A Method for Generating Semantically Opaque
                   Interface Identifiers with IPv6 Stateless
                   Address Autoconfiguration (SLAAC)";
    }
  }
  description
    "The origin of an address.";
}



typedef neighbor-origin {
  type enumeration {
    enum other {
      description
        "None of the following.";
    }
    enum static {
      description
        "Indicates that the mapping has been statically
         configured - for example, using NETCONF or a Command Line
         Interface.";
    }
    enum dynamic {
      description
        "Indicates that the mapping has been dynamically resolved
         using, e.g., IPv4 ARP or the IPv6 Neighbor Discovery
         protocol.";
    }
  }
  description
    "The origin of a neighbor entry.";
}

/*
 * Configuration data nodes
 */

augment "/if:interfaces/if:interface" {
  description
    "Parameters for configuring IP on interfaces.

     If an interface is not capable of running IP, the server
     must not allow the client to configure these parameters.";

  container ipv4 {
    presence
      "Enables IPv4 unless the 'enabled' leaf
       (which defaults to 'true') is set to 'false'";
    description
      "Parameters for the IPv4 address family.";
```


Copyright© 2022 Ciena® Corporation

```
leaf enabled {
  type boolean;
  default true;
  description
    "Controls whether IPv4 is enabled or disabled on this
     interface.  When IPv4 is enabled, this interface is
     connected to an IPv4 stack, and the interface can send
     and receive IPv4 packets.";
}
leaf forwarding {
  type boolean;
  default false;
  description
    "Controls IPv4 packet forwarding of datagrams received by,
     but not addressed to, this interface.  IPv4 routers
     forward datagrams.  IPv4 hosts do not (except those
     source-routed via the host).";
}
leaf mtu {
  type uint16 {
    range "68..max";
  }
  units octets;
  description
    "The size, in octets, of the largest IPv4 packet that the
     interface will send and receive.

     The server may restrict the allowed values for this leaf,
     depending on the interface's type.

     If this leaf is not configured, the operationally used MTU
     depends on the interface's type.";
  reference
    "RFC 791: Internet Protocol";
}
list address {
  key "ip";
  description
    "The list of configured IPv4 addresses on the interface.";

  leaf ip {
    type inet:ipv4-address-no-zone;
    description
      "The IPv4 address on the interface.";
  }



  choice subnet {
    mandatory true;
    description
      "The subnet can be specified as a prefix-length, or,
       if the server supports non-contiguous netmasks, as
       a netmask.";
```


Copyright© 2022 Ciena® Corporation

```
        leaf prefix-length {
          type uint8 {
            range "0..32";
          }
          description
            "The length of the subnet prefix.";
        }
        leaf netmask {
          if-feature ipv4-non-contiguous-netmasks;
          type yang:dotted-quad;
          description
            "The subnet specified as a netmask.";
        }
      }
    }
    list neighbor {
      key "ip";
      description
        "A list of mappings from IPv4 addresses to
         link-layer addresses.

         Entries in this list are used as static entries in the
         ARP Cache.";
      reference
        "RFC 826: An Ethernet Address Resolution Protocol";

      leaf ip {
        type inet:ipv4-address-no-zone;
        description
          "The IPv4 address of the neighbor node.";
      }
      leaf link-layer-address {
        type yang:phys-address;
        mandatory true;
        description
          "The link-layer address of the neighbor node.";
      }
    }

  }


  container ipv6 {
    presence
      "Enables IPv6 unless the 'enabled' leaf
       (which defaults to 'true') is set to 'false'";
    description
      "Parameters for the IPv6 address family.";

    leaf enabled {
      type boolean;
      default true;
      description
        "Controls whether IPv6 is enabled or disabled on this
         interface.  When IPv6 is enabled, this interface is
         connected to an IPv6 stack, and the interface can send
         and receive IPv6 packets.";
    }
    leaf forwarding {
      type boolean;
```


Copyright© 2022 Ciena® Corporation

```
    default false;
    description
      "Controls IPv6 packet forwarding of datagrams received by,
       but not addressed to, this interface.  IPv6 routers
       forward datagrams.  IPv6 hosts do not (except those
       source-routed via the host).";
    reference
      "RFC 4861: Neighbor Discovery for IP version 6 (IPv6)
                 Section 6.2.1, IsRouter";
  }
  leaf mtu {
    type uint32 {
      range "1280..max";
    }
    units octets;
    description
      "The size, in octets, of the largest IPv6 packet that the
       interface will send and receive.

       The server may restrict the allowed values for this leaf,
       depending on the interface's type.

       If this leaf is not configured, the operationally used MTU
       depends on the interface's type.";
    reference
      "RFC 2460: Internet Protocol, Version 6 (IPv6) Specification
                 Section 5";
  }


  list address {
    key "ip";
    description
      "The list of configured IPv6 addresses on the interface.";

    leaf ip {
      type inet:ipv6-address-no-zone;
      description
        "The IPv6 address on the interface.";
    }
    leaf prefix-length {
      type uint8 {
        range "0..128";
      }
      mandatory true;
      description
        "The length of the subnet prefix.";
    }
  }
  list neighbor {
    key "ip";
    description
      "A list of mappings from IPv6 addresses to
       link-layer addresses.

       Entries in this list are used as static entries in the
       Neighbor Cache.";
    reference
      "RFC 4861: Neighbor Discovery for IP version 6 (IPv6)";
```


Copyright© 2022 Ciena® Corporation

```
    leaf ip {
      type inet:ipv6-address-no-zone;
      description
        "The IPv6 address of the neighbor node.";
    }
    leaf link-layer-address {
      type yang:phys-address;
      mandatory true;
      description
        "The link-layer address of the neighbor node.";
    }
  }




  leaf dup-addr-detect-transmits {
    type uint32;
    default 1;
    description
      "The number of consecutive Neighbor Solicitation messages
       sent while performing Duplicate Address Detection on a
       tentative address.  A value of zero indicates that
       Duplicate Address Detection is not performed on
       tentative addresses.  A value of one indicates a single
       transmission with no follow-up retransmissions.";
    reference
      "RFC 4862: IPv6 Stateless Address Autoconfiguration";
  }
  container autoconf {
    description
      "Parameters to control the autoconfiguration of IPv6
       addresses, as described in RFC 4862.";
    reference
      "RFC 4862: IPv6 Stateless Address Autoconfiguration";

    leaf create-global-addresses {
      type boolean;
      default true;
      description
        "If enabled, the host creates global addresses as
         described in RFC 4862.";
      reference
        "RFC 4862: IPv6 Stateless Address Autoconfiguration
                   Section 5.5";
    }
    leaf create-temporary-addresses {
      if-feature ipv6-privacy-autoconf;
      type boolean;
      default false;
      description
        "If enabled, the host creates temporary addresses as
         described in RFC 4941.";
      reference
        "RFC 4941: Privacy Extensions for Stateless Address
                   Autoconfiguration in IPv6";
    }
```


Copyright© 2022 Ciena® Corporation

```
      leaf temporary-valid-lifetime {
        if-feature ipv6-privacy-autoconf;
        type uint32;
        units "seconds";
        default 604800;
        description
          "The time period during which the temporary address
           is valid.";
        reference
          "RFC 4941: Privacy Extensions for Stateless Address
                     Autoconfiguration in IPv6
                     - TEMP_VALID_LIFETIME";
      }
      leaf temporary-preferred-lifetime {
        if-feature ipv6-privacy-autoconf;
        type uint32;
        units "seconds";
        default 86400;
        description
          "The time period during which the temporary address is
           preferred.";
        reference
          "RFC 4941: Privacy Extensions for Stateless Address
                     Autoconfiguration in IPv6
                     - TEMP_PREFERRED_LIFETIME";
      }
    }
  }
}

/*
 * Operational state data nodes
 */

augment "/if:interfaces-state/if:interface" {
  description
    "Data nodes for the operational state of IP on interfaces.";

  container ipv4 {
    presence "Present if IPv4 is enabled on this interface";
    config false;
    description
      "Interface-specific parameters for the IPv4 address family.";




    leaf forwarding {
      type boolean;
      description
        "Indicates whether IPv4 packet forwarding is enabled or
         disabled on this interface.";
    }
```


Copyright© 2022 Ciena® Corporation

```
leaf mtu {
  type uint16 {
    range "68..max";
  }
  units octets;
  description
    "The size, in octets, of the largest IPv4 packet that the
     interface will send and receive.";
  reference
    "RFC 791: Internet Protocol";
}
list address {
  key "ip";
  description
    "The list of IPv4 addresses on the interface.";

  leaf ip {
    type inet:ipv4-address-no-zone;
    description
      "The IPv4 address on the interface.";
  }
  choice subnet {
    description
      "The subnet can be specified as a prefix-length, or,
       if the server supports non-contiguous netmasks, as
       a netmask.";
    leaf prefix-length {
      type uint8 {
        range "0..32";
      }
      description
        "The length of the subnet prefix.";
    }
    leaf netmask {
      if-feature ipv4-non-contiguous-netmasks;
      type yang:dotted-quad;
      description
        "The subnet specified as a netmask.";
    }
  }


  leaf origin {
    type ip-address-origin;
    description
      "The origin of this address.";
  }
}
list neighbor {
  key "ip";
  description
    "A list of mappings from IPv4 addresses to
     link-layer addresses.

     This list represents the ARP Cache.";
  reference
    "RFC 826: An Ethernet Address Resolution Protocol";

  leaf ip {
    type inet:ipv4-address-no-zone;
```


Copyright© 2022 Ciena® Corporation

```
        description
          "The IPv4 address of the neighbor node.";
      }
      leaf link-layer-address {
        type yang:phys-address;
        description
          "The link-layer address of the neighbor node.";
      }
      leaf origin {
        type neighbor-origin;
        description
          "The origin of this neighbor entry.";
      }
    }

  }

  container ipv6 {
    presence "Present if IPv6 is enabled on this interface";
    config false;
    description
      "Parameters for the IPv6 address family.";









    leaf forwarding {
      type boolean;
      default false;
      description
        "Indicates whether IPv6 packet forwarding is enabled or
         disabled on this interface.";
      reference
        "RFC 4861: Neighbor Discovery for IP version 6 (IPv6)
                   Section 6.2.1, IsRouter";
    }
    leaf mtu {
      type uint32 {
        range "1280..max";
      }
      units octets;
      description
        "The size, in octets, of the largest IPv6 packet that the
         interface will send and receive.";
      reference
        "RFC 2460: Internet Protocol, Version 6 (IPv6) Specification
                   Section 5";
    }
    list address {
      key "ip";
      description
        "The list of IPv6 addresses on the interface.";

      leaf ip {
        type inet:ipv6-address-no-zone;
        description
```


Copyright© 2022 Ciena® Corporation

```
      "The IPv6 address on the interface.";
  }
  leaf prefix-length {
    type uint8 {
      range "0..128";
    }
    mandatory true;
    description
      "The length of the subnet prefix.";
  }
  leaf origin {
    type ip-address-origin;
    description
      "The origin of this address.";
  }



  leaf status {
    type enumeration {
      enum preferred {
        description
          "This is a valid address that can appear as the
           destination or source address of a packet.";
      }
      enum deprecated {
        description
          "This is a valid but deprecated address that should
           no longer be used as a source address in new
           communications, but packets addressed to such an
           address are processed as expected.";
      }
      enum invalid {
        description
          "This isn't a valid address, and it shouldn't appear
           as the destination or source address of a packet.";
      }
      enum inaccessible {
        description
          "The address is not accessible because the interface
           to which this address is assigned is not
           operational.";
      }
      enum unknown {
        description
          "The status cannot be determined for some reason.";
      }
      enum tentative {
        description
          "The uniqueness of the address on the link is being
           verified.  Addresses in this state should not be
           used for general communication and should only be
           used to determine the uniqueness of the address.";
      }
      enum duplicate {
        description
          "The address has been determined to be non-unique on
           the link and so must not be used.";
      }
```


Copyright© 2022 Ciena® Corporation

```
        enum optimistic {
          description
            "The address is available for use, subject to
             restrictions, while its uniqueness on a link is
             being verified.";
        }
      }
      description
        "The status of an address.  Most of the states correspond
         to states from the IPv6 Stateless Address
         Autoconfiguration protocol.";
      reference
        "RFC 4293: Management Information Base for the
                   Internet Protocol (IP)
                   - IpAddressStatusTC
         RFC 4862: IPv6 Stateless Address Autoconfiguration";
    }
  }
  list neighbor {
    key "ip";
    description
      "A list of mappings from IPv6 addresses to
       link-layer addresses.

       This list represents the Neighbor Cache.";
    reference
      "RFC 4861: Neighbor Discovery for IP version 6 (IPv6)";

    leaf ip {
      type inet:ipv6-address-no-zone;
      description
        "The IPv6 address of the neighbor node.";
    }
    leaf link-layer-address {
      type yang:phys-address;
      description
        "The link-layer address of the neighbor node.";
    }
    leaf origin {
      type neighbor-origin;
      description
        "The origin of this neighbor entry.";
    }
    leaf is-router {
      type empty;
      description
        "Indicates that the neighbor node acts as a router.";
    }
    leaf state {
      type enumeration {
        enum incomplete {
          description
            "Address resolution is in progress, and the link-layer
             address of the neighbor has not yet been
```


Copyright© 2022 Ciena® Corporation

```
                determined.";
            }
            enum reachable {
              description
                "Roughly speaking, the neighbor is known to have been
                 reachable recently (within tens of seconds ago).";
            }
            enum stale {
              description
                "The neighbor is no longer known to be reachable, but
                 until traffic is sent to the neighbor no attempt
                 should be made to verify its reachability.";
            }
            enum delay {
              description
                "The neighbor is no longer known to be reachable, and
                 traffic has recently been sent to the neighbor.
                 Rather than probe the neighbor immediately, however,
                 delay sending probes for a short while in order to
                 give upper-layer protocols a chance to provide
                 reachability confirmation.";
            }
            enum probe {
              description
                "The neighbor is no longer known to be reachable, and
                 unicast Neighbor Solicitation probes are being sent
                 to verify reachability.";
            }
          }
          description
            "The Neighbor Unreachability Detection state of this
             entry.";
          reference
            "RFC 4861: Neighbor Discovery for IP version 6 (IPv6)
                       Section 7.3.2";
        }
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ietf-netconf-acm@2012-02-22.yang

```
module ietf-netconf-acm {

  namespace "urn:ietf:params:xml:ns:yang:ietf-netconf-acm";

  prefix "nacm";

  import ietf-yang-types {
    prefix yang;
  }

  organization
    "IETF NETCONF (Network Configuration) Working Group";

  contact
    "WG Web:   <http://tools.ietf.org/wg/netconf/>
     WG List:  <mailto:netconf@ietf.org>

     WG Chair: Mehmet Ersue
               <mailto:mehmet.ersue@nsn.com>

     WG Chair: Bert Wijnen
               <mailto:bertietf@bwijnen.net>

     Editor:   Andy Bierman
               <mailto:andy@yumaworks.com>

     Editor:   Martin Bjorklund
               <mailto:mbj@tail-f.com>";

  description
    "NETCONF Access Control Model.

     Copyright (c) 2012 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject
     to the license terms contained in, the Simplified BSD
     License set forth in Section 4.c of the IETF Trust's
     Legal Provisions Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC 6536; see
     the RFC itself for full legal notices.";

  revision "2012-02-22" {
    description
      "Initial version";
    reference
      "RFC 6536: Network Configuration Protocol (NETCONF)
                 Access Control Model";
  }

  /*
   * Extension statements
   */

  extension default-deny-write {
```


Copyright© 2022 Ciena® Corporation

```
    description
      "Used to indicate that the data model node
       represents a sensitive security system parameter.

       If present, and the NACM module is enabled (i.e.,
       /nacm/enable-nacm object equals 'true'), the NETCONF server
       will only allow the designated 'recovery session' to have
       write access to the node.  An explicit access control rule is
       required for all other users.

       The 'default-deny-write' extension MAY appear within a data
       definition statement.  It is ignored otherwise.";
  }

  extension default-deny-all {
    description
      "Used to indicate that the data model node
       controls a very sensitive security system parameter.

       If present, and the NACM module is enabled (i.e.,
       /nacm/enable-nacm object equals 'true'), the NETCONF server
       will only allow the designated 'recovery session' to have
       read, write, or execute access to the node.  An explicit
       access control rule is required for all other users.

       The 'default-deny-all' extension MAY appear within a data
       definition statement, 'rpc' statement, or 'notification'
       statement.  It is ignored otherwise.";
  }

  /*
   * Derived types
   */

  typedef user-name-type {
    type string {
      length "1..max";
    }
    description
      "General Purpose Username string.";
  }

  typedef matchall-string-type {
    type string {
      pattern "\*";
    }
    description
      "The string containing a single asterisk '*' is used
       to conceptually represent all possible values
       for the particular leaf using this data type.";
  }

  typedef access-operations-type {
    type bits {
      bit create {
        description
          "Any protocol operation that creates a
           new data node.";
      }
      bit read {
```


Copyright© 2022 Ciena® Corporation

```
      description
        "Any protocol operation or notification that
         returns the value of a data node.";
    }
    bit update {
      description
        "Any protocol operation that alters an existing
         data node.";
    }
    bit delete {
      description
        "Any protocol operation that removes a data node.";
    }
    bit exec {
      description
        "Execution access to the specified protocol operation.";
    }
  }
  description
    "NETCONF Access Operation.";
}

typedef group-name-type {
  type string {
    length "1..max";
    pattern "[^\*].*";
  }
  description
    "Name of administrative group to which
     users can be assigned.";
}

typedef action-type {
  type enumeration {
    enum permit {
      description
        "Requested action is permitted.";
    }
    enum deny {
      description
        "Requested action is denied.";
    }
  }
  description
    "Action taken by the server when a particular
     rule matches.";
}

typedef node-instance-identifier {
  type yang:xpath1.0;
  description
    "Path expression used to represent a special
     data node instance identifier string.

     A node-instance-identifier value is an
     unrestricted YANG instance-identifier expression.
     All the same rules as an instance-identifier apply
     except predicates for keys are optional.  If a key
     predicate is missing, then the node-instance-identifier
     represents all possible server instances for that key.
```


Copyright© 2022 Ciena® Corporation

```
      This XPath expression is evaluated in the following context:

       o  The set of namespace declarations are those in scope on
          the leaf element where this type is used.

       o  The set of variable bindings contains one variable,
          'USER', which contains the name of the user of the current
           session.

       o  The function library is the core function library, but
          note that due to the syntax restrictions of an
          instance-identifier, no functions are allowed.

       o  The context node is the root node in the data tree.";
}

/*
 * Data definition statements
 */

container nacm {
  nacm:default-deny-all;

  description
    "Parameters for NETCONF Access Control Model.";

  leaf enable-nacm {
    type boolean;
    default true;
    description
      "Enables or disables all NETCONF access control
       enforcement.  If 'true', then enforcement
       is enabled.  If 'false', then enforcement
       is disabled.";
  }

  leaf read-default {
    type action-type;
    default "permit";
    description
      "Controls whether read access is granted if
       no appropriate rule is found for a
       particular read request.";
  }

  leaf write-default {
    type action-type;
    default "deny";
    description
      "Controls whether create, update, or delete access
       is granted if no appropriate rule is found for a
       particular write request.";
  }

  leaf exec-default {
    type action-type;
    default "permit";
    description
      "Controls whether exec access is granted if no appropriate
```


Copyright© 2022 Ciena® Corporation

```
      rule is found for a particular protocol operation request.";
  }

  leaf enable-external-groups {
    type boolean;
    default true;
    description
      "Controls whether the server uses the groups reported by the
       NETCONF transport layer when it assigns the user to a set of
       NACM groups.  If this leaf has the value 'false', any group
       names reported by the transport layer are ignored by the
       server.";
  }

  leaf denied-operations {
    type yang:zero-based-counter32;
    config false;
    mandatory true;
    description
      "Number of times since the server last restarted that a
       protocol operation request was denied.";
  }

  leaf denied-data-writes {
    type yang:zero-based-counter32;
    config false;
    mandatory true;
    description
      "Number of times since the server last restarted that a
       protocol operation request to alter
       a configuration datastore was denied.";
  }

  leaf denied-notifications {
    type yang:zero-based-counter32;
    config false;
    mandatory true;
    description
      "Number of times since the server last restarted that
       a notification was dropped for a subscription because
       access to the event type was denied.";
  }

  container groups {
    description
      "NETCONF Access Control Groups.";

    list group {
      key name;

      description
        "One NACM Group Entry.  This list will only contain
         configured entries, not any entries learned from
         any transport protocols.";

      leaf name {
        type group-name-type;
        description
          "Group name associated with this entry.";
      }
```


Copyright© 2022 Ciena® Corporation

```
      leaf-list user-name {
        type user-name-type;
        description
          "Each entry identifies the username of
           a member of the group associated with
           this entry.";
      }
    }
  }

  list rule-list {
    key "name";
    ordered-by user;
    description
      "An ordered collection of access control rules.";

    leaf name {
      type string {
        length "1..max";
      }
      description
        "Arbitrary name assigned to the rule-list.";
    }
    leaf-list group {
      type union {
        type matchall-string-type;
        type group-name-type;
      }
      description
        "List of administrative groups that will be
         assigned the associated access rights
         defined by the 'rule' list.

         The string '*' indicates that all groups apply to the
         entry.";
    }

    list rule {
      key "name";
      ordered-by user;
      description
        "One access control rule.

         Rules are processed in user-defined order until a match is
         found.  A rule matches if 'module-name', 'rule-type', and
         'access-operations' match the request.  If a rule
         matches, the 'action' leaf determines if access is granted
         or not.";

      leaf name {
        type string {
          length "1..max";
        }
        description
          "Arbitrary name assigned to the rule.";
      }

      leaf module-name {
        type union {
```


Copyright© 2022 Ciena® Corporation

```
      type matchall-string-type;
      type string;
    }
    }
    default "*";
    description
      "Name of the module associated with this rule.

       This leaf matches if it has the value '*' or if the
       object being accessed is defined in the module with the
       specified module name.";
  }
  choice rule-type {
    description
      "This choice matches if all leafs present in the rule
       match the request.  If no leafs are present, the
       choice matches all requests.";
    case protocol-operation {
      leaf rpc-name {
        type union {
          type matchall-string-type;
          type string;
        }
        description
          "This leaf matches if it has the value '*' or if
           its value equals the requested protocol operation
           name.";
      }
    }
    case notification {
      leaf notification-name {
        type union {
          type matchall-string-type;
          type string;
        }
        description
          "This leaf matches if it has the value '*' or if its
           value equals the requested notification name.";
      }
    }
    case data-node {
      leaf path {
        type node-instance-identifier;
        mandatory true;
        description
          "Data Node Instance Identifier associated with the
           data node controlled by this rule.

           Configuration data or state data instance
           identifiers start with a top-level data node.  A
           complete instance identifier is required for this
           type of path value.

           The special value '/' refers to all possible
           datastore contents.";
      }
    }
  }

  leaf access-operations {
    type union {
```


Copyright© 2022 Ciena® Corporation

```
            type matchall-string-type;
            type access-operations-type;
          }
          }
          default "*";
          description
            "Access operations associated with this rule.

             This leaf matches if it has the value '*' or if the
             bit corresponding to the requested operation is set.";
        }

        leaf action {
          type action-type;
          mandatory true;
          description
            "The access control action associated with the
             rule.  If a rule is determined to match a
             particular request, then this object is used
             to determine whether to permit or deny the
             request.";
        }

        leaf comment {
          type string;
          description
            "A textual description of the access rule.";
        }
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ietf-netconf-monitoring.yang

```
module ietf-netconf-monitoring {

  namespace "urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring";
  prefix "ncm";

  import ietf-yang-types { prefix yang; }
  import ietf-inet-types { prefix inet; }

  organization
    "IETF NETCONF (Network Configuration) Working Group";

  contact
    "WG Web:   <http://tools.ietf.org/wg/netconf/>
     WG List:  <mailto:netconf@ietf.org>

     WG Chair: Mehmet Ersue
               <mailto:mehmet.ersue@nsn.com>

     WG Chair: Bert Wijnen
               <mailto:bertietf@bwijnen.net>

     Editor:   Mark Scott
               <mailto:mark.scott@ericsson.com>

     Editor:   Martin Bjorklund
               <mailto:mbj@tail-f.com>";

  description
    "NETCONF Monitoring Module.
     All elements in this module are read-only.

     Copyright (c) 2010 IETF Trust and the persons identified as
     authors of the code. All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject
     to the license terms contained in, the Simplified BSD
     License set forth in Section 4.c of the IETF Trust's
     Legal Provisions Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC 6022; see
     the RFC itself for full legal notices.";

  revision 2010-10-04 {
    description
      "Initial revision.";
    reference
      "RFC 6022: YANG Module for NETCONF Monitoring";
  }

  typedef netconf-datastore-type {
    type enumeration {
      enum running;
      enum candidate;
      enum startup;
    }
    description
```


Copyright© 2022 Ciena® Corporation

```
      "Enumeration of possible NETCONF datastore types.";
    reference
      "RFC 4741: NETCONF Configuration Protocol";
  }

  identity transport {
    description
      "Base identity for NETCONF transport types.";
  }

  identity netconf-ssh {
    base transport;
    description
      "NETCONF over Secure Shell (SSH).";
    reference
      "RFC 4742: Using the NETCONF Configuration Protocol
                 over Secure SHell (SSH)";
  }

  identity netconf-soap-over-beep {
    base transport;
    description
      "NETCONF over Simple Object Access Protocol (SOAP) over
       Blocks Extensible Exchange Protocol (BEEP).";
    reference
      "RFC 4743: Using NETCONF over the Simple Object
                 Access Protocol (SOAP)";
  }

  identity netconf-soap-over-https {
    base transport;
    description
      "NETCONF over Simple Object Access Protocol (SOAP)
      over Hypertext Transfer Protocol Secure (HTTPS).";
    reference
      "RFC 4743: Using NETCONF over the Simple Object
                 Access Protocol (SOAP)";
  }

  identity netconf-beep {
    base transport;
    description
      "NETCONF over Blocks Extensible Exchange Protocol (BEEP).";
    reference
      "RFC 4744: Using the NETCONF Protocol over the
                 Blocks Extensible Exchange Protocol (BEEP)";
  }

  identity netconf-tls {
    base transport;
    description
      "NETCONF over Transport Layer Security (TLS).";
    reference
      "RFC 5539: NETCONF over Transport Layer Security (TLS)";
  }

  identity schema-format {
    description
      "Base identity for data model schema languages.";
  }
```


Copyright© 2022 Ciena® Corporation

```
identity xsd {
  base schema-format;
  description
    "W3C XML Schema Definition.";
  reference
    "W3C REC REC-xmlschema-1-20041028:
       XML Schema Part 1: Structures";
}

identity yang {
  base schema-format;
  description
    "The YANG data modeling language for NETCONF.";
  reference
    "RFC 6020:  YANG - A Data Modeling Language for the
                Network Configuration Protocol (NETCONF)";
}

identity yin {
  base schema-format;
  description
    "The YIN syntax for YANG.";
  reference
    "RFC 6020:  YANG - A Data Modeling Language for the
                Network Configuration Protocol (NETCONF)";
}

identity rng {
  base schema-format;
  description
    "Regular Language for XML Next Generation (RELAX NG).";
  reference
    "ISO/IEC 19757-2:2008: RELAX NG";
}

identity rnc {
  base schema-format;
  description
    "Relax NG Compact Syntax";
  reference
    "ISO/IEC 19757-2:2008: RELAX NG";
}

grouping common-counters {
  description
    "Counters that exist both per session, and also globally,
     accumulated from all sessions.";

  leaf in-rpcs {
    type yang:zero-based-counter32;
    description
      "Number of correct <rpc> messages received.";
  }
  leaf in-bad-rpcs {
    type yang:zero-based-counter32;
    description
      "Number of messages received when an <rpc> message was expected,
       that were not correct <rpc> messages.  This includes XML parse
       errors and errors on the rpc layer.";
```


Copyright© 2022 Ciena® Corporation

```
    }
    leaf out-rpc-errors {
      type yang:zero-based-counter32;
      description
        "Number of <rpc-reply> messages sent that contained an
         <rpc-error> element.";
    }
    leaf out-notifications {
      type yang:zero-based-counter32;
      description
        "Number of <notification> messages sent.";
    }
  }

  container netconf-state {
    config false;
    description
      "The netconf-state container is the root of the monitoring
       data model.";

    container capabilities {
      description
        "Contains the list of NETCONF capabilities supported by the
         server.";

      leaf-list capability {
        type inet:uri;
        description
          "List of NETCONF capabilities supported by the server.";
      }
    }

    container datastores {
      description
        "Contains the list of NETCONF configuration datastores.";

      list datastore {
        key name;
        description
          "List of NETCONF configuration datastores supported by
           the NETCONF server and related information.";

        leaf name {
          type netconf-datastore-type;
          description
            "Name of the datastore associated with this list entry.";
        }
        container locks {
          presence
            "This container is present only if the datastore
             is locked.";
          description
            "The NETCONF <lock> and <partial-lock> operations allow
             a client to lock specific resources in a datastore.  The
             NETCONF server will prevent changes to the locked
             resources by all sessions except the one that acquired
             the lock(s).

             Monitoring information is provided for each datastore
             entry including details such as the session that acquired
```


Copyright© 2022 Ciena® Corporation

```
    the lock, the type of lock (global or partial) and the
    list of locked resources.  Multiple locks per datastore
    are supported.";

  grouping lock-info {
    description
      "Lock related parameters, common to both global and
       partial locks.";

    leaf locked-by-session {
      type uint32;
      mandatory true;
      description
        "The session ID of the session that has locked
         this resource.  Both a global lock and a partial
         lock MUST contain the NETCONF session-id.

         If the lock is held by a session that is not managed
         by the NETCONF server (e.g., a CLI session), a session
         id of 0 (zero) is reported.";
      reference
        "RFC 4741: NETCONF Configuration Protocol";
    }
    leaf locked-time {
      type yang:date-and-time;
      mandatory true;
      description
        "The date and time of when the resource was
         locked.";
    }
  }

  choice lock-type {
    description
      "Indicates if a global lock or a set of partial locks
       are set.";

    container global-lock {
      description
        "Present if the global lock is set.";
      uses lock-info;
    }

    list partial-lock {
      key lock-id;
      description
        "List of partial locks.";
      reference
        "RFC 5717: Partial Lock Remote Procedure Call (RPC) for
                   NETCONF";

      leaf lock-id {
        type uint32;
        description
          "This is the lock id returned in the <partial-lock>
           response.";
      }
      uses lock-info;
      leaf-list select {
        type yang:xpath1.0;
```


Copyright© 2022 Ciena® Corporation

```
              min-elements 1;
              description
                "The xpath expression that was used to request
                 the lock.  The select expression indicates the
                 original intended scope of the lock.";
            }
            leaf-list locked-node {
              type instance-identifier;
              description
                "The list of instance-identifiers (i.e., the
                 locked nodes).

                 The scope of the partial lock is defined by the list
                 of locked nodes.";
            }
          }
        }
      }
    }
  }

  container schemas {
    description
      "Contains the list of data model schemas supported by the
       server.";

    list schema {
      key "identifier version format";

      description
        "List of data model schemas supported by the server.";

      leaf identifier {
        type string;
        description
          "Identifier to uniquely reference the schema.  The
           identifier is used in the <get-schema> operation and may
           be used for other purposes such as file retrieval.

           For modeling languages that support or require a data
           model name (e.g., YANG module name) the identifier MUST
           match that name.  For YANG data models, the identifier is
           the name of the module or submodule.  In other cases, an
           identifier such as a filename MAY be used instead.";
      }
      leaf version {
        type string;
        description
          "Version of the schema supported.  Multiple versions MAY be
           supported simultaneously by a NETCONF server.  Each
           version MUST be reported individually in the schema list,
           i.e., with same identifier, possibly different location,
           but different version.

           For YANG data models, version is the value of the most
           recent YANG 'revision' statement in the module or
           submodule, or the empty string if no 'revision' statement
           is present.";
      }
      leaf format {
```


Copyright© 2022 Ciena® Corporation

```
        type identityref {
          base schema-format;
        }
        description
          "The data modeling language the schema is written
           in (currently xsd, yang, yin, rng, or rnc).

           For YANG data models, 'yang' format MUST be supported and
           'yin' format MAY also be provided.";
      }
      leaf namespace {
        type inet:uri;
        mandatory true;
        description
          "The XML namespace defined by the data model.

           For YANG data models, this is the module's namespace.
           If the list entry describes a submodule, this field
           contains the namespace of the module to which the
           submodule belongs.";
      }
      leaf-list location {
        type union {
          type enumeration {
            enum "NETCONF";
          }
          type inet:uri;
        }
        description
          "One or more locations from which the schema can be
           retrieved.  This list SHOULD contain at least one
           entry per schema.

           A schema entry may be located on a remote file system
           (e.g., reference to file system for ftp retrieval) or
           retrieved directly from a server supporting the
           <get-schema> operation (denoted by the value 'NETCONF').";
      }
    }
  }

  container sessions {
    description
      "The sessions container includes session-specific data for
       NETCONF management sessions.  The session list MUST include
       all currently active NETCONF sessions.";

    list session {
      key session-id;
      description
        "All NETCONF sessions managed by the NETCONF server
         MUST be reported in this list.";

      leaf session-id {
        type uint32 {
          range "1..max";
        }
        description
          "Unique identifier for the session.  This value is the
           NETCONF session identifier, as defined in RFC 4741.";
```


Copyright© 2022 Ciena® Corporation

```
        reference
          "RFC 4741: NETCONF Configuration Protocol";
      }
      leaf transport {
        type identityref {
          base transport;
        }
        mandatory true;
        description
          "Identifies the transport for each session, e.g.,
          'netconf-ssh', 'netconf-soap', etc.";
      }
      leaf username  {
        type string;
        mandatory true;
        description
          "The username is the client identity that was authenticated
          by the NETCONF transport protocol.  The algorithm used to
          derive the username is NETCONF transport protocol specific
          and in addition specific to the authentication mechanism
          used by the NETCONF transport protocol.";
      }
      leaf source-host {
        type inet:host;
        description
          "Host identifier of the NETCONF client.  The value
           returned is implementation specific (e.g., hostname,
           IPv4 address, IPv6 address)";
      }
      leaf login-time {
        type yang:date-and-time;
        mandatory true;
        description
          "Time at the server at which the session was established.";
      }
      uses common-counters {
        description
          "Per-session counters.  Zero based with following reset
           behaviour:
             - at start of a session
             - when max value is reached";
      }
    }
  }

  container statistics {
    description
      "Statistical data pertaining to the NETCONF server.";

    leaf netconf-start-time {
      type yang:date-and-time;
      description
        "Date and time at which the management subsystem was
         started.";
    }
    leaf in-bad-hellos {
      type yang:zero-based-counter32;
      description
        "Number of sessions silently dropped because an
        invalid <hello> message was received.  This includes <hello>
```


Copyright© 2022 Ciena® Corporation

```
          messages with a 'session-id' attribute, bad namespace, and
          bad capability declarations.";
      }
      leaf in-sessions {
        type yang:zero-based-counter32;
        description
          "Number of sessions started.  This counter is incremented
           when a <hello> message with a <session-id> is sent.

          'in-sessions' - 'in-bad-hellos' =
              'number of correctly started netconf sessions'";
      }
      leaf dropped-sessions {
        type yang:zero-based-counter32;
        description
          "Number of sessions that were abnormally terminated, e.g.,
           due to idle timeout or transport close.  This counter is not
           incremented when a session is properly closed by a
           <close-session> operation, or killed by a <kill-session>
           operation.";
      }
      uses common-counters {
        description
          "Global counters, accumulated from all sessions.
           Zero based with following reset behaviour:
             - re-initialization of NETCONF server
             - when max value is reached";
      }
    }
  }

  rpc get-schema {
    description
      "This operation is used to retrieve a schema from the
       NETCONF server.

       Positive Response:
         The NETCONF server returns the requested schema.

       Negative Response:
         If requested schema does not exist, the <error-tag> is
         'invalid-value'.

         If more than one schema matches the requested parameters, the
         <error-tag> is 'operation-failed', and <error-app-tag> is
         'data-not-unique'.";

    input {
      leaf identifier {
        type string;
        mandatory true;
        description
          "Identifier for the schema list entry.";
      }
      leaf version {
        type string;
        description
          "Version of the schema requested.  If this parameter is not
           present, and more than one version of the schema exists on
           the server, a 'data-not-unique' error is returned, as
```


Copyright© 2022 Ciena® Corporation

```
            described above.";
        }
        leaf format {
          type identityref {
            base schema-format;
          }
          description
             "The data modeling language of the schema.  If this
              parameter is not present, and more than one formats of
              the schema exists on the server, a 'data-not-unique' error
              is returned, as described above.";
        }
      }
      output {
          anyxml data {
            description
              "Contains the schema content.";
        }
      }
    }
}
```


Copyright© 2022 Ciena® Corporation

ietf-snmp-common.yang

```
submodule ietf-snmp-common {

  belongs-to ietf-snmp {
    prefix snmp;
  }

  import ietf-yang-types {
    prefix yang;
  }

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

  contact
    "WG Web:   <http://tools.ietf.org/wg/netmod/>
     WG List:  <mailto:netmod@ietf.org>
     WG Chair: Thomas Nadeau
               <mailto:tnadeau@lucidvision.com>

     WG Chair: Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>

     Editor:   Martin Bjorklund
               <mailto:mbj@tail-f.com>

     Editor:   Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>";

  description
    "This submodule contains a collection of common YANG definitions
     for configuring SNMP engines.

     Copyright (c) 2014 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject
     to the license terms contained in, the Simplified BSD License
     set forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC 7407; see
     the RFC itself for full legal notices.";

  revision 2014-12-10 {
    description
      "Initial revision.";
    reference
      "RFC 7407: A YANG Data Model for SNMP Configuration";
  }

  /* Collection of SNMP-specific data types */

  typedef admin-string {
    type string {
      length "0..255";
    }
```

```
    description
      "Represents SnmpAdminString as defined in RFC 3411.

      Note that the size of an SnmpAdminString is measured in
      octets, not characters.";

    reference
      "RFC 3411: An Architecture for Describing Simple Network
         Management Protocol (SNMP) Management Frameworks.
         SNMP-FRAMEWORK-MIB.SnmpAdminString";
  }

  typedef identifier {
    type admin-string {
      length "1..32";
    }
    description
      "Identifiers are used to name items in the SNMP configuration
      datastore.";
  }

  typedef context-name {
    type admin-string {
      length "0..32";
    }
    description
      "The context type represents an SNMP context name.";
    reference
      "RFC 3411: An Architecture for Describing Simple Network
         Management Protocol (SNMP) Management Frameworks";
  }

  typedef security-name {
    type admin-string {
      length "1..32";
    }
    description
      "The security-name type represents an SNMP security name.";
    reference
      "RFC 3411: An Architecture for Describing Simple Network
         Management Protocol (SNMP) Management Frameworks";
  }

  typedef security-model {
    type union {
      type enumeration {
        enum v1  { value 1; }
        enum v2c { value 2; }
        enum usm { value 3; }
        enum tsm { value 4; }
      }
      type int32 {
        range "1..2147483647";
      }
    }
    reference
      "RFC 3411: An Architecture for Describing Simple Network
         Management Protocol (SNMP) Management Frameworks";
  }
```


Copyright© 2022 Ciena® Corporation

```
  typedef security-model-or-any {
    type union {
      type enumeration {
        enum any { value 0; }
      }
      type security-model;
    }
    reference
      "RFC 3411: An Architecture for Describing Simple Network
         Management Protocol (SNMP) Management Frameworks";
  }

  typedef security-level {
    type enumeration {
      enum no-auth-no-priv { value 1; }
      enum auth-no-priv    { value 2; }
      enum auth-priv       { value 3; }
    }
    reference
      "RFC 3411: An Architecture for Describing Simple Network
         Management Protocol (SNMP) Management Frameworks";
  }

  typedef engine-id {
    type yang:hex-string {
      pattern '([0-9a-fA-F]){2}(:([0-9a-fA-F]){2}){4,31}';
    }
    description
      "The engine ID specified as a list of colon-specified
      hexadecimal octets, e.g., '80:00:02:b8:04:61:62:63'.";
    reference
      "RFC 3411: An Architecture for Describing Simple Network
         Management Protocol (SNMP) Management Frameworks";
  }

  typedef wildcard-object-identifier {
    type string;
    description
      "The wildcard-object-identifier type represents an SNMP object
      identifier where subidentifiers can be given either as a label,
      in numeric form, or a wildcard, represented by an asterisk
      ('*').";
  }

  typedef tag-value {
    type string {
      length "0..255";
    }
    description
      "Represents SnmpTagValue as defined in RFC 3413.

      Note that the size of an SnmpTagValue is measured in
      octets, not characters.";
    reference
      "RFC 3413: Simple Network Management Protocol (SNMP)
         Applications.
         SNMP-TARGET-MIB.SnmpTagValue";
  }

  container snmp {
```


Copyright© 2022 Ciena® Corporation

```
    description
      "Top-level container for SNMP-related configuration and
      status objects.";
  }

}
```


Copyright© 2022 Ciena® Corporation

ietf-snmp-community.yang

```
submodule ietf-snmp-community {

  belongs-to ietf-snmp {
    prefix snmp;
  }

  import ietf-netconf-acm {
    prefix nacm;
  }

  include ietf-snmp-common;
  include ietf-snmp-target;
  include ietf-snmp-proxy;

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

  contact
    "WG Web:   <http://tools.ietf.org/wg/netmod/>
     WG List:  <mailto:netmod@ietf.org>

     WG Chair: Thomas Nadeau
               <mailto:tnadeau@lucidvision.com>

     WG Chair: Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>
     Editor:   Martin Bjorklund
               <mailto:mbj@tail-f.com>

     Editor:   Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>";

  description
    "This submodule contains a collection of YANG definitions
     for configuring community-based SNMP.

     Copyright (c) 2014 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject
     to the license terms contained in, the Simplified BSD License
     set forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC 7407; see
     the RFC itself for full legal notices.";

  reference
    "RFC 3584: Coexistence between Version 1, Version 2, and
       Version 3 of the Internet-standard Network
       Management Framework";

  revision 2014-12-10 {
    description
      "Initial revision.";
    reference
```


Copyright© 2022 Ciena® Corporation

```
      "RFC 7407: A YANG Data Model for SNMP Configuration";
  }

  augment /snmp:snmp {

    list community {
      key index;

      description
        "List of communities.";
      reference
        "RFC 3584: Coexistence between Version 1, Version 2,
           and Version 3 of the Internet-standard
           Network Management Framework.
           SNMP-COMMUNITY-MIB.snmpCommunityTable";

      leaf index {
        type snmp:identifier;
        description
          "Index into the community list.";
        reference
          "RFC 3584: Coexistence between Version 1, Version 2,
             and Version 3 of the Internet-standard
             Network Management Framework.
             SNMP-COMMUNITY-MIB.snmpCommunityIndex";
      }
      choice name {
        nacm:default-deny-all;
        description
          "The community name, specified as either a string or
           a binary value.  The binary name is used when the
           community name contains characters that are not legal
           in a string.

           If not set, the value of 'security-name' is operationally
           used as the snmpCommunityName.";
        reference
          "RFC 3584: Coexistence between Version 1, Version 2,
             and Version 3 of the Internet-standard
             Network Management Framework.
             SNMP-COMMUNITY-MIB.snmpCommunityName";
        leaf text-name {
          type string;
          description
            "A community name that can be represented as a
             YANG string.";
        }
        leaf binary-name {
          type binary;
          description
            "A community name represented as a binary value.";
        }
      }
      leaf security-name {
        type snmp:security-name;
        mandatory true;
        nacm:default-deny-all;
        description
          "The snmpCommunitySecurityName of this entry.";
        reference
```


Copyright© 2022 Ciena® Corporation

```
          "RFC 3584: Coexistence between Version 1, Version 2,
             and Version 3 of the Internet-standard
             Network Management Framework.
             SNMP-COMMUNITY-MIB.snmpCommunitySecurityName";
      }
      leaf engine-id {
        if-feature snmp:proxy;
        type snmp:engine-id;
        description
          "If not set, the value of the local SNMP engine is
           operationally used by the device.";
        reference
          "RFC 3584: Coexistence between Version 1, Version 2,
             and Version 3 of the Internet-standard
             Network Management Framework.
             SNMP-COMMUNITY-MIB.snmpCommunityContextEngineID";
      }
      leaf context {
        type snmp:context-name;
        default "";
        description
          "The context in which management information is accessed
           when using the community string specified by this entry.";
        reference
          "RFC 3584: Coexistence between Version 1, Version 2,
             and Version 3 of the Internet-standard
             Network Management Framework.
             SNMP-COMMUNITY-MIB.snmpCommunityContextName";
      }
      leaf target-tag {
        type snmp:tag-value;
        description
          "Used to limit access for this community to the specified
           targets.

           Implementations MAY restrict the values of this leaf
           to be one of the available values of /snmp/target/tag in
           a valid configuration.";
        reference
          "RFC 3584: Coexistence between Version 1, Version 2,
             and Version 3 of the Internet-standard
             Network Management Framework.
             SNMP-COMMUNITY-MIB.snmpCommunityTransportTag";
      }
    }
  }

  grouping v1-target-params {
    container v1 {
      description
        "SNMPv1 parameters type.
         Represents snmpTargetParamsMPModel '0',
         snmpTargetParamsSecurityModel '1', and
         snmpTargetParamsSecurityLevel 'noAuthNoPriv'.";
      leaf security-name {
        type snmp:security-name;
        mandatory true;
        description
          "Implementations MAY restrict the values of this leaf
           to be one of the available values of
```


Copyright© 2022 Ciena® Corporation

```
          /snmp/community/security-name in a valid configuration.";
        reference
          "RFC 3413: Simple Network Management Protocol (SNMP).
            Applications.
            SNMP-TARGET-MIB.snmpTargetParamsSecurityName";
      }
    }
  }

  grouping v2c-target-params {
    container v2c {
      description
        "SNMPv2 community parameters type.
         Represents snmpTargetParamsMPModel '1',
         snmpTargetParamsSecurityModel '2', and
         snmpTargetParamsSecurityLevel 'noAuthNoPriv'.";
      leaf security-name {
        type snmp:security-name;
        mandatory true;
        description
          "Implementations MAY restrict the values of this leaf
           to be one of the available values of
           /snmp/community/security-name in a valid configuration.";
        reference
          "RFC 3413: Simple Network Management Protocol (SNMP).
            Applications.
            SNMP-TARGET-MIB.snmpTargetParamsSecurityName";
      }
    }
  }

  augment /snmp:snmp/snmp:target-params/snmp:params {
    case v1 {
      uses v1-target-params;
    }
    case v2c {
      uses v2c-target-params;
    }
  }

  augment /snmp:snmp/snmp:target {
    when "snmp:v1 or snmp:v2c";
    leaf mms {
      type union {
        type enumeration {
          enum "unknown" { value 0; }
        }
        type int32 {
          range "484..max";
        }
      }
      default "484";
      description
        "The maximum message size.";
      reference
        "RFC 3584: Coexistence between Version 1, Version 2,
           and Version 3 of the Internet-standard
           Network Management Framework.
           SNMP-COMMUNITY-MIB.snmpTargetAddrMMS";
    }
```


Copyright© 2022 Ciena® Corporation

```
    }
}
```


Copyright© 2022 Ciena® Corporation

ietf-snmp-engine.yang

```
submodule ietf-snmp-engine {

  belongs-to ietf-snmp {
    prefix snmp;
  }

  import ietf-inet-types {
    prefix inet;
  }

  include ietf-snmp-common;

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

  contact
    "WG Web:   <http://tools.ietf.org/wg/netmod/>
     WG List:  <mailto:netmod@ietf.org>

     WG Chair: Thomas Nadeau
               <mailto:tnadeau@lucidvision.com>

     WG Chair: Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>

     Editor:   Martin Bjorklund
               <mailto:mbj@tail-f.com>

     Editor:   Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>";

  description
    "This submodule contains a collection of YANG definitions
     for configuring SNMP engines.

     Copyright (c) 2014 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject
     to the license terms contained in, the Simplified BSD License
     set forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (http://trustee.ietf.org/license-info).
     This version of this YANG module is part of RFC 7407; see
     the RFC itself for full legal notices.";

  revision 2014-12-10 {
    description
      "Initial revision.";
    reference
      "RFC 7407: A YANG Data Model for SNMP Configuration";
  }

  augment /snmp:snmp {

    container engine {
```

```
description
  "Configuration of the SNMP engine.";

leaf enabled {
  type boolean;
  default "false";
  description
    "Enables the SNMP engine.";
}

list listen {
  key "name";
  description
    "Configuration of the transport endpoints on which the
     engine listens.";

  leaf name {
    type snmp:identifier;
    description
      "An arbitrary name for the list entry.";
  }

  choice transport {
    mandatory true;
    description
      "The transport-protocol-specific parameters for this
       endpoint.  Submodules providing configuration for
       additional transports are expected to augment this
       choice.";
    case udp {
      container udp {
        leaf ip {
          type inet:ip-address;
          mandatory true;
          description
            "The IPv4 or IPv6 address on which the engine
             listens.";
        }
        leaf port {
          type inet:port-number;
          description
            "The UDP port on which the engine listens.

             If the port is not configured, an engine that
             acts as a Command Responder uses port 161, and
             an engine that acts as a Notification Receiver
             uses port 162.";
        }
      }
    }
  }
}



container version {
  description
    "SNMP version used by the engine.";
  leaf v1 {
    type empty;
```


Copyright© 2022 Ciena® Corporation

```
        }
        leaf v2c {
          type empty;
        }
        leaf v3 {
          type empty;
        }
      }

      leaf engine-id {
        type snmp:engine-id;
        description
          "The local SNMP engine's administratively assigned unique
           identifier.

           If this leaf is not set, the device automatically
           calculates an engine ID, as described in RFC 3411.  A
           server MAY initialize this leaf with the automatically
           created value.";
        reference
          "RFC 3411: An Architecture for Describing Simple Network
             Management Protocol (SNMP) Management
             Frameworks.
             SNMP-FRAMEWORK-MIB.snmpEngineID";
      }

      leaf enable-authen-traps {
        type boolean;
        description
          "Indicates whether the SNMP entity is permitted to
           generate authenticationFailure traps.";
        reference
          "RFC 3418: Management Information Base (MIB) for the
             Simple Network Management Protocol (SNMP)
             SNMPv2-MIB.snmpEnableAuthenTraps";
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ietf-snmp-notification.yang

```
submodule ietf-snmp-notification {

  belongs-to ietf-snmp {
    prefix snmp;
  }

  include ietf-snmp-common;
  include ietf-snmp-target;

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

  contact
    "WG Web:   <http://tools.ietf.org/wg/netmod/>
     WG List:  <mailto:netmod@ietf.org>

     WG Chair: Thomas Nadeau
               <mailto:tnadeau@lucidvision.com>
     WG Chair: Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>

     Editor:   Martin Bjorklund
               <mailto:mbj@tail-f.com>

     Editor:   Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>";

  description
    "This submodule contains a collection of YANG definitions
     for configuring SNMP notifications.

     Copyright (c) 2014 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject
     to the license terms contained in, the Simplified BSD License
     set forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC 7407; see
     the RFC itself for full legal notices.";

  reference
    "RFC 3413: Simple Network Management Protocol (SNMP)
       Applications";

  revision 2014-12-10 {
    description
      "Initial revision.";
    reference
      "RFC 7407: A YANG Data Model for SNMP Configuration";
  }

  feature notification-filter {
    description
      "A server implements this feature if it supports SNMP
```

```
      notification filtering.";
    reference
      "RFC 3413: Simple Network Management Protocol (SNMP)
         Applications";
  }

  augment /snmp:snmp {

    list notify {
      key name;
      description
        "Targets that will receive notifications.

         Entries in this list are mapped 1-1 to entries in
         snmpNotifyTable, except that if an entry in snmpNotifyTable
         has an snmpNotifyTag for which no snmpTargetAddrEntry
         exists, then the snmpNotifyTable entry is not mapped to an
         entry in this list.";
      reference
        "RFC 3413: Simple Network Management Protocol (SNMP).
           Applications.
           SNMP-NOTIFICATION-MIB.snmpNotifyTable";

      leaf name {
        type snmp:identifier;
        description
          "An arbitrary name for the list entry.";
        reference
          "RFC 3413: Simple Network Management Protocol (SNMP).
             Applications.
             SNMP-NOTIFICATION-MIB.snmpNotifyName";
      }
      leaf tag {
        type snmp:tag-value;
        mandatory true;
        description
          "Target tag, selects a set of notification targets.

           Implementations MAY restrict the values of this leaf
           to be one of the available values of /snmp/target/tag in
           a valid configuration.";
        reference
          "RFC 3413: Simple Network Management Protocol (SNMP).
             Applications.
             SNMP-NOTIFICATION-MIB.snmpNotifyTag";
      }
      leaf type {
        type enumeration {
          enum trap { value 1; }
          enum inform { value 2; }
        }
        default trap;
        description
          "Defines the notification type to be generated.";


        reference
          "RFC 3413: Simple Network Management Protocol (SNMP).
             Applications.
             SNMP-NOTIFICATION-MIB.snmpNotifyType";
```


Copyright© 2022 Ciena® Corporation

```
    }
  }

  list notify-filter-profile {
    if-feature snmp:notification-filter;
    key name;

    description
      "Notification filter profiles.

       The leaf /snmp/target/notify-filter-profile is used
       to associate a filter profile with a target.

       If an entry in this list is referred to by one or more
       /snmp/target/notify-filter-profile items, each such
       notify-filter-profile is represented by one
       snmpNotifyFilterProfileEntry.

       If an entry in this list is not referred to by any
       /snmp/target/notify-filter-profile, the entry is not mapped
       to snmpNotifyFilterProfileTable.";
    reference
      "RFC 3413: Simple Network Management Protocol (SNMP).
         Applications.
         SNMP-NOTIFICATION-MIB.snmpNotifyFilterProfileTable
         SNMP-NOTIFICATION-MIB.snmpNotifyFilterTable";

    leaf name {
      type snmp:identifier;
      description
        "Name of the filter profile.";
      reference
        "RFC 3413: Simple Network Management Protocol (SNMP).
           Applications.
           SNMP-NOTIFICATION-MIB.snmpNotifyFilterProfileName";
    }

    leaf-list include {
      type snmp:wildcard-object-identifier;
      description
        "A family of subtrees included in this filter.";



      reference
        "RFC 3413: Simple Network Management Protocol (SNMP).
           Applications.
           SNMP-NOTIFICATION-MIB.snmpNotifyFilterSubtree
           SNMP-NOTIFICATION-MIB.snmpNotifyFilterMask
           SNMP-NOTIFICATION-MIB.snmpNotifyFilterType";
    }

    leaf-list exclude {
      type snmp:wildcard-object-identifier;
      description
        "A family of subtrees excluded from this filter.";
      reference
        "RFC 3413: Simple Network Management Protocol (SNMP).
           Applications.
           SNMP-NOTIFICATION-MIB.snmpNotifyFilterSubtree
```


Copyright© 2022 Ciena® Corporation

```
              SNMP-NOTIFICATION-MIB.snmpNotifyFilterMask
              SNMP-NOTIFICATION-MIB.snmpNotifyFilterType";
      }
    }

  }

  augment /snmp:snmp/snmp:target-params {
    reference
      "RFC 3413: Simple Network Management Protocol (SNMP).
         Applications.
         SNMP-NOTIFICATION-MIB.snmpNotifyFilterProfileTable";
    leaf notify-filter-profile {
      if-feature snmp:notification-filter;
      type leafref {
        path "/snmp/notify-filter-profile/name";
      }
      description
        "This leafref leaf is used to represent the sparse
         relationship between the /snmp/target-params list and the
         /snmp/notify-filter-profile list.";
      reference
        "RFC 3413: Simple Network Management Protocol (SNMP).
           Applications.
           SNMP-NOTIFICATION-MIB.snmpNotifyFilterProfileName";
    }
  }

}
```


Copyright© 2022 Ciena® Corporation

ietf-snmp-proxy.yang

```
submodule ietf-snmp-proxy {

  belongs-to ietf-snmp {
    prefix snmp;
  }

  include ietf-snmp-common;
  include ietf-snmp-target;

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

  contact
    "WG Web:   <http://tools.ietf.org/wg/netmod/>
     WG List:  <mailto:netmod@ietf.org>

     WG Chair: Thomas Nadeau
               <mailto:tnadeau@lucidvision.com>

     WG Chair: Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>

     Editor:   Martin Bjorklund
               <mailto:mbj@tail-f.com>

     Editor:   Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>";

  description
    "This submodule contains a collection of YANG definitions
     for configuring SNMP proxies.

     Copyright (c) 2014 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject
     to the license terms contained in, the Simplified BSD License
     set forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC 7407; see
     the RFC itself for full legal notices.";
  reference
    "RFC 3413: Simple Network Management Protocol (SNMP)
       Applications";

  revision 2014-12-10 {
    description
      "Initial revision.";
    reference
      "RFC 7407: A YANG Data Model for SNMP Configuration";
  }

  feature proxy {
    description
      "A server implements this feature if it can act as an
```

```
      SNMP proxy.";
    reference
      "RFC 3413: Simple Network Management Protocol (SNMP)
         Applications";
  }

  augment /snmp:snmp {
    if-feature snmp:proxy;

    list proxy {
      key name;

      description
        "List of proxy parameters.";
      reference
        "RFC 3413: Simple Network Management Protocol (SNMP).
           Applications.
           SNMP-PROXY-MIB.snmpProxyTable";

      leaf name {
        type snmp:identifier;
        description
          "Identifies the proxy parameter entry.";
        reference
          "RFC 3413: Simple Network Management Protocol (SNMP).
             Applications.
             SNMP-PROXY-MIB.snmpProxyName";
      }
      leaf type {
        type enumeration {
          enum read { value 1; }
          enum write { value 2; }
          enum trap { value 3; }
          enum inform { value 4; }
        }
        mandatory true;
        reference
          "RFC 3413: Simple Network Management Protocol (SNMP).
             Applications.
             SNMP-PROXY-MIB.snmpProxyType";
      }
      leaf context-engine-id {
        type snmp:engine-id;
        mandatory true;
        reference
          "RFC 3413: Simple Network Management Protocol (SNMP).
             Applications.
             SNMP-PROXY-MIB.snmpProxyContextEngineID";
      }
      leaf context-name {
        type snmp:context-name;
        reference
          "RFC 3413: Simple Network Management Protocol (SNMP).
             Applications.
             SNMP-PROXY-MIB.snmpProxyContextName";
      }
      leaf target-params-in {
        type snmp:identifier;
        description
          "The name of a target parameters list entry.
```


Copyright© 2022 Ciena® Corporation

```
            Implementations MAY restrict the values of this
            leaf to be one of the available values of
            /snmp/target-params/name in a valid configuration.";
          reference
            "RFC 3413: Simple Network Management Protocol (SNMP).
              Applications.
              SNMP-PROXY-MIB.snmpProxyTargetParamsIn";
        }
        leaf single-target-out {
          when "../type = 'read' or ../type = 'write'";
          type snmp:identifier;
          description
            "Implementations MAY restrict the values of this leaf
             to be one of the available values of /snmp/target/name in
             a valid configuration.";
          reference
            "RFC 3413: Simple Network Management Protocol (SNMP).
              Applications.
              SNMP-PROXY-MIB.snmpProxySingleTargetOut";
        }

        leaf multiple-target-out {
          when "../type = 'trap' or ../type = 'inform'";
          type snmp:tag-value;
          description
            "Implementations MAY restrict the values of this leaf
             to be one of the available values of /snmp/target/tag in
             a valid configuration.";
          reference
            "RFC 3413: Simple Network Management Protocol (SNMP).
              Applications.
              SNMP-PROXY-MIB.snmpProxyMultipleTargetOut";
        }
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ietf-snmp-ssh.yang

```
submodule ietf-snmp-ssh {

  belongs-to ietf-snmp {
    prefix snmp;
  }

  import ietf-inet-types {
    prefix inet;
  }

  include ietf-snmp-common;
  include ietf-snmp-engine;
  include ietf-snmp-target;

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

  contact
    "WG Web:   <http://tools.ietf.org/wg/netmod/>
     WG List:  <mailto:netmod@ietf.org>

     WG Chair: Thomas Nadeau
               <mailto:tnadeau@lucidvision.com>

     WG Chair: Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>

     Editor:   Martin Bjorklund
               <mailto:mbj@tail-f.com>

     Editor:   Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>";

  description
    "This submodule contains a collection of YANG definitions for
     configuring the Secure Shell Transport Model (SSHTM)
     of SNMP.

     Copyright (c) 2014 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject
     to the license terms contained in, the Simplified BSD License
     set forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC 7407; see
     the RFC itself for full legal notices.";

  reference
    "RFC 5592: Secure Shell Transport Model for the
       Simple Network Management Protocol (SNMP)";

  revision 2014-12-10 {
    description
      "Initial revision.";
```

```
    reference
      "RFC 7407: A YANG Data Model for SNMP Configuration";
  }

  feature sshtm {
    description
      "A server implements this feature if it supports the
      Secure Shell Transport Model for SNMP.";
    reference
      "RFC 5592: Secure Shell Transport Model for the
         Simple Network Management Protocol (SNMP)";
  }

  augment /snmp:snmp/snmp:engine/snmp:listen/snmp:transport {
    if-feature sshtm;
    case ssh {
      container ssh {
        description
          "The IPv4 or IPv6 address and port to which the
           engine listens for SNMP messages over SSH.";

        leaf ip {
          type inet:ip-address;
          mandatory true;
          description
            "The IPv4 or IPv6 address on which the engine listens
             for SNMP messages over SSH.";
        }
        leaf port {
          type inet:port-number;
          description
            "The TCP port on which the engine listens for SNMP
             messages over SSH.

             If the port is not configured, an engine that
             acts as a Command Responder uses port 5161, and
             an engine that acts as a Notification Receiver
             uses port 5162.";
        }
      }
    }
  }

  augment /snmp:snmp/snmp:target/snmp:transport {
    if-feature sshtm;
    case ssh {
      reference
        "RFC 5592: Secure Shell Transport Model for the
           Simple Network Management Protocol (SNMP).
           SNMP-SSH-TM-MIB.snmpSSHDomain";
      container ssh {
        leaf ip {
          type inet:host;
          mandatory true;
          reference
            "RFC 3413: Simple Network Management Protocol (SNMP).
               Applications.
               SNMP-TARGET-MIB.snmpTargetAddrTAddress
             RFC 5592: Secure Shell Transport Model for the
               Simple Network Management Protocol (SNMP).
```


Copyright© 2022 Ciena® Corporation

```
                SNMP-SSH-TM-MIB.SnmpSSHAddress";
        }
        leaf port {
          type inet:port-number;
          default 5161;
          reference
            "RFC 3413: Simple Network Management Protocol (SNMP).
               Applications.
               SNMP-TARGET-MIB.snmpTargetAddrTAddress
             RFC 5592: Secure Shell Transport Model for the
               Simple Network Management Protocol (SNMP).
               SNMP-SSH-TM-MIB.SnmpSSHAddress";
        }
        leaf username {
          type string;
          reference
            "RFC 3413: Simple Network Management Protocol (SNMP).
               Applications.
               SNMP-TARGET-MIB.snmpTargetAddrTAddress
             RFC 5592: Secure Shell Transport Model for the
               Simple Network Management Protocol (SNMP).
               SNMP-SSH-TM-MIB.SnmpSSHAddress";
        }
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ietf-snmp-target.yang

```
submodule ietf-snmp-target {

  belongs-to ietf-snmp {
    prefix snmp;
  }

  import ietf-inet-types {
    prefix inet;
  }

  include ietf-snmp-common;

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

  contact
    "WG Web:   <http://tools.ietf.org/wg/netmod/>
     WG List:  <mailto:netmod@ietf.org>

     WG Chair: Thomas Nadeau
               <mailto:tnadeau@lucidvision.com>

     WG Chair: Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>

     Editor:   Martin Bjorklund
               <mailto:mbj@tail-f.com>

     Editor:   Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>";

  description
    "This submodule contains a collection of YANG definitions
     for configuring SNMP targets.

     Copyright (c) 2014 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject
     to the license terms contained in, the Simplified BSD License
     set forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (http://trustee.ietf.org/license-info).
     This version of this YANG module is part of RFC 7407; see
     the RFC itself for full legal notices.";

  reference
    "RFC 3413: Simple Network Management Protocol (SNMP)
       Applications";

  revision 2014-12-10 {
    description
      "Initial revision.";
    reference
      "RFC 7407: A YANG Data Model for SNMP Configuration";
  }
```


Copyright© 2022 Ciena® Corporation

```
augment /snmp:snmp {

  list target {
    key name;
    description
      "List of targets.";
    reference
      "RFC 3413: Simple Network Management Protocol (SNMP)
         Applications.
         SNMP-TARGET-MIB.snmpTargetAddrTable";

    leaf name {
      type snmp:identifier;
      description
        "Identifies the target.";
      reference
        "RFC 3413: Simple Network Management Protocol (SNMP)
           Applications.
           SNMP-TARGET-MIB.snmpTargetAddrName";
    }
    choice transport {
      mandatory true;
      description
        "Transport address of the target.

         The snmpTargetAddrTDomain and snmpTargetAddrTAddress
         objects are mapped to transport-specific YANG nodes.  Each
         transport is configured as a separate case in this
         choice.  Submodules providing configuration for additional
         transports are expected to augment this choice.";



      reference
        "RFC 3413: Simple Network Management Protocol (SNMP)
           Applications.
           SNMP-TARGET-MIB.snmpTargetAddrTDomain
           SNMP-TARGET-MIB.snmpTargetAddrTAddress";
      case udp {
        reference
         "RFC 3417: Transport Mappings for the Simple Network
            Management Protocol (SNMP).
            SNMPv2-TM.snmpUDPDomain
          RFC 3419: Textual Conventions for Transport Addresses.
            TRANSPORT-ADDRESS-MIB.transportDomainUdpIpv4
            TRANSPORT-ADDRESS-MIB.transportDomainUdpIpv4z
            TRANSPORT-ADDRESS-MIB.transportDomainUdpIpv6
            TRANSPORT-ADDRESS-MIB.transportDomainUdpIpv6z";
        container udp {
          leaf ip {
            type inet:ip-address;
            mandatory true;
            reference
              "RFC 3413: Simple Network Management Protocol (SNMP).
                 SNMP-TARGET-MIB.snmpTargetAddrTAddress";
          }
          leaf port {
            type inet:port-number;
            default 162;
```


Copyright© 2022 Ciena® Corporation

```
          description
            "UDP port number.";
          reference
            "RFC 3413: Simple Network Management Protocol (SNMP).
               SNMP-TARGET-MIB.snmpTargetAddrTAddress";
        }
        leaf prefix-length {
          type uint8;
          description
            "The value of this leaf must match the value of
             ../snmp:ip.  If ../snmp:ip contains an IPv4 address,
             this leaf must be less than or equal to 32.  If it
             contains an IPv6 address, it must be less than or
             equal to 128.

             Note that the prefix-length is currently only used
             by the Community-based Security Model to filter
             incoming messages.  Furthermore, the prefix-length
             filtering does not cover all possible filters
             supported by the corresponding MIB object.";


          reference
            "RFC 3584: Coexistence between Version 1, Version 2,
               and Version 3 of the Internet-standard
               Network Management Framework.
               SNMP-COMMUNITY-MIB.snmpTargetAddrTMask";
        }
      }
    }
  }
  leaf-list tag {
    type snmp:tag-value;
    description
      "List of tag values used to select target addresses.";
    reference
      "RFC 3413: Simple Network Management Protocol (SNMP).
         Applications.
         SNMP-TARGET-MIB.snmpTargetAddrTagList";
  }
  leaf timeout {
    type uint32;
    units "0.01 seconds";
    default 1500;
    description
      "Needed only if this target can receive
       InformRequest-PDUs.";
    reference
      "RFC 3413: Simple Network Management Protocol (SNMP).
         Applications.
         SNMP-TARGET-MIB.snmpTargetAddrTimeout";
  }
  leaf retries {
    type uint8;
    default 3;
    description
      "Needed only if this target can receive
       InformRequest-PDUs.";
    reference
      "RFC 3413: Simple Network Management Protocol (SNMP).
```


Copyright© 2022 Ciena® Corporation

```
              Applications.
              SNMP-TARGET-MIB.snmpTargetAddrRetryCount";
      }
      leaf target-params {
        type snmp:identifier;
        mandatory true;
        reference
          "RFC 3413: Simple Network Management Protocol (SNMP).
             Applications.
             SNMP-TARGET-MIB.snmpTargetAddrParams";
      }
    }

    list target-params {
      key name;
      description
        "List of target parameters.";
      reference
        "RFC 3413: Simple Network Management Protocol (SNMP).
           Applications.
           SNMP-TARGET-MIB.snmpTargetParamsTable";

      leaf name {
        type snmp:identifier;
      }
      choice params {
        description
          "This choice is augmented with case nodes containing
           configuration parameters specific to the security model.";
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ietf-snmp-tls.yang

```
submodule ietf-snmp-tls {

  belongs-to ietf-snmp {
    prefix snmp;
  }

  import ietf-inet-types {
    prefix inet;
  }
  import ietf-x509-cert-to-name {
    prefix x509c2n;
  }

  include ietf-snmp-common;
  include ietf-snmp-engine;
  include ietf-snmp-target;

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

  contact
    "WG Web:   <http://tools.ietf.org/wg/netmod/>
     WG List:  <mailto:netmod@ietf.org>

     WG Chair: Thomas Nadeau
               <mailto:tnadeau@lucidvision.com>

     WG Chair: Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>

     Editor:   Martin Bjorklund
               <mailto:mbj@tail-f.com>

     Editor:   Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>";



  description
    "This submodule contains a collection of YANG definitions for
     configuring the Transport Layer Security Transport Model (TLSTM)
     of SNMP.

     Copyright (c) 2014 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject
     to the license terms contained in, the Simplified BSD License
     set forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC 7407; see
     the RFC itself for full legal notices.";

  reference
    "RFC 6353: Transport Layer Security (TLS) Transport Model for
```


Copyright© 2022 Ciena® Corporation

```
      the Simple Network Management Protocol (SNMP)";

  revision 2014-12-10 {
    description
      "Initial revision.";
    reference
      "RFC 7407: A YANG Data Model for SNMP Configuration";
  }

  feature tlstm {
    description
      "A server implements this feature if it supports the
      Transport Layer Security Transport Model for SNMP.";
    reference
      "RFC 6353: Transport Layer Security (TLS) Transport Model for
         the Simple Network Management Protocol (SNMP)";
  }

  augment /snmp:snmp/snmp:engine/snmp:listen/snmp:transport {
    if-feature tlstm;
    case tls {
      container tls {
        description
          "A list of IPv4 and IPv6 addresses and ports to which the
           engine listens for SNMP messages over TLS.";



        leaf ip {
          type inet:ip-address;
          mandatory true;
          description
            "The IPv4 or IPv6 address on which the engine listens
             for SNMP messages over TLS.";
        }
        leaf port {
          type inet:port-number;
          description
            "The TCP port on which the engine listens for SNMP
             messages over TLS.

             If the port is not configured, an engine that
             acts as a Command Responder uses port 10161, and
             an engine that acts as a Notification Receiver
             uses port 10162.";
        }
      }
    }
    case dtls {
      container dtls {
        description
          "A list of IPv4 and IPv6 addresses and ports to which the
           engine listens for SNMP messages over DTLS.";

        leaf ip {
          type inet:ip-address;
          mandatory true;
          description
            "The IPv4 or IPv6 address on which the engine listens
             for SNMP messages over DTLS.";
```


Copyright© 2022 Ciena® Corporation

```
        }
        leaf port {
          type inet:port-number;
          description
            "The UDP port on which the engine listens for SNMP
             messages over DTLS.

             If the port is not configured, an engine that
             acts as a Command Responder uses port 10161, and
             an engine that acts as a Notification Receiver
             uses port 10162.";
        }
      }
    }
  }

  augment /snmp:snmp {
    if-feature tlstm;
    container tlstm {
      uses x509c2n:cert-to-name {
        description
          "Defines how certificates are mapped to names.  The
           resulting name is used as a security name.";
        refine cert-to-name/map-type {
          description
            "Mappings that use the snmpTlstmCertToTSNData column
             need to augment the cert-to-name list with
             additional configuration objects corresponding
             to the snmpTlstmCertToTSNData value.  Such objects
             should use the 'when' statement to make them
             conditional based on the map-type.";
        }
      }
    }
  }

  grouping tls-transport {
    leaf ip {
      type inet:host;
      mandatory true;
      reference
        "RFC 3413: Simple Network Management Protocol (SNMP).
           Applications.
           SNMP-TARGET-MIB.snmpTargetAddrTAddress
         RFC 6353: Transport Layer Security (TLS) Transport Model
           for the Simple Network Management Protocol (SNMP).
           SNMP-TLS-TM-MIB.SnmpTLSAddress";
    }
    leaf port {
      type inet:port-number;
      default 10161;
      reference
        "RFC 3413: Simple Network Management Protocol (SNMP).
           Applications.
           SNMP-TARGET-MIB.snmpTargetAddrTAddress
         RFC 6353: Transport Layer Security (TLS) Transport Model
           for the Simple Network Management Protocol (SNMP).
           SNMP-TLS-TM-MIB.SnmpTLSAddress";
    }
    leaf client-fingerprint {
```


Copyright© 2022 Ciena® Corporation

```
      type x509c2n:tls-fingerprint;
      reference
        "RFC 6353: Transport Layer Security (TLS) Transport Model
           for the Simple Network Management Protocol (SNMP).
           SNMP-TLS-TM-MIB.snmpTlstmParamsClientFingerprint";
    }
    leaf server-fingerprint {
      type x509c2n:tls-fingerprint;
      reference
        "RFC 6353: Transport Layer Security (TLS) Transport Model
           for the Simple Network Management Protocol (SNMP).
           SNMP-TLS-TM-MIB.snmpTlstmAddrServerFingerprint";
    }
    leaf server-identity {
      type snmp:admin-string;
      reference
        "RFC 6353: Transport Layer Security (TLS) Transport Model
           for the Simple Network Management Protocol (SNMP).
           SNMP-TLS-TM-MIB.snmpTlstmAddrServerIdentity";
    }
  }

  augment /snmp:snmp/snmp:target/snmp:transport {
    if-feature tlstm;
    case tls {
      reference
        "RFC 6353: Transport Layer Security (TLS) Transport Model
           for the Simple Network Management Protocol (SNMP).
           SNMP-TLS-TM-MIB.snmpTLSTCPDomain";
      container tls {
        uses tls-transport;
      }
    }
  }

  augment /snmp:snmp/snmp:target/snmp:transport {
    if-feature tlstm;
    case dtls {
      reference
        "RFC 6353: Transport Layer Security (TLS) Transport Model
           for the Simple Network Management Protocol (SNMP).
           SNMP-TLS-TM-MIB.snmpDTLSUDPDomain";
      container dtls {
        uses tls-transport;
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ietf-snmp-tsm.yang

```
submodule ietf-snmp-tsm {

  belongs-to ietf-snmp {
    prefix snmp;
  }

  include ietf-snmp-common;
  include ietf-snmp-target;
  include ietf-snmp-proxy;

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

  contact
    "WG Web:   <http://tools.ietf.org/wg/netmod/>
     WG List:  <mailto:netmod@ietf.org>


     WG Chair: Thomas Nadeau
               <mailto:tnadeau@lucidvision.com>

     WG Chair: Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>

     Editor:   Martin Bjorklund
               <mailto:mbj@tail-f.com>

     Editor:   Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>";

  description
    "This submodule contains a collection of YANG definitions for
     configuring the Transport Security Model (TSM) of SNMP.

     Copyright (c) 2014 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject
     to the license terms contained in, the Simplified BSD License
     set forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC 7407; see
     the RFC itself for full legal notices.";

  reference
    "RFC 5591: Transport Security Model for the
       Simple Network Management Protocol (SNMP)";

  revision 2014-12-10 {
    description
      "Initial revision.";
    reference
      "RFC 7407: A YANG Data Model for SNMP Configuration";
  }
```

```
  feature tsm {
    description
      "A server implements this feature if it supports the
      Transport Security Model for SNMP.";
    reference
      "RFC 5591: Transport Security Model for the
         Simple Network Management Protocol (SNMP)";
  }
  augment /snmp:snmp {
    if-feature tsm;
    container tsm {
      description
        "Configuration of the Transport Security Model.";

      leaf use-prefix {
        type boolean;
        default false;
        reference
          "RFC 5591: Transport Security Model for the Simple
             Network Management Protocol (SNMP).
             SNMP-TSM-MIB.snmpTsmConfigurationUsePrefix";
      }
    }
  }

  grouping tsm-target-params {
    container tsm {
      description
        "Transport-based security SNMPv3 parameters type.

         Represents snmpTargetParamsMPModel '3' and
         snmpTargetParamsSecurityModel '4'.";
      leaf security-name {
        type snmp:security-name;
        mandatory true;
        reference
          "RFC 3413: Simple Network Management Protocol (SNMP).
             Applications.
             SNMP-TARGET-MIB.snmpTargetParamsSecurityName";
      }
      leaf security-level {
        type snmp:security-level;
        mandatory true;
        reference
          "RFC 3413: Simple Network Management Protocol (SNMP).
             Applications.
             SNMP-TARGET-MIB.snmpTargetParamsSecurityLevel";
      }
    }
  }

  augment /snmp:snmp/snmp:target-params/snmp:params {
    if-feature tsm;
    case tsm {
      uses tsm-target-params;
    }
  }

}
```


Copyright© 2022 Ciena® Corporation

ietf-snmp-usm.yang

```
submodule ietf-snmp-usm {

  belongs-to ietf-snmp {
    prefix snmp;
  }

  import ietf-yang-types {
    prefix yang;
  }
  import ietf-netconf-acm {
    prefix nacm;
  }

  include ietf-snmp-common;
  include ietf-snmp-target;
  include ietf-snmp-proxy;

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

  contact
    "WG Web:   <http://tools.ietf.org/wg/netmod/>
     WG List:  <mailto:netmod@ietf.org>

     WG Chair: Thomas Nadeau
               <mailto:tnadeau@lucidvision.com>

     WG Chair: Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>

     Editor:   Martin Bjorklund
               <mailto:mbj@tail-f.com>

     Editor:   Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>";

  description
    "This submodule contains a collection of YANG definitions for
     configuring the User-based Security Model (USM) of SNMP.

     Copyright (c) 2014 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject
     to the license terms contained in, the Simplified BSD License
     set forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC 7407; see
     the RFC itself for full legal notices.";


  reference
    "RFC 3414: User-based Security Model (USM) for version 3 of the
       Simple Network Management Protocol (SNMPv3)";
```


Copyright© 2022 Ciena® Corporation

```
  revision 2014-12-10 {
    description
      "Initial revision.";
    reference
      "RFC 7407: A YANG Data Model for SNMP Configuration";
  }

  grouping key {
    leaf key {
      type yang:hex-string;
      mandatory true;
      nacm:default-deny-all;
      description
        "Localized key specified as a list of colon-specified
         hexadecimal octets.";
    }
  }

  grouping user-list {
    list user {
      key "name";

      reference
        "RFC 3414: User-based Security Model (USM) for version 3
           of the Simple Network Management Protocol (SNMPv3).
           SNMP-USER-BASED-SM-MIB.usmUserTable";

      leaf name {
        type snmp:identifier;
        reference
          "RFC 3414: User-based Security Model (USM) for version 3
             of the Simple Network Management Protocol (SNMPv3).
             SNMP-USER-BASED-SM-MIB.usmUserName";
      }
      container auth {
        presence "enables authentication";
        description
          "Enables authentication of the user.";
        choice protocol {
          mandatory true;
          reference
            "RFC 3414: User-based Security Model (USM) for version 3
               of the Simple Network Management Protocol (SNMPv3).
               SNMP-USER-BASED-SM-MIB.usmUserAuthProtocol";
          container md5 {
            uses key;
            reference
              "RFC 3414: User-based Security Model (USM) for
                 version 3 of the Simple Network Management Protocol
                 (SNMPv3).
                 SNMP-USER-BASED-SM-MIB.usmHMACMD5AuthProtocol";
          }
          container sha {
            uses key;
            reference
              "RFC 3414: User-based Security Model (USM) for
                 version 3 of the Simple Network Management Protocol
                 (SNMPv3).
                 SNMP-USER-BASED-SM-MIB.usmHMACSHAAuthProtocol";
          }
```


Copyright© 2022 Ciena® Corporation

```
      }
    }
    container priv {
      must "../auth" {
        error-message
          "when privacy (confidentiality) is used, "
        + "authentication must also be used";
      }
      presence "enables encryption";
      description
        "Enables encryption of SNMP messages.";

      choice protocol {
        mandatory true;
        reference
          "RFC 3414: User-based Security Model (USM) for version 3
             of the Simple Network Management Protocol (SNMPv3).
             SNMP-USER-BASED-SM-MIB.usmUserPrivProtocol";
        container des {
          uses key;
          reference
            "RFC 3414: User-based Security Model (USM) for
               version 3 of the Simple Network Management Protocol
               (SNMPv3).
               SNMP-USER-BASED-SM-MIB.usmDESPrivProtocol";
        }
        container aes {
          uses key;




          reference
            "RFC 3826: The Advanced Encryption Standard (AES)
               Cipher Algorithm in the SNMP User-based Security
               Model.
               SNMP-USM-AES-MIB.usmAesCfb128Protocol";
        }
      }
    }
  }
}

augment /snmp:snmp {

  container usm {
    description
      "Configuration of the User-based Security Model.";
    container local {
      uses user-list;
    }

    list remote {
      key "engine-id";

      leaf engine-id {
        type snmp:engine-id;
        reference
          "RFC 3414: User-based Security Model (USM) for version 3
             of the Simple Network Management Protocol (SNMPv3).
```


Copyright© 2022 Ciena® Corporation

```
                SNMP-USER-BASED-SM-MIB.usmUserEngineID";
        }

        uses user-list;
      }
    }
  }

  grouping usm-target-params {
    container usm {
      description
        "User-based SNMPv3 parameters type.

         Represents snmpTargetParamsMPModel '3' and
         snmpTargetParamsSecurityModel '3'.";
      leaf user-name {
        type snmp:security-name;
        mandatory true;


        reference
          "RFC 3413: Simple Network Management Protocol (SNMP).
             Applications.
             SNMP-TARGET-MIB.snmpTargetParamsSecurityName";
      }
      leaf security-level {
        type snmp:security-level;
        mandatory true;
        reference
          "RFC 3413: Simple Network Management Protocol (SNMP).
             Applications.
             SNMP-TARGET-MIB.snmpTargetParamsSecurityLevel";
      }
    }
  }

  augment /snmp:snmp/snmp:target-params/snmp:params {
    case usm {
      uses usm-target-params;
    }
  }

}
```


Copyright© 2022 Ciena® Corporation

ietf-snmp-vacm.yang

```
submodule ietf-snmp-vacm {

  belongs-to ietf-snmp {
    prefix snmp;
  }

  include ietf-snmp-common;

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

  contact
    "WG Web:   <http://tools.ietf.org/wg/netmod/>
     WG List:  <mailto:netmod@ietf.org>

     WG Chair: Thomas Nadeau
               <mailto:tnadeau@lucidvision.com>

     WG Chair: Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>

     Editor:   Martin Bjorklund
               <mailto:mbj@tail-f.com>

     Editor:   Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>";

  description
    "This submodule contains a collection of YANG definitions
     for configuring the View-based Access Control Model (VACM)
     of SNMP.

     Copyright (c) 2014 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject
     to the license terms contained in, the Simplified BSD License
     set forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC 7407; see
     the RFC itself for full legal notices.";

  reference
    "RFC 3415: View-based Access Control Model (VACM) for the
       Simple Network Management Protocol (SNMP)";

  revision 2014-12-10 {
    description
      "Initial revision.";
    reference
      "RFC 7407: A YANG Data Model for SNMP Configuration";
  }

  typedef view-name {
    type snmp:identifier;
```

```
    description
      "The view-name type represents an SNMP VACM view name.";
  }

  typedef group-name {
    type snmp:identifier;
    description
      "The group-name type represents an SNMP VACM group name.";
  }

  augment /snmp:snmp {

    container vacm {
      description
        "Configuration of the View-based Access Control Model.";

      list group {
        key name;
        description
          "VACM groups.

           This data model has a different structure than the MIB.
           Groups are explicitly defined in this list, and group
           members are defined in the 'member' list (mapped to
           vacmSecurityToGroupTable), and access for the group is
           defined in the 'access' list (mapped to
           vacmAccessTable).";
        reference
          "RFC 3415: View-based Access Control Model (VACM) for the
             Simple Network Management Protocol (SNMP).
             SNMP-VIEW-BASED-ACM-MIB.vacmSecurityToGroupTable
             SNMP-VIEW-BASED-ACM-MIB.vacmAccessTable";

        leaf name {
          type group-name;
          description
            "The name of this VACM group.";
          reference
            "RFC 3415: View-based Access Control Model (VACM) for the
               Simple Network Management Protocol (SNMP).
               SNMP-VIEW-BASED-ACM-MIB.vacmGroupName";
        }

        list member {
          key "security-name";
          description
            "A member of this VACM group.

             A specific combination of security-name and
             security-model MUST NOT be present in more than
             one group.";
          reference
            "RFC 3415: View-based Access Control Model (VACM) for the
               Simple Network Management Protocol (SNMP).
               SNMP-VIEW-BASED-ACM-MIB.vacmSecurityToGroupTable";

          leaf security-name {
            type snmp:security-name;
            description
              "The securityName of a group member.";
```


Copyright© 2022 Ciena® Corporation

```
      reference
        "RFC 3415: View-based Access Control Model (VACM) for
           the Simple Network Management Protocol (SNMP).
           SNMP-VIEW-BASED-ACM-MIB.vacmSecurityName";
    }

    leaf-list security-model {
      type snmp:security-model;
      min-elements 1;
      description
        "The security models under which this security-name
         is a member of this group.";
      reference
        "RFC 3415: View-based Access Control Model (VACM) for
           the Simple Network Management Protocol (SNMP).
           SNMP-VIEW-BASED-ACM-MIB.vacmSecurityModel";
    }
  }

  list access {
    key "context security-model security-level";
    description
      "Definition of access right for groups.";
    reference
      "RFC 3415: View-based Access Control Model (VACM) for
         the Simple Network Management Protocol (SNMP).
         SNMP-VIEW-BASED-ACM-MIB.vacmAccessTable";

    leaf context {
      type snmp:context-name;
      description
        "The context (prefix) under which the access rights
         apply.";
      reference
        "RFC 3415: View-based Access Control Model (VACM) for
           the Simple Network Management Protocol (SNMP).
           SNMP-VIEW-BASED-ACM-MIB.vacmAccessContextPrefix";
    }

    leaf context-match {
      type enumeration {
        enum exact { value 1; }
        enum prefix { value 2; }
      }
      default exact;
      reference
        "RFC 3415: View-based Access Control Model (VACM) for
           the Simple Network Management Protocol (SNMP).
           SNMP-VIEW-BASED-ACM-MIB.vacmAccessContextMatch";
    }

    leaf security-model {
      type snmp:security-model-or-any;
      description
        "The security model under which the access rights
        apply.";
      reference
        "RFC 3415: View-based Access Control Model (VACM) for
           the Simple Network Management Protocol (SNMP).
           SNMP-VIEW-BASED-ACM-MIB.vacmAccessSecurityModel";
```


Copyright© 2022 Ciena® Corporation

```
}

leaf security-level {
  type snmp:security-level;
  description
    "The minimum security level under which the access
     rights apply.";
  reference
    "RFC 3415: View-based Access Control Model (VACM) for
       the Simple Network Management Protocol (SNMP).
       SNMP-VIEW-BASED-ACM-MIB.vacmAccessSecurityLevel";
}

leaf read-view {
  type view-name;
  description
    "The name of the MIB view of the SNMP context
     authorizing read access.  If this leaf does not
     exist in a configuration, it maps to a zero-length
     vacmAccessReadViewName.

     Implementations MAY restrict the values of this
     leaf to be one of the available values of
     /snmp/vacm/view/name in a valid configuration.";
  reference
    "RFC 3415: View-based Access Control Model (VACM) for
       the Simple Network Management Protocol (SNMP).
       SNMP-VIEW-BASED-ACM-MIB.vacmAccessReadViewName";
}

leaf write-view {
  type view-name;
  description
    "The name of the MIB view of the SNMP context
     authorizing write access.  If this leaf does not
     exist in a configuration, it maps to a zero-length
     vacmAccessWriteViewName.

     Implementations MAY restrict the values of this
     leaf to be one of the available values of
     /snmp/vacm/view/name in a valid configuration.";
  reference
    "RFC 3415: View-based Access Control Model (VACM) for
       the Simple Network Management Protocol (SNMP).
       SNMP-VIEW-BASED-ACM-MIB.vacmAccessWriteViewName";
}

leaf notify-view {
  type view-name;
  description
    "The name of the MIB view of the SNMP context
     authorizing notify access.  If this leaf does not
     exist in a configuration, it maps to a zero-length
     vacmAccessNotifyViewName.

     Implementations MAY restrict the values of this
     leaf to be one of the available values of
     /snmp/vacm/view/name in a valid configuration.";
  reference
    "RFC 3415: View-based Access Control Model (VACM) for
```


Copyright© 2022 Ciena® Corporation

```
                    the Simple Network Management Protocol (SNMP).
                    SNMP-VIEW-BASED-ACM-MIB.vacmAccessNotifyViewName";
          }
        }
      }

      list view {
        key name;
        description
          "Definition of MIB views.";
        reference
          "RFC 3415: View-based Access Control Model (VACM) for
             the Simple Network Management Protocol (SNMP).
             SNMP-VIEW-BASED-ACM-MIB.vacmViewTreeFamilyTable";

        leaf name {
          type view-name;
          description
            "The name of this VACM MIB view.";

          reference
            "RFC 3415: View-based Access Control Model (VACM) for
               the Simple Network Management Protocol (SNMP).
               SNMP-VIEW-BASED-ACM-MIB.vacmViewTreeFamilyName";
        }

        leaf-list include {
          type snmp:wildcard-object-identifier;
          description
            "A family of subtrees included in this MIB view.";
          reference
              "RFC 3415: View-based Access Control Model (VACM) for
                 the Simple Network Management Protocol (SNMP).
                 SNMP-VIEW-BASED-ACM-MIB.vacmViewTreeFamilySubtree
                 SNMP-VIEW-BASED-ACM-MIB.vacmViewTreeFamilyMask
                 SNMP-VIEW-BASED-ACM-MIB.vacmViewTreeFamilyType";
        }

        leaf-list exclude {
          type snmp:wildcard-object-identifier;
          description
            "A family of subtrees excluded from this MIB view.";
          reference
            "RFC 3415: View-based Access Control Model (VACM) for
               the Simple Network Management Protocol (SNMP).
               SNMP-VIEW-BASED-ACM-MIB.vacmViewTreeFamilySubtree
               SNMP-VIEW-BASED-ACM-MIB.vacmViewTreeFamilyMask
               SNMP-VIEW-BASED-ACM-MIB.vacmViewTreeFamilyType";
        }
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ietf-snmp.yang

```
module ietf-snmp {

  namespace "urn:ietf:params:xml:ns:yang:ietf-snmp";
  prefix snmp;

  include ietf-snmp-common {
    revision-date 2014-12-10;
  }
  include ietf-snmp-engine {
    revision-date 2014-12-10;
  }
  include ietf-snmp-target {
    revision-date 2014-12-10;
  }
  include ietf-snmp-notification {
    revision-date 2014-12-10;
  }
  include ietf-snmp-proxy {
    revision-date 2014-12-10;
  }
  include ietf-snmp-community {
    revision-date 2014-12-10;
  }
  include ietf-snmp-usm {
    revision-date 2014-12-10;
  }
  include ietf-snmp-tsm {
    revision-date 2014-12-10;
  }
  include ietf-snmp-vacm {
    revision-date 2014-12-10;
  }
  include ietf-snmp-tls {
    revision-date 2014-12-10;
  }
  include ietf-snmp-ssh {
    revision-date 2014-12-10;
  }

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

  contact
    "WG Web:   <http://tools.ietf.org/wg/netmod/>
     WG List:  <mailto:netmod@ietf.org>

     WG Chair: Thomas Nadeau
               <mailto:tnadeau@lucidvision.com>

     WG Chair: Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>

     Editor:   Martin Bjorklund
               <mailto:mbj@tail-f.com>

     Editor:   Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>";
  description
```


Copyright© 2022 Ciena® Corporation

```
    "This module contains a collection of YANG definitions for
     configuring SNMP engines.

     Copyright (c) 2014 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject
     to the license terms contained in, the Simplified BSD License
     set forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC 7407; see
     the RFC itself for full legal notices.";

  revision 2014-12-10 {
    description
      "Initial revision.";
    reference
      "RFC 7407: A YANG Data Model for SNMP Configuration";
  }

}
```

ietf-syslog-types.yang

```
module ietf-syslog-types {
  namespace "urn:ietf:params:xml:ns:yang:ietf-syslog-types";
  prefix syslogtypes;

  organization "IETF NETMOD (NETCONF Data Modeling Language) Working
                Group";
  contact
    "WG Web:   <http://tools.ietf.org/wg/netmod/>
     WG List:  <mailto:netmod@ietf.org>

     WG Chair: Lou Berger
               <mailto:lberger@labn.net>

     WG Chair: Jurgen Schonwalder
               <mailto:j.schoenwaelder@jacobs-university.de>

     WG Chair: Kent Watsen
               <mailto:kwatsen@juniper.net>

     Editor:   Kiran Agrahara Sreenivasa
               <mailto:kkoushik@cisco.com>

     Editor:   Clyde Wildes
               <mailto:cwildes@cisco.com>";
  description
    "This module contains a collection of YANG type definitions for
     SYSLOG.

     Copyright (c) 2016 IETF Trust and the persons identified as
     authors of the code. All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject to
     the license terms contained in, the Simplified BSD License set
     forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
     NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'MAY', and
     'OPTIONAL' in the module text are to be interpreted as described
     in RFC 2119 (http://tools.ietf.org/html/rfc2119).

     This version of this YANG module is part of RFC XXXX
     (http://tools.ietf.org/html/rfcXXXX); see the RFC itself for
     full legal notices.";

  reference
    "RFC 5424: The Syslog Protocol";

  revision 2016-05-10 {
    description
      "Initial Revision";
    reference
      "RFC XXXX: SYSLOG YANG Model";
  }

  typedef severity {
```


Copyright© 2022 Ciena® Corporation

```
    type enumeration {
      enum "emergency" {
        value 0;
        description
          "Emergency Level Msg";
      }
      enum "alert" {
        value 1;
        description
          "Alert Level Msg";
      }
      enum "critical" {
        value 2;
        description
          "Critical Level Msg";
      }
      enum "error" {
        value 3;
        description
          "Error Level Msg";
      }
      enum "warning" {
        value 4;
        description
          "Warning Level Msg";
      }
      enum "notice" {
        value 5;
        description
          "Notification Level Msg";
      }
      enum "info" {
        value 6;
        description
          "Informational Level Msg";
      }
      enum "debug" {
        value 7;
        description
          "Debugging Level Msg";
      }
    }
    description
      "The definitions for Syslog message severity as per RFC 5424.";
  }

  identity syslog-facility {
    description
      "This identity is used as a base for all syslog facilities as
       per RFC 5424.";
  }

  identity kern {
    base syslog-facility;
    description
      "The facility for kernel messages (0) as defined in RFC 5424.";
  }

  identity user {
    base syslog-facility;
```


Copyright© 2022 Ciena® Corporation

```
    description
      "The facility for user-level messages (1) as defined in RFC 5424.";
  }

  identity mail {
    base syslog-facility;
    description
      "The facility for the mail system (2) as defined in RFC 5424.";
  }

  identity daemon {
    base syslog-facility;
    description
      "The facility for the system daemons (3) as defined in RFC 5424.";
   }

  identity auth {
    base syslog-facility;
    description
      "The facility for security/authorization messages (4) as defined
       in RFC 5424.";
  }

  identity syslog {
    base syslog-facility;
    description
      "The facility for messages generated internally by syslogd
       facility (5)as defined in RFC 5424.";
  }

  identity lpr {
    base syslog-facility;
    description
      "The facility for the line printer subsystem (6) as defined in
       RFC 5424.";
  }

  identity news {
    base syslog-facility;
    description
      "The facility for the network news subsystem (7) as defined in
       RFC 5424.";
  }

  identity uucp {
    base syslog-facility;
    description
      "The facility for the UUCP subsystem (8) as defined in RFC 5424.";
  }

  identity cron {
    base syslog-facility;
    description
      "The facility for the clock daemon (9) as defined in RFC 5424.";
  }

  identity authpriv {
    base syslog-facility;
    description
      "The facility for privileged security/authorization messages (10)
```


Copyright© 2022 Ciena® Corporation

```
      as defined in RFC 5424.";
  }

  identity ftp {
    base syslog-facility;
    description
      "The facility for the FTP daemon (11) as defined in RFC 5424.";
  }

  identity ntp {
    base syslog-facility;
    description
      "The facility for the NTP subsystem (12) as defined in RFC 5424.";
  }

  identity audit {
    base syslog-facility;
    description
      "The facility for log audit messages (13) as defined in RFC 5424.";
  }

  identity console {
    base syslog-facility;
    description
      "The facility for log alert messages (14) as defined in RFC 5424.";
  }

  identity cron2 {
    base syslog-facility;
    description
      "The facility for the second clock daemon (15) as defined in
       RFC 5424.";
  }

  identity local0 {
    base syslog-facility;
    description
      "The facility for local use 0 messages (16) as defined in
       RFC 5424.";
  }

  identity local1 {
    base syslog-facility;
    description
      "The facility for local use 1 messages (17) as defined in
       RFC 5424.";
  }

  identity local2 {
    base syslog-facility;
    description
      "The facility for local use 2 messages (18) as defined in
       RFC 5424.";
  }

  identity local3 {
    base syslog-facility;
    description
      "The facility for local use 3 messages (19) as defined in
       RFC 5424.";
```


Copyright© 2022 Ciena® Corporation

```
  }

  identity local4 {
    base syslog-facility;
    description
      "The facility for local use 4 messages (20) as defined in
       RFC 5424.";
  }

  identity local5 {
    base syslog-facility;
    description
      "The facility for local use 5 messages (21) as defined in
       RFC 5424.";
  }

  identity local6 {
    base syslog-facility;
    description
      "The facility for local use 6 messages (22) as defined in
       RFC 5424.";
  }

  identity local7 {
    base syslog-facility;
    description
      "The facility for local use 7 messages (23) as defined in
       RFC 5424.";
  }
}
```


Copyright© 2022 Ciena® Corporation

ietf-syslog.yang

```
module ietf-syslog {
  namespace "urn:ietf:params:xml:ns:yang:ietf-syslog";
  prefix syslog;

  import ietf-inet-types {
    prefix inet;
  }

  import ietf-interfaces {
    prefix if;
  }

  import ietf-syslog-types {
    prefix syslogtypes;
  }

  organization "IETF NETMOD (NETCONF Data Modeling Language)
  Working Group";
  contact
    "WG Web:   <http://tools.ietf.org/wg/netmod/>
     WG List:  <mailto:netmod@ietf.org>

     WG Chair: Lou Berger
               <mailto:lberger@labn.net>

     WG Chair: Jurgen Schonwalder
               <mailto:j.schoenwaelder@jacobs-university.de>

     WG Chair: Kent Watsen
               <mailto:kwatsen@juniper.net>

     Editor:   Kiran Agrahara Sreenivasa
               <mailto:kkoushik@cisco.com>

     Editor:   Clyde Wildes
               <mailto:cwildes@cisco.com>";
  description
    "This module contains a collection of YANG definitions
     for Syslog configuration.

     Copyright (c) 2015 IETF Trust and the persons identified as
     authors of the code. All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject to
     the license terms contained in, the Simplified BSD License set
     forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
     NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'MAY', and
     'OPTIONAL' in the module text are to be interpreted as described
     in RFC 2119 (http://tools.ietf.org/html/rfc2119).

     This version of this YANG module is part of RFC XXXX
     (http://tools.ietf.org/html/rfcXXXX); see the RFC itself for
     full legal notices.";
```


Copyright© 2022 Ciena® Corporation

```
  reference
    "RFC 5424: The Syslog Protocol
     RFC 5848: Signed Syslog Messages";

  revision 2016-03-20{
    description
      "Initial Revision";
    reference
      "RFC XXXX: SYSLOG YANG Model";
  }

  feature console-action {
    description
      "This feature indicates that logging to the console is
       supported.";
  }

  feature buffer-action {
    description
      "This feature indicates that logging to an in-memory
       buffer is supported.";
  }

  feature buffer-limit-bytes {
    description
      "This feature indicates that local memory logging buffers
       are limited in size using a limit expressed in bytes.";
  }

  feature buffer-limit-messages {
    description
      "This feature indicates that local memory logging buffers
       are limited in size using a limit expressed in number
       of messages.";
  }

  feature file-limit-size {
    description
      "This feature indicates that file logging resources
       are managed using size and number limits.";
  }

  feature file-limit-duration {
    description
      "This feature indicates that file logging resources
       are managed using time based limits.";
  }

  feature terminal-action {
    description
      "This feature indicates that logging to a terminal
       is supported.";
  }

  feature terminal-facility-user-logging-config {
    description
      "This feature represents the ability to adjust
       log message settings for individual terminal
       devices.";
```


Copyright© 2022 Ciena® Corporation

```
}

feature session-action {
  description
    "This feature indicates that logging to user
     CLI session is supported.";
}

feature session-facility-user-logging-config {
  description
    "This feature represents the ability to adjust
     log message settings for individual user CLI
     sessions.";
}

feature selector-sevop-config {
  description
    "This feature represents the ability to select messages
     using the additional operators equal to, or not equal to
     when comparing the Syslog message severity.";
}

feature selector-match-config {
  description
    "This feature represents the ability to select messages based
     on a Posix 1003.2 regular expression pattern match.";
}

feature structured-data-config {
  description
    "This feature represents the ability to log messages
     in structured-data format as per RFC 5424.";
}

feature signed-messages-config {
  description
    "This feature represents the ability to configure signed
     syslog messages according to RFC 5848.";
}

grouping syslog-severity {
  description
    "This grouping defines the Syslog severity which is used to
     select log messages.";
  leaf severity {
    type union {
      type syslogtypes:severity;
      type enumeration {
        enum all {
          value -1;
          description
            "This enum describes the case where all severities
             are selected.";
        }
        enum none {
          value -2;
          description
            "This enum describes the case where no severities
             are selected.";
        }
```


Copyright© 2022 Ciena® Corporation

```
        }
      }
      mandatory true;
      description
        "This leaf specifies the Syslog message severity. When
         severity is specified, the default severity comparison
         is all messages of the specified severity and greater are
         selected. 'all' is a special case which means all severities
         are selected. 'none' is a special case which means that
         no selection should occur or disable this filter.";
    }
    leaf severity-operator {
      when '../severity != "all" and
           ../severity != "none"' {
        description
          "The severity-operator is not applicable for severity 'all' or
          severity 'none'";
      }
      if-feature selector-sevop-config;
      type enumeration {
        enum equals-or-higher {
          description
            "This enum specifies all messages of the specified
             severity and higher are logged according to the
             given log-action";
        }
        enum equals {
          description
            "This enum specifies all messages that are for
             the specified severity are logged according to the
             given log-action";
        }
        enum not-equals {
          description
            "This enum specifies all messages that are not for
             the specified severity are logged according to the
             given log-action";
        }
      }
      default equals-or-higher;
      description
        "This leaf describes the option to specify how the
         severity comparison is performed.";
    }
  }

  grouping syslog-selector {
    description
      "This grouping defines a Syslog selector which is used to
       select log messages for the log-action (buffer, file,
       etc). Choose one of the following:
         no-log-facility
         log-facility [<facility> <severity>...]";
    container log-selector {
      description
        "This container describes the log selector parameters
         for Syslog.";
      choice selector-facility {
        mandatory true;
        description
```


Copyright© 2022 Ciena® Corporation

```
          "This choice describes the option to specify no
           facilities, or a specific facility which can be
           all for all facilities.";
        case no-log-facility {
          description
            "This case specifies no facilities will match when
             comparing the Syslog message facility. This is a
             method that can be used to effectively disable a
             particular log-action (buffer, file, etc).";
          leaf no-facilities {
            type empty;
            description
            "This leaf specifies that no facilities are selected
             for this log-action.";
          }
        }
        case log-facility {
          description
            "This case specifies one or more specified facilities
             will match when comparing the Syslog message facility.";
          list log-facility {
            key facility;
            description
              "This list describes a collection of Syslog
               facilities and severities.";
            leaf facility {
              type union {
                type identityref {
                  base syslogtypes:syslog-facility;
                }
                type enumeration {
                  enum all {
                    description
                      "This enum describes the case where all
                       facilities are requested.";
                  }
                }
              }
              description
                "The leaf uniquely identifies a Syslog facility.";
            }
            uses syslog-severity;
          }
        }
      }
      leaf pattern-match {
        if-feature selector-match-config;
        type string;
        description
          "This leaf desribes a Posix 1003.2 regular expression
           string that can be used to select a Syslog message for
           logging. The match is performed on the RFC 5424
           SYSLOG-MSG field.";
      }
    }
  }

  grouping syslog-structured-data {
    description
      "This grouping defines the Syslog structured data option
```


Copyright© 2022 Ciena® Corporation

```
       which is used to select the format used to write log
       messages.";
    leaf structured-data {
      if-feature structured-data-config;
      type boolean;
      default false;
      description
        "This leaf describes how log messages are written to
         the log file. If true, messages will be written
         with one or more STRUCTURED-DATA elements as per
         RFC5424; if false, messages will be written with
         STRUCTURED-DATA = NILVALUE.";
    }
  }

  container syslog {
    description
      "This container describes the configuration parameters for
       Syslog.";
    container log-actions {
      description
        "This container describes the log-action parameters
         for Syslog.";
      container console {
        if-feature console-action;
        presence "Enables logging console configuration";
        description
          "This container describes the configuration parameters for
           console logging.";
        uses syslog-selector;
      }
      container buffer {
        if-feature buffer-action;
        description
          "This container describes the configuration parameters for
           local memory buffer logging. The buffer is circular in
           nature, so newer messages overwrite older messages after
           the buffer is filled. The method used to read syslog messages
           from the buffer is supplied by the local implementation.";
        list log-buffer {
          key name;
          description
            "This list describes a collection of local logging
             memory buffers. If buffer size limits are not supplied, it
             is assumed that the local implementation defined
             limits will be used.";
          leaf name {
            type string;
            description
              "This leaf specifies the name of the log buffer.";
          }
          uses syslog-selector;
          leaf buffer-size-bytes {
            if-feature buffer-limit-bytes;
            type uint64;
            units "bytes";
            description
              "This leaf configures the amount of memory (in bytes) that
               will be dedicated to the local memory logging buffer.
               The default value varies by implementation.";
```


Copyright© 2022 Ciena® Corporation

```
      }
      leaf buffer-size-messages {
        if-feature buffer-limit-messages;
        type uint64;
        units "log messages";
        description
          "This leaf configures the amount number of log messages that
           can be stored in the local memory logging buffer. The
           default value varies by implementation.";
      }
      uses syslog-structured-data;
    }
  }
  container file {
    description
      "This container describes the configuration parameters for
       file logging. If file-archive limits are not supplied, it
       is assumed that the local implementation defined limits will
       be used.";
    list log-file {
      key "name";
      description
        "This list describes a collection of local logging
         files.";
      leaf name {
        type inet:uri {
          pattern 'file:.*';
        }
        description
          "This leaf specifies the name of the log file which
           MUST use the uri scheme file:.";
      }
      uses syslog-selector;
      uses syslog-structured-data;
      container file-archive {
        description
          "This container describes the configuration
           parameters for log file archiving.";
        leaf number-of-files {
          if-feature file-limit-size;
          type uint32;
          description
            "This leaf specifies the maximum number of log
             files retained. Specify 1 for implementations
             that only support one log file.";
        }
        leaf max-file-size {
          if-feature file-limit-size;
          type uint64;
          units "megabytes";
          description
            "This leaf specifies the maximum log file size.";
        }
        leaf rollover {
          if-feature file-limit-duration;
          type uint32;
          units "minutes";
          description
            "This leaf specifies the length of time that log
             events should be written to a specific log file.
```


Copyright© 2022 Ciena® Corporation

```
              Log events that arrive after the rollover period
              cause the current log file to be closed and a new
              log file to be opened.";
          }
          leaf retention {
            if-feature file-limit-duration;
            type uint16;
            units "hours";
            description
              "This leaf specifies the length of time that
               completed/closed log event files should be stored
               in the file system before they are deleted.";
          }
        }
      }
    }
    container remote {
      description
        "This container describes the configuration parameters for
         remote logging.";
      list destination {
        key "name";
        description
          "This list describes a collection of remote logging
           destinations.";
        leaf name {
          type string;
          description
            "An arbitrary name for the endpoint to connect to.";
        }
        choice transport {
          mandatory true;
          description
            "This choice describes the transport option.";
          case tcp {
            container tcp {
              description
                "This container describes the TCP transport
                 options.";
              reference
                "RFC 6587: Transmission of Syslog Messages over TCP";
              leaf address {
                type inet:host;
                description
                  "The leaf uniquely specifies the address of
                   the remote host. One of the following must
                   be specified: an ipv4 address, an ipv6
                   address, or a host name.";
              }
              leaf port {
                type inet:port-number;
                default 514;
                description
                  "This leaf specifies the port number used to
                   deliver messages to the remote server.";
              }
            }
          }
          case udp {
            container udp {
```


Copyright© 2022 Ciena® Corporation

```
        description
          "This container describes the UDP transport
           options.";
        reference
          "RFC 5426: Transmission of Syslog Messages over UDP";
        leaf address {
          type inet:host;
          description
            "The leaf uniquely specifies the address of
             the remote host. One of the following must be
             specified: an ipv4 address, an ipv6 address,
             or a host name.";
        }
        leaf port {
          type inet:port-number;
          default 514;
          description
            "This leaf specifies the port number used to
             deliver messages to the remote server.";
        }
      }
    }
    case tls {
      container tls {
        description
          "This container describes the TLS transport options.";
        reference
          "RFC 5425: Transport Layer Security (TLS) Transport
           Mapping for Syslog ";
        leaf address {
          type inet:host;
          description
            "The leaf uniquely specifies the address of
             the remote host. One of the following must be
             specified: an ipv4 address, an ipv6 address,
             or a host name.";
        }
        leaf port {
          type inet:port-number;
          default 6514;
          description
            "This leaf specifies the port number used to
             deliver messages to the remote server.";
        }
      }
    }
  }
  uses syslog-selector;
  leaf destination-facility {
    type identityref {
      base syslogtypes:syslog-facility;
    }
    default syslogtypes:local7;
    description
      "This leaf specifies the facility used in messages
       delivered to the remote server.";
  }
  leaf source-interface {
    type if:interface-ref;
    description
```


Copyright© 2022 Ciena® Corporation

```
      "This leaf sets the source interface for the remote
       Syslog server. Either the interface name or the
       interface IP address can be specified. If not set,
       messages sent to a remote syslog server will
       contain the IP address of the interface the syslog
       message uses to exit the network element";
  }
  uses syslog-structured-data;
  container syslog-sign {
    if-feature signed-messages-config;
    presence
      "If present, syslog-sign is activated.";
    description
      "This container describes the configuration
       parameters for signed syslog messages as described
       by RFC 5848.";
    reference
      "RFC 5848: Signed Syslog Messages";
    leaf cert-initial-repeat {
      type uint16;
      mandatory true;
      description
      "This leaf specifies the number of times each
       Certificate Block should be sent before the first
       message is sent.";
    }
    leaf cert-resend-delay {
      type uint16;
      mandatory true;
      description
        "This leaf specifies the maximum time delay in
         seconds until resending the Certificate Block.";
    }
    leaf cert-resend-count {
      type uint16;
      mandatory true;
      description
        "This leaf specifies the maximum number of other
         syslog messages to send until resending the
         Certificate Block.";
    }
    leaf sig-max-delay {
      type uint16;
      mandatory true;
      description
        "This leaf specifies when to generate a new
         Signature Block. If this many seconds have
         elapsed since the message with the first message
         number of the Signature Block was sent, a new
         Signature Block should be generated.";
    }
    leaf sig-number-resends {
      type uint16;
      mandatory true;
      description
        "This leaf specifies the number of times a
         Signature Block is resent. (It is recommended to
         select a value of greater than 0 in particular
         when the UDP transport [RFC5426] is used.).";
    }
```


Copyright© 2022 Ciena® Corporation

```
        leaf sig-resend-delay {
          type uint16;
          mandatory true;
          description
            "This leaf specifies when to send the next
             Signature Block transmission based on time. If
             this many seconds have elapsed since the previous
             sending of this Signature Block, resend it.";
        }
        leaf sig-resend-count {
          type uint16;
          mandatory true;
          description
            "This leaf specifies when to send the next
             Signature Block transmission based on a count.
             If this many other syslog messages have been sent
             since the previous sending of this Signature
             Block, resend it.";
        }
      }
    }
  }
  container terminal {
    if-feature terminal-action;
    description
      "This container describes the configuration parameters for
       the terminal logging configuration.";
    container all-terminals {
      presence "Enables logging to all terminals.";
      description
        "This container describes the configuration
         parameters for all terminals.";
      uses syslog-selector;
    }
    list terminal {
      if-feature terminal-facility-user-logging-config;
      key "name";
      description
        "This list describes a collection of ";
      leaf name {
        type string;
        description
          "This leaf uniquely describes a terminal which
           will receive log messages.";
      }
      uses syslog-selector;
    }
  }
  container session {
    if-feature session-action;
    description
      "This container describes the configuration parameters for
       user CLI session logging configuration.";
    container all-users {
      presence "Enables logging to all user sessions.";
      description
        "This container describes the configuration
         parameters for all users.";
      uses syslog-selector;
    }
```


Copyright© 2022 Ciena® Corporation

```
        list user {
          if-feature session-facility-user-logging-config;
          key "name";
          description
            "This list describes a collection of user names.";
          leaf name {
            type string;
            description
              "This leaf uniquely describes a user name which
               is the login name of the user whose session
               is to receive log messages.";
          }
          uses syslog-selector;
        }
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

ietf-system.yang

```
module ietf-system {
  namespace "urn:ietf:params:xml:ns:yang:ietf-system";
  prefix "sys";

  import ietf-yang-types {
    prefix yang;
  }

  import ietf-inet-types {
    prefix inet;
  }

  import ietf-netconf-acm {
    prefix nacm;
  }

  import iana-crypt-hash {
    prefix ianach;
  }

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";




  contact
    "WG Web:   <http://tools.ietf.org/wg/netmod/>
     WG List:  <mailto:netmod@ietf.org>

     WG Chair: Thomas Nadeau
               <mailto:tnadeau@lucidvision.com>

     WG Chair: Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>

     Editor:   Andy Bierman
               <mailto:andy@yumaworks.com>

     Editor:   Martin Bjorklund
               <mailto:mbj@tail-f.com>";

  description
    "This module contains a collection of YANG definitions for the
     configuration and identification of some common system
     properties within a device containing a NETCONF server.  This
     includes data node definitions for system identification,
     time-of-day management, user management, DNS resolver
     configuration, and some protocol operations for system
     management.

     Copyright (c) 2014 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
```

```
     without modification, is permitted pursuant to, and subject
     to the license terms contained in, the Simplified BSD License
     set forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC 7317; see
     the RFC itself for full legal notices.";

  revision 2017-05-25 {
    description
      "Initial revision.";
    reference
      "RFC 7317: A YANG Data Model for System Management";
  }




 /*
  * Typedefs
  */

  typedef timezone-name {
    type string;
    description
      "A time zone name as used by the Time Zone Database,
       sometimes referred to as the 'Olson Database'.

       The exact set of valid values is an implementation-specific
       matter.  Client discovery of the exact set of time zone names
       for a particular server is out of scope.";
    reference
      "RFC 6557: Procedures for Maintaining the Time Zone Database";
   }

  /*
   * Features
   */

  feature radius {
    description
      "Indicates that the device can be configured as a RADIUS
       client.";
    reference
      "RFC 2865: Remote Authentication Dial In User Service (RADIUS)";
  }

  feature authentication {
    description
      "Indicates that the device supports configuration of
       user authentication.";
  }

  feature local-users {
    if-feature authentication;
    description
      "Indicates that the device supports configuration of
       local user authentication.";
  }
```


Copyright© 2022 Ciena® Corporation

```
  feature radius-authentication {
    if-feature radius;
    if-feature authentication;
    description
      "Indicates that the device supports configuration of user
       authentication over RADIUS.";
    reference
      "RFC 2865: Remote Authentication Dial In User Service (RADIUS)
       RFC 5607: Remote Authentication Dial-In User Service (RADIUS)
                 Authorization for Network Access Server (NAS)
                 Management";
  }

  feature ntp {
    description
      "Indicates that the device can be configured to use one or
       more NTP servers to set the system date and time.";
  }

  feature ntp-udp-port {
    if-feature ntp;
    description
      "Indicates that the device supports the configuration of
       the UDP port for NTP servers.

       This is a 'feature', since many implementations do not support
       any port other than the default port.";
  }

  feature timezone-name {
    description
      "Indicates that the local time zone on the device
       can be configured to use the TZ database
       to set the time zone and manage daylight saving time.";
    reference
      "RFC 6557: Procedures for Maintaining the Time Zone Database";
  }

  feature dns-udp-tcp-port {
    description
      "Indicates that the device supports the configuration of
       the UDP and TCP port for DNS servers.

       This is a 'feature', since many implementations do not support
       any port other than the default port.";
  }


  /*
   * Identities
   */

  identity authentication-method {
```


Copyright© 2022 Ciena® Corporation

```
    description
      "Base identity for user authentication methods.";
  }

  identity radius {
    base authentication-method;
    description
      "Indicates user authentication using RADIUS.";
    reference
      "RFC 2865: Remote Authentication Dial In User Service (RADIUS)
       RFC 5607: Remote Authentication Dial-In User Service (RADIUS)
                 Authorization for Network Access Server (NAS)
                 Management";
  }

  identity local-users {
    base authentication-method;
    description
      "Indicates password-based authentication of locally
       configured users.";
  }

  identity radius-authentication-type {
    description
      "Base identity for RADIUS authentication types.";
  }

  identity radius-pap {
    base radius-authentication-type;
    description
      "The device requests Password Authentication Protocol (PAP)
       authentication from the RADIUS server.";
    reference
      "RFC 2865: Remote Authentication Dial In User Service (RADIUS)";
  }









  identity radius-chap {
    base radius-authentication-type;
    description
      "The device requests Challenge Handshake Authentication
       Protocol (CHAP) authentication from the RADIUS server.";
    reference
      "RFC 2865: Remote Authentication Dial In User Service (RADIUS)";
  }

  /*
   * Configuration data nodes
   */

  container system {
    description
      "System group configuration.";
```


Copyright© 2022 Ciena® Corporation

```
    leaf contact {
      type string;
      description
        "The administrator contact information for the system.

         A server implementation MAY map this leaf to the sysContact
         MIB object.  Such an implementation needs to use some
         mechanism to handle the differences in size and characters
         allowed between this leaf and sysContact.  The definition of
         such a mechanism is outside the scope of this document.";
      reference
        "RFC 3418: Management Information Base (MIB) for the
                   Simple Network Management Protocol (SNMP)
                   SNMPv2-MIB.sysContact";
    }
    leaf hostname {
      type inet:domain-name;
     description
       "The name of the host.  This name can be a single domain
        label or the fully qualified domain name of the host.";
    }
    leaf location {
      type string;
      description
        "The system location.

         A server implementation MAY map this leaf to the sysLocation
         MIB object.  Such an implementation needs to use some
         mechanism to handle the differences in size and characters
         allowed between this leaf and sysLocation.  The definition
         of such a mechanism is outside the scope of this document.";
      reference
        "RFC 3418: Management Information Base (MIB) for the
                   Simple Network Management Protocol (SNMP)
                   SNMPv2-MIB.sysLocation";
    }

    container clock {
      description
        "Configuration of the system date and time properties.";

      choice timezone {
        description
          "The system time zone information.";

        case timezone-name {
          if-feature timezone-name;
          leaf timezone-name {
            type timezone-name;
            description
              "The TZ database name to use for the system, such
               as 'Europe/Stockholm'.";
          }
        }
        case timezone-utc-offset {
          leaf timezone-utc-offset {
            type int16 {
              range "-1500 .. 1500";
            }
            units "minutes";
```


Copyright© 2022 Ciena® Corporation

```
          description
            "The number of minutes to add to UTC time to
             identify the time zone for this system.  For example,
             'UTC - 8:00 hours' would be represented as '-480'.
             Note that automatic daylight saving time adjustment
             is not provided if this object is used.";
        }
      }
    }
  }

  container ntp {
    if-feature ntp;
    presence
      "Enables the NTP client unless the 'enabled' leaf
       (which defaults to 'true') is set to 'false'";
    description
      "Configuration of the NTP client.";

    leaf enabled {
      type boolean;
      default true;
      description
        "Indicates that the system should attempt to
         synchronize the system clock with an NTP server
         from the 'ntp/server' list.";
    }
    list server {
      key name;
      description
        "List of NTP servers to use for system clock
         synchronization.  If '/system/ntp/enabled'
         is 'true', then the system will attempt to
         contact and utilize the specified NTP servers.";

      leaf name {
        type string;
        description
          "An arbitrary name for the NTP server.";
      }
      choice transport {
        mandatory true;
        description
          "The transport-protocol-specific parameters for this
           server.";

        case udp {
          container udp {
            description
              "Contains UDP-specific configuration parameters
               for NTP.";
            leaf address {
              type inet:host;
              mandatory true;
              description
                "The address of the NTP server.";
            }
            leaf port {
              if-feature ntp-udp-port;
              type inet:port-number;
```


Copyright© 2022 Ciena® Corporation

```
              default 123;
              description
                "The port number of the NTP server.";
            }
          }
        }
      }
      leaf association-type {
        type enumeration {
          enum server {
            description
              "Use client association mode.  This device
               will not provide synchronization to the
               configured NTP server.";
          }
          enum peer {
            description
              "Use symmetric active association mode.
               This device may provide synchronization
               to the configured NTP server.";
          }
          enum pool {
            description
              "Use client association mode with one or
               more of the NTP servers found by DNS
               resolution of the domain name given by
               the 'address' leaf.  This device will not
               provide synchronization to the servers.";
          }
        }
        default server;
        description
          "The desired association type for this NTP server.";
      }
      leaf iburst {
        type boolean;
        default false;
        description
          "Indicates whether this server should enable burst
           synchronization or not.";
      }
      leaf prefer {
        type boolean;
        default false;
        description
          "Indicates whether this server should be preferred
           or not.";
      }
    }
  }

  container dns-resolver {
    description
      "Configuration of the DNS resolver.";

    leaf-list search {
      type inet:domain-name;
      ordered-by user;
      description
        "An ordered list of domains to search when resolving
```


Copyright© 2022 Ciena® Corporation

```
      a host name.";
  }
  list server {
    key name;
    ordered-by user;
    description
      "List of the DNS servers that the resolver should query.

       When the resolver is invoked by a calling application, it
       sends the query to the first name server in this list.  If
       no response has been received within 'timeout' seconds,
       the resolver continues with the next server in the list.
       If no response is received from any server, the resolver
       continues with the first server again.  When the resolver
       has traversed the list 'attempts' times without receiving
       any response, it gives up and returns an error to the
       calling application.

       Implementations MAY limit the number of entries in this
       list.";

    leaf name {
      type string;
      description
        "An arbitrary name for the DNS server.";
    }
    choice transport {
      mandatory true;
      description
        "The transport-protocol-specific parameters for this
         server.";

      case udp-and-tcp {
        container udp-and-tcp {
          description
            "Contains UDP- and TCP-specific configuration
             parameters for DNS.";
          reference
            "RFC 1035: Domain Names - Implementation and
                       Specification
             RFC 5966: DNS Transport over TCP - Implementation
                       Requirements";

          leaf address {
            type inet:ip-address;
            mandatory true;
            description
              "The address of the DNS server.";
          }
          leaf port {
            if-feature dns-udp-tcp-port;
            type inet:port-number;
            default 53;
            description
              "The UDP and TCP port number of the DNS server.";
          }
        }
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

```
    container options {
      description
        "Resolver options.  The set of available options has been
         limited to those that are generally available across
         different resolver implementations and generally useful.";
      leaf timeout {
        type uint8 {
          range "1..max";
        }
        units "seconds";
        default "5";
        description
          "The amount of time the resolver will wait for a
           response from each remote name server before
           retrying the query via a different name server.";
      }
      leaf attempts {
        type uint8 {
          range "1..max";
        }
        default "2";
        description
          "The number of times the resolver will send a query to
           all of its name servers before giving up and returning
           an error to the calling application.";
      }
    }
  }



  container radius {
    if-feature radius;

    description
      "Configuration of the RADIUS client.";

    list server {
      key name;
      ordered-by user;
      description
        "List of RADIUS servers used by the device.

         When the RADIUS client is invoked by a calling
         application, it sends the query to the first server in
         this list.  If no response has been received within
         'timeout' seconds, the client continues with the next
         server in the list.  If no response is received from any
         server, the client continues with the first server again.
         When the client has traversed the list 'attempts' times
         without receiving any response, it gives up and returns an
         error to the calling application.";

      leaf name {
        type string;
        description
          "An arbitrary name for the RADIUS server.";
      }
      choice transport {
        mandatory true;
```


Copyright© 2022 Ciena® Corporation

```
      description
        "The transport-protocol-specific parameters for this
         server.";

      case udp {
        container udp {
          description
            "Contains UDP-specific configuration parameters
             for RADIUS.";
          leaf address {
            type inet:host;
            mandatory true;
            description
              "The address of the RADIUS server.";
          }




          leaf authentication-port {
            type inet:port-number;
            default "1812";
            description
              "The port number of the RADIUS server.";
          }
          leaf shared-secret {
            type string;
            mandatory true;
            nacm:default-deny-all;
            description
              "The shared secret, which is known to both the
               RADIUS client and server.";
            reference
              "RFC 2865: Remote Authentication Dial In User
                         Service (RADIUS)";
          }
        }
      }
    }
    leaf authentication-type {
      type identityref {
        base radius-authentication-type;
      }
      default radius-pap;
      description
        "The authentication type requested from the RADIUS
         server.";
    }
  }
  container options {
    description
      "RADIUS client options.";

    leaf timeout {
      type uint8 {
        range "1..max";
      }
      units "seconds";
      default "5";
      description
```


Copyright© 2022 Ciena® Corporation

```
          "The number of seconds the device will wait for a
           response from each RADIUS server before trying with a
           different server.";
      }



      leaf attempts {
        type uint8 {
          range "1..max";
        }
        default "2";
        description
          "The number of times the device will send a query to
           all of its RADIUS servers before giving up.";
      }
    }
  }

  container authentication {
    nacm:default-deny-write;
    if-feature authentication;

     description
       "The authentication configuration subtree.";

     leaf-list user-authentication-order {
       type identityref {
         base authentication-method;
       }
       must '(. != "sys:radius" or ../../radius/server)' {
         error-message
           "When 'radius' is used, a RADIUS server"
         + " must be configured.";
         description
           "When 'radius' is used as an authentication method,
            a RADIUS server must be configured.";
       }
       ordered-by user;

       description
         "When the device authenticates a user with a password,
          it tries the authentication methods in this leaf-list in
          order.  If authentication with one method fails, the next
          method is used.  If no method succeeds, the user is
          denied access.

          An empty user-authentication-order leaf-list still allows
          authentication of users using mechanisms that do not
          involve a password.

          If the 'radius-authentication' feature is advertised by
          the NETCONF server, the 'radius' identity can be added to
          this list.

          If the 'local-users' feature is advertised by the
          NETCONF server, the 'local-users' identity can be
          added to this list.";
     }
```


Copyright© 2022 Ciena® Corporation

```
list user {
  if-feature local-users;
  key name;
  description
    "The list of local users configured on this device.";

  leaf name {
    type string;
    description
     "The user name string identifying this entry.";
  }
  leaf password {
    type ianach:crypt-hash;
    description
      "The password for this entry.";
  }
  list authorized-key {
    key name;
    description
      "A list of public SSH keys for this user.  These keys
       are allowed for SSH authentication, as described in
       RFC 4253.";
    reference
      "RFC 4253: The Secure Shell (SSH) Transport Layer
                 Protocol";

    leaf name {
      type string;
      description
        "An arbitrary name for the SSH key.";
    }









    leaf algorithm {
      type string;
      mandatory true;
      description
        "The public key algorithm name for this SSH key.

         Valid values are the values in the IANA 'Secure Shell
         (SSH) Protocol Parameters' registry, Public Key
         Algorithm Names.";
      reference
        "IANA 'Secure Shell (SSH) Protocol Parameters'
         registry, Public Key Algorithm Names";
    }
    leaf key-data {
      type binary;
      mandatory true;
      description
```


Copyright© 2022 Ciena® Corporation

```
              "The binary public key data for this SSH key, as
               specified by RFC 4253, Section 6.6, i.e.:

                 string    certificate or public key format
                           identifier
                 byte[n]   key/certificate data.";
            reference
              "RFC 4253: The Secure Shell (SSH) Transport Layer
                         Protocol";
          }
        }
      }
    }
  }

  /*
   * Operational state data nodes
   */

  container system-state {
    config false;
    description
      "System group operational state.";

    container platform {
      description
        "Contains vendor-specific information for
         identifying the system platform and operating system.";
      reference
        "IEEE Std 1003.1-2008 - sys/utsname.h";

      leaf os-name {
        type string;
        description
          "The name of the operating system in use -
           for example, 'Linux'.";
        reference
          "IEEE Std 1003.1-2008 - utsname.sysname";
      }
      leaf os-release {
        type string;
        description
          "The current release level of the operating
           system in use.  This string MAY indicate
           the OS source code revision.";
        reference
          "IEEE Std 1003.1-2008 - utsname.release";
      }
      leaf os-version {
        type string;
        description
          "The current version level of the operating
           system in use.  This string MAY indicate
           the specific OS build date and target variant
           information.";
        reference
          "IEEE Std 1003.1-2008 - utsname.version";
      }
      leaf machine {
        type string;
```


Copyright© 2022 Ciena® Corporation

```
        description
          "A vendor-specific identifier string representing
           the hardware in use.";
        reference
          "IEEE Std 1003.1-2008 - utsname.machine";
      }
    }

    container clock {
      description
        "Monitoring of the system date and time properties.";

      leaf current-datetime {
        type yang:date-and-time;
        description
          "The current system date and time.";
      }


      leaf boot-datetime {
        type yang:date-and-time;
        description
          "The system date and time when the system last restarted.";
      }
    }
  }

  rpc set-current-datetime {
    nacm:default-deny-all;
    description
      "Set the /system-state/clock/current-datetime leaf
       to the specified value.

       If the system is using NTP (i.e., /system/ntp/enabled
       is set to 'true'), then this operation will fail with
       error-tag 'operation-failed' and error-app-tag value of
       'ntp-active'.";
    input {
      leaf current-datetime {
        type yang:date-and-time;
        mandatory true;
        description
          "The current system date and time.";
      }
    }
  }

  rpc system-restart {
    nacm:default-deny-all;
    description
      "Request that the entire system be restarted immediately.
       A server SHOULD send an rpc reply to the client before
       restarting the system.";
  }

  rpc system-shutdown {
    nacm:default-deny-all;
    description
      "Request that the entire system be shut down immediately.
       A server SHOULD send an rpc reply to the client before
```


Copyright© 2022 Ciena® Corporation

```
        shutting down the system.";
  }

}
```


Copyright© 2022 Ciena® Corporation

ietf-x509-cert-to-name.yang

```
module ietf-x509-cert-to-name {

  namespace "urn:ietf:params:xml:ns:yang:ietf-x509-cert-to-name";
  prefix x509c2n;

  import ietf-yang-types {
    prefix yang;
  }

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

  contact
    "WG Web:   <http://tools.ietf.org/wg/netmod/>
     WG List:  <mailto:netmod@ietf.org>

     WG Chair: Thomas Nadeau
               <mailto:tnadeau@lucidvision.com>

     WG Chair: Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>

     Editor:   Martin Bjorklund
               <mailto:mbj@tail-f.com>

     Editor:   Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>";

  description
    "This module contains a collection of YANG definitions for
     extracting a name from an X.509 certificate.
     The algorithm used to extract a name from an X.509 certificate
     was first defined in RFC 6353.

     Copyright (c) 2014 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject
     to the license terms contained in, the Simplified BSD License
     set forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC 7407; see
     the RFC itself for full legal notices.";

  reference
    "RFC 6353: Transport Layer Security (TLS) Transport Model for
       the Simple Network Management Protocol (SNMP)";

  revision 2014-12-10 {
    description
      "Initial revision.";
    reference
      "RFC 7407: A YANG Data Model for SNMP Configuration";

  }
```


Copyright© 2022 Ciena® Corporation

```
typedef tls-fingerprint {
  type yang:hex-string {
    pattern '([0-9a-fA-F]){2}(:([0-9a-fA-F]){2}){0,254}';
  }
  description
   "A fingerprint value that can be used to uniquely reference
    other data of potentially arbitrary length.

    A tls-fingerprint value is composed of a 1-octet hashing
    algorithm identifier followed by the fingerprint value.  The
    first octet value identifying the hashing algorithm is taken
    from the IANA 'TLS HashAlgorithm Registry' (RFC 5246).  The
    remaining octets are filled using the results of the hashing
    algorithm.";
  reference
   "RFC 6353: Transport Layer Security (TLS) Transport Model
      for the Simple Network Management Protocol (SNMP).
      SNMP-TLS-TM-MIB.SnmpTLSFingerprint";
}

/* Identities */

identity cert-to-name {
  description
    "Base identity for algorithms to derive a name from a
     certificate.";
}

identity specified {
  base cert-to-name;
  description
    "Directly specifies the name to be used for the certificate.
     The value of the leaf 'name' in the cert-to-name list is
     used.";
  reference
    "RFC 6353: Transport Layer Security (TLS) Transport Model
       for the Simple Network Management Protocol (SNMP).
       SNMP-TLS-TM-MIB.snmpTlstmCertSpecified";
}

identity san-rfc822-name {
  base cert-to-name;
  description
    "Maps a subjectAltName's rfc822Name to a name.  The local part
     of the rfc822Name is passed unaltered, but the host-part of
     the name must be passed in lowercase.  For example, the
     rfc822Name field FooBar@Example.COM is mapped to name
     FooBar@example.com.";
  reference
    "RFC 6353: Transport Layer Security (TLS) Transport Model
       for the Simple Network Management Protocol (SNMP).
       SNMP-TLS-TM-MIB.snmpTlstmCertSANRFC822Name";
}

identity san-dns-name {
  base cert-to-name;
  description
    "Maps a subjectAltName's dNSName to a name after first
     converting it to all lowercase (RFC 5280 does not specify
```


Copyright© 2022 Ciena® Corporation

```
      converting to lowercase, so this involves an extra step).
      This mapping results in a 1:1 correspondence between
      subjectAltName dNSName values and the name values.";
  reference
    "RFC 6353: Transport Layer Security (TLS) Transport Model
       for the Simple Network Management Protocol (SNMP).
       SNMP-TLS-TM-MIB.snmpTlstmCertSANDNSName";
}

identity san-ip-address {
  base cert-to-name;
  description
    "Maps a subjectAltName's iPAddress to a name by
     transforming the binary-encoded address as follows:

       1) for IPv4, the value is converted into a
          decimal-dotted quad address (e.g., '192.0.2.1').

       2) for IPv6 addresses, the value is converted into a
          32-character, all-lowercase hexadecimal string
          without any colon separators.

     This mapping results in a 1:1 correspondence between
     subjectAltName iPAddress values and the name values.";
  reference
    "RFC 6353: Transport Layer Security (TLS) Transport Model
       for the Simple Network Management Protocol (SNMP).
       SNMP-TLS-TM-MIB.snmpTlstmCertSANIpAddress";
}

identity san-any {
  base cert-to-name;
  description
    "Maps any of the following fields using the corresponding
     mapping algorithms:

       +------------+-----------------+
       | Type       | Algorithm       |
       |------------+-----------------|
       | rfc822Name | san-rfc822-name |
       | dNSName    | san-dns-name    |
       | iPAddress  | san-ip-address  |
       +------------+-----------------+

     The first matching subjectAltName value found in the
     certificate of the above types MUST be used when deriving
     the name.  The mapping algorithm specified in the
     'Algorithm' column MUST be used to derive the name.

     This mapping results in a 1:1 correspondence between
     subjectAltName values and name values.  The three sub-mapping
     algorithms produced by this combined algorithm cannot produce
     conflicting results between themselves.";
  reference
    "RFC 6353: Transport Layer Security (TLS) Transport Model
       for the Simple Network Management Protocol (SNMP).
       SNMP-TLS-TM-MIB.snmpTlstmCertSANAny";
}

identity common-name {
```


Copyright© 2022 Ciena® Corporation

```
    base cert-to-name;
    description
      "Maps a certificate's CommonName to a name after converting
       it to a UTF-8 encoding.  The usage of CommonNames is
       deprecated, and users are encouraged to use subjectAltName
       mapping methods instead.  This mapping results in a 1:1
       correspondence between certificate CommonName values and name
       values.";
    reference
      "RFC 6353: Transport Layer Security (TLS) Transport Model
         for the Simple Network Management Protocol (SNMP).
         SNMP-TLS-TM-MIB.snmpTlstmCertCommonName";
  }

  /*
   * Groupings
   */

  grouping cert-to-name {
    description
      "Defines nodes for mapping certificates to names.  Modules
       that use this grouping should describe how the resulting
       name is used.";

    list cert-to-name {
      key id;
      description
        "This list defines how certificates are mapped to names.
         The name is derived by considering each cert-to-name
         list entry in order.  The cert-to-name entry's fingerprint
         determines whether the list entry is a match:

         1) If the cert-to-name list entry's fingerprint value
            matches that of the presented certificate, then consider
            the list entry a successful match.

         2) If the cert-to-name list entry's fingerprint value
            matches that of a locally held copy of a trusted CA
            certificate, and that CA certificate was part of the CA
            certificate chain to the presented certificate, then
            consider the list entry a successful match.

        Once a matching cert-to-name list entry has been found, the
        map-type is used to determine how the name associated with
        the certificate should be determined.  See the map-type
        leaf's description for details on determining the name value.
        If it is impossible to determine a name from the cert-to-name
        list entry's data combined with the data presented in the
        certificate, then additional cert-to-name list entries MUST
        be searched to look for another potential match.

        Security administrators are encouraged to make use of
        certificates with subjectAltName fields that can be mapped to
        names so that a single root CA certificate can allow all
        child certificates' subjectAltName fields to map directly to
        a name via a 1:1 transformation.";
      reference
       "RFC 6353: Transport Layer Security (TLS) Transport Model
          for the Simple Network Management Protocol (SNMP).
          SNMP-TLS-TM-MIB.snmpTlstmCertToTSNEntry";
```


Copyright© 2022 Ciena® Corporation

```
leaf id {
  type uint32;
  description
    "The id specifies the order in which the entries in the
     cert-to-name list are searched.  Entries with lower
     numbers are searched first.";
  reference
    "RFC 6353: Transport Layer Security (TLS) Transport Model
       for the Simple Network Management Protocol
       (SNMP).
       SNMP-TLS-TM-MIB.snmpTlstmCertToTSNID";
}

leaf fingerprint {
  type x509c2n:tls-fingerprint;
  mandatory true;
  description
    "Specifies a value with which the fingerprint of the
     full certificate presented by the peer is compared.  If
     the fingerprint of the full certificate presented by the
     peer does not match the fingerprint configured, then the
     entry is skipped, and the search for a match continues.";
  reference
    "RFC 6353: Transport Layer Security (TLS) Transport Model
       for the Simple Network Management Protocol
       (SNMP).
       SNMP-TLS-TM-MIB.snmpTlstmCertToTSNFingerprint";
}

leaf map-type {
  type identityref {
    base cert-to-name;
  }
  mandatory true;
  description
    "Specifies the algorithm used to map the certificate
     presented by the peer to a name.

     Mappings that need additional configuration objects should
     use the 'when' statement to make them conditional based on
     the map-type.";
  reference
    "RFC 6353: Transport Layer Security (TLS) Transport Model
       for the Simple Network Management Protocol
       (SNMP).
       SNMP-TLS-TM-MIB.snmpTlstmCertToTSNMapType";
}

leaf name {
  when "../map-type = 'x509c2n:specified'";
  type string;
  mandatory true;
  description
    "Directly specifies the NETCONF username when the
     map-type is 'specified'.";
  reference
    "RFC 6353: Transport Layer Security (TLS) Transport Model
       for the Simple Network Management Protocol
       (SNMP).
```


Copyright© 2022 Ciena® Corporation

```
                SNMP-TLS-TM-MIB.snmpTlstmCertToTSNData";
        }
      }
    }
}
```


Copyright© 2022 Ciena® Corporation

ietf-yang-smiv2.yang

```
module ietf-yang-smiv2 {

  namespace "urn:ietf:params:xml:ns:yang:ietf-yang-smiv2";
  prefix "smiv2";

  organization
   "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

  contact
   "WG Web:   http://tools.ietf.org/wg/netmod/
    WG List:  netmod@ietf.org

    WG Chair: David Kessens
              david.kessens@nsn.com

    WG Chair: Juergen Schoenwaelder
              j.schoenwaelder@jacobs-university.de

    Editor:   Juergen Schoenwaelder
              j.schoenwaelder@jacobs-university.de";

  description
   "This module defines YANG extensions that are used to translate
    SMIv2 concepts into YANG.

    Copyright (c) 2012 IETF Trust and the persons identified as
    authors of the code.  All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject
    to the license terms contained in, the Simplified BSD License
    set forth in Section 4 of the IETF Trust's Legal Provisions

    Relating to IETF Documents
    http://trustee.ietf.org/license-info.

    This version of this YANG module is part of RFC 6643 see
    the RFC itself for full legal notices.";

  revision 2012-06-22 {
    description
     "Initial revision.";
    reference
     "RFC 6643: Translation of Structure of Management Information
      Version 2 (SMIv2) MIB Modules to YANG Modules";
  }

  identity object-identity {
    description
     "Base identity for all SMIv2 OBJECT-IDENTITYs.";
  }

  typedef opaque {
    type binary;
    description
     "The Opaque type supports the capability to pass arbitrary ASN.1
      syntax.  A value is encoded using the ASN.1 Basic Encoding Rules
      into a string of octets.  This, in turn, is encoded as an OCTET
```

```
     STRING, in effect 'double-wrapping' the original ASN.1 value.

     In the value set and its semantics, this type is equivalent to
     the Opaque type of the SMIv2.  This type exists in the SMIv2
     solely for backward-compatibility reasons and this is also
     true for this YANG data type.";
   reference
    "RFC 2578: Structure of Management Information Version 2 (SMIv2)";
}

extension display-hint {
   argument "format";
   description
    "The display-hint statement takes as an argument the DISPLAY-HINT
     assigned to an SMIv2 textual convention.";
   reference
    "RFC 2579: Textual Conventions for SMIv2";
}

extension max-access {
   argument "access";
   description
    "The max-access statement takes as an argument the MAX-ACCESS
     assigned to an SMIv2 object definition.

     The MAX-ACCESS value is SMIv2 specific and has no impact on
     the access provided to YANG objects through protocols such
     as NETCONF.";
   reference
    "RFC 2578: Structure of Management Information Version 2 (SMIv2)";
}

extension defval {
   argument "value";
   description
    "The defval statement takes as an argument a default value
     defined by an SMIv2 DEFVAL clause.  Note that the value is in
     the SMIv2 value space defined by the SMIv2 syntax of the
     corresponding object and not in the YANG value space
     defined by the corresponding YANG data type.";
   reference
    "RFC 2578: Structure of Management Information Version 2 (SMIv2)";
}

extension implied {
   argument "index";
   description
    "If an SMIv2 INDEX object is preceded by the IMPLIED keyword, then
     the implied statement is present in the YANG module and takes as
     an argument the name of the IMPLIED index object.";
   reference
    "RFC 2578: Structure of Management Information Version 2 (SMIv2)";
 }

extension alias {
   argument "descriptor";
   description
    "The alias statement introduces an SMIv2 descriptor.  The body of
     the alias statement is expected to contain an oid statement that
     provides the numeric OID associated with the descriptor.";
```


Copyright© 2022 Ciena® Corporation

```
    reference
     "RFC 2578: Structure of Management Information Version 2 (SMIv2)";
  }

  extension oid {
    argument "value";
    description
     "The oid statement takes as an argument the object identifier
      assigned to an SMIv2 definition.  The object identifier value
      is written in decimal dotted notation.";
    reference
     "RFC 2578: Structure of Management Information Version 2 (SMIv2)";
  }

  extension subid {
    argument "value";
    description
     "The subid statement takes as an argument the last sub-identifier
      of the object identifier assigned to an SMIv2 definition.  The
      sub-identifier value is a single positive decimal natural number.
      The subid statement may not be used as a substatement to any
      top-level node in a YANG document.  The subid substatement may
      be used only as a substatement to a node having a parent node
      defined with either an smiv2:oid or smiv2:subid substatement.";
    reference
     "RFC 2578: Structure of Management Information Version 2 (SMIv2)";
  }
}
```


Copyright© 2022 Ciena® Corporation

ietf-yang-types.yang

```
// generated from rfc6991.txt

   module ietf-yang-types {

     namespace "urn:ietf:params:xml:ns:yang:ietf-yang-types";
     prefix "yang";

     organization
      "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

     contact
      "WG Web:   <http://tools.ietf.org/wg/netmod/>
       WG List:  <mailto:netmod@ietf.org>
       WG Chair: David Kessens
                 <mailto:david.kessens@nsn.com>

       WG Chair: Juergen Schoenwaelder
                 <mailto:j.schoenwaelder@jacobs-university.de>

       Editor:   Juergen Schoenwaelder
                 <mailto:j.schoenwaelder@jacobs-university.de>";

     description
      "This module contains a collection of generally useful derived
       YANG data types.

       Copyright (c) 2013 IETF Trust and the persons identified as
       authors of the code.  All rights reserved.

       Redistribution and use in source and binary forms, with or
       without modification, is permitted pursuant to, and subject
       to the license terms contained in, the Simplified BSD License
       set forth in Section 4.c of the IETF Trust's Legal Provisions
       Relating to IETF Documents
       (http://trustee.ietf.org/license-info).

       This version of this YANG module is part of RFC 6991; see
       the RFC itself for full legal notices.";

     revision 2013-07-15 {
       description
        "This revision adds the following new data types:
         - yang-identifier
         - hex-string
         - uuid
         - dotted-quad";
       reference
        "RFC 6991: Common YANG Data Types";
     }

     revision 2010-09-24 {
       description
        "Initial revision.";
       reference
        "RFC 6021: Common YANG Data Types";
     }

     /*** collection of counter and gauge types ***/
```


Copyright© 2022 Ciena® Corporation

```
typedef counter32 {
  type uint32;
  description
   "The counter32 type represents a non-negative integer
    that monotonically increases until it reaches a
    maximum value of 2^32-1 (4294967295 decimal), when it
    wraps around and starts increasing again from zero.

    Counters have no defined 'initial' value, and thus, a
    single value of a counter has (in general) no information
    content.  Discontinuities in the monotonically increasing
    value normally occur at re-initialization of the
    management system, and at other times as specified in the
    description of a schema node using this type.  If such
    other times can occur, for example, the creation of
    a schema node of type counter32 at times other than
    re-initialization, then a corresponding schema node
    should be defined, with an appropriate type, to indicate
    the last discontinuity.

    The counter32 type should not be used for configuration
    schema nodes.  A default statement SHOULD NOT be used in
    combination with the type counter32.

    In the value set and its semantics, this type is equivalent
    to the Counter32 type of the SMIv2.";
  reference
   "RFC 2578: Structure of Management Information Version 2
              (SMIv2)";
}

typedef zero-based-counter32 {
  type yang:counter32;
  default "0";
  description
   "The zero-based-counter32 type represents a counter32
    that has the defined 'initial' value zero.

    A schema node of this type will be set to zero (0) on creation
    and will thereafter increase monotonically until it reaches
    a maximum value of 2^32-1 (4294967295 decimal), when it
    wraps around and starts increasing again from zero.

    Provided that an application discovers a new schema node
    of this type within the minimum time to wrap, it can use the
    'initial' value as a delta.  It is important for a management
    station to be aware of this minimum time and the actual time
    between polls, and to discard data if the actual time is too
    long or there is no defined minimum time.

    In the value set and its semantics, this type is equivalent
    to the ZeroBasedCounter32 textual convention of the SMIv2.";
  reference
    "RFC 4502: Remote Network Monitoring Management Information
               Base Version 2";
}

typedef counter64 {
  type uint64;
```


Copyright© 2022 Ciena® Corporation

```
    description
     "The counter64 type represents a non-negative integer
      that monotonically increases until it reaches a
      maximum value of 2^64-1 (18446744073709551615 decimal),
      when it wraps around and starts increasing again from zero.

      Counters have no defined 'initial' value, and thus, a
      single value of a counter has (in general) no information
      content.  Discontinuities in the monotonically increasing
      value normally occur at re-initialization of the
      management system, and at other times as specified in the
      description of a schema node using this type.  If such
      other times can occur, for example, the creation of
      a schema node of type counter64 at times other than
      re-initialization, then a corresponding schema node
      should be defined, with an appropriate type, to indicate
      the last discontinuity.

      The counter64 type should not be used for configuration
      schema nodes.  A default statement SHOULD NOT be used in
      combination with the type counter64.

      In the value set and its semantics, this type is equivalent
      to the Counter64 type of the SMIv2.";
    reference
     "RFC 2578: Structure of Management Information Version 2
                (SMIv2)";
  }

  typedef zero-based-counter64 {
    type yang:counter64;
    default "0";
    description
     "The zero-based-counter64 type represents a counter64 that
      has the defined 'initial' value zero.

      A schema node of this type will be set to zero (0) on creation
      and will thereafter increase monotonically until it reaches
      a maximum value of 2^64-1 (18446744073709551615 decimal),
      when it wraps around and starts increasing again from zero.

      Provided that an application discovers a new schema node
      of this type within the minimum time to wrap, it can use the
      'initial' value as a delta.  It is important for a management
      station to be aware of this minimum time and the actual time
      between polls, and to discard data if the actual time is too
      long or there is no defined minimum time.

      In the value set and its semantics, this type is equivalent
      to the ZeroBasedCounter64 textual convention of the SMIv2.";
    reference
     "RFC 2856: Textual Conventions for Additional High Capacity
                Data Types";
  }

  typedef gauge32 {
    type uint32;
    description
     "The gauge32 type represents a non-negative integer, which
      may increase or decrease, but shall never exceed a maximum
```


Copyright© 2022 Ciena® Corporation

```
      value, nor fall below a minimum value.  The maximum value
      cannot be greater than 2^32-1 (4294967295 decimal), and
      the minimum value cannot be smaller than 0.  The value of
      a gauge32 has its maximum value whenever the information
      being modeled is greater than or equal to its maximum
      value, and has its minimum value whenever the information
      being modeled is smaller than or equal to its minimum value.
      If the information being modeled subsequently decreases
      below (increases above) the maximum (minimum) value, the
      gauge32 also decreases (increases).

      In the value set and its semantics, this type is equivalent
      to the Gauge32 type of the SMIv2.";
    reference
     "RFC 2578: Structure of Management Information Version 2
                (SMIv2)";
  }

  typedef gauge64 {
    type uint64;
    description
     "The gauge64 type represents a non-negative integer, which
      may increase or decrease, but shall never exceed a maximum
      value, nor fall below a minimum value.  The maximum value
      cannot be greater than 2^64-1 (18446744073709551615), and
      the minimum value cannot be smaller than 0.  The value of
      a gauge64 has its maximum value whenever the information
      being modeled is greater than or equal to its maximum
      value, and has its minimum value whenever the information
      being modeled is smaller than or equal to its minimum value.
      If the information being modeled subsequently decreases
      below (increases above) the maximum (minimum) value, the
      gauge64 also decreases (increases).

      In the value set and its semantics, this type is equivalent
      to the CounterBasedGauge64 SMIv2 textual convention defined
      in RFC 2856";
    reference
     "RFC 2856: Textual Conventions for Additional High Capacity
                Data Types";
  }

  /*** collection of identifier-related types ***/

  typedef object-identifier {
    type string {
      pattern '(([0-1](\.[1-3]?[0-9]))|(2\.(0|([1-9]\d*))))'
            + '(\.(0|([1-9]\d*)))*';
    }
    description
     "The object-identifier type represents administratively
      assigned names in a registration-hierarchical-name tree.

      Values of this type are denoted as a sequence of numerical
      non-negative sub-identifier values.  Each sub-identifier
      value MUST NOT exceed 2^32-1 (4294967295).  Sub-identifiers
      are separated by single dots and without any intermediate
      whitespace.

      The ASN.1 standard restricts the value space of the first
```


Copyright© 2022 Ciena® Corporation

```
      sub-identifier to 0, 1, or 2.  Furthermore, the value space
      of the second sub-identifier is restricted to the range
      0 to 39 if the first sub-identifier is 0 or 1.  Finally,
      the ASN.1 standard requires that an object identifier
      has always at least two sub-identifiers.  The pattern
      captures these restrictions.

      Although the number of sub-identifiers is not limited,
      module designers should realize that there may be
      implementations that stick with the SMIv2 limit of 128
      sub-identifiers.

      This type is a superset of the SMIv2 OBJECT IDENTIFIER type
      since it is not restricted to 128 sub-identifiers.  Hence,
      this type SHOULD NOT be used to represent the SMIv2 OBJECT
      IDENTIFIER type; the object-identifier-128 type SHOULD be
      used instead.";
    reference
     "ISO9834-1: Information technology -- Open Systems
      Interconnection -- Procedures for the operation of OSI
      Registration Authorities: General procedures and top
      arcs of the ASN.1 Object Identifier tree";
  }

  typedef object-identifier-128 {
    type object-identifier {
      pattern '\d*(\.\d*){1,127}';
    }
    description
     "This type represents object-identifiers restricted to 128
      sub-identifiers.

      In the value set and its semantics, this type is equivalent
      to the OBJECT IDENTIFIER type of the SMIv2.";
    reference
     "RFC 2578: Structure of Management Information Version 2
                (SMIv2)";
  }

  typedef yang-identifier {
    type string {
      length "1..max";
      pattern '[a-zA-Z_][a-zA-Z0-9\-_.]*';
      pattern '.|..|[^xX].*|.[^mM].*|..[^lL].*';
    }
    description
      "A YANG identifier string as defined by the 'identifier'
       rule in Section 12 of RFC 6020.  An identifier must
       start with an alphabetic character or an underscore
       followed by an arbitrary sequence of alphabetic or
       numeric characters, underscores, hyphens, or dots.

       A YANG identifier MUST NOT start with any possible
       combination of the lowercase or uppercase character
       sequence 'xml'.";
    reference
      "RFC 6020: YANG - A Data Modeling Language for the Network
                 Configuration Protocol (NETCONF)";
  }
```


Copyright© 2022 Ciena® Corporation

```
/*** collection of types related to date and time***/

typedef date-and-time {
  type string {
    pattern '\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}(\.\d+)?'
          + '(Z|[\+\-]\d{2}:\d{2})';
  }
  description
   "The date-and-time type is a profile of the ISO 8601
    standard for representation of dates and times using the
    Gregorian calendar.  The profile is defined by the
    date-time production in Section 5.6 of RFC 3339.

    The date-and-time type is compatible with the dateTime XML
    schema type with the following notable exceptions:

    (a) The date-and-time type does not allow negative years.

    (b) The date-and-time time-offset -00:00 indicates an unknown
        time zone (see RFC 3339) while -00:00 and +00:00 and Z
        all represent the same time zone in dateTime.

    (c) The canonical format (see below) of data-and-time values
        differs from the canonical format used by the dateTime XML
        schema type, which requires all times to be in UTC using
        the time-offset 'Z'.

    This type is not equivalent to the DateAndTime textual
    convention of the SMIv2 since RFC 3339 uses a different
    separator between full-date and full-time and provides
    higher resolution of time-secfrac.

    The canonical format for date-and-time values with a known time
    zone uses a numeric time zone offset that is calculated using
    the device's configured known offset to UTC time.  A change of
    the device's offset to UTC time will cause date-and-time values
    to change accordingly.  Such changes might happen periodically
    in case a server follows automatically daylight saving time
    (DST) time zone offset changes.  The canonical format for
    date-and-time values with an unknown time zone (usually
    referring to the notion of local time) uses the time-offset
    -00:00.";
  reference
   "RFC 3339: Date and Time on the Internet: Timestamps
    RFC 2579: Textual Conventions for SMIv2
    XSD-TYPES: XML Schema Part 2: Datatypes Second Edition";
}

typedef timeticks {
  type uint32;
  description
   "The timeticks type represents a non-negative integer that
    represents the time, modulo 2^32 (4294967296 decimal), in
    hundredths of a second between two epochs.  When a schema
    node is defined that uses this type, the description of
    the schema node identifies both of the reference epochs.

    In the value set and its semantics, this type is equivalent
    to the TimeTicks type of the SMIv2.";
  reference
```


Copyright© 2022 Ciena® Corporation

```
    "RFC 2578: Structure of Management Information Version 2
               (SMIv2)";
}

typedef timestamp {
  type yang:timeticks;
  description
   "The timestamp type represents the value of an associated
    timeticks schema node at which a specific occurrence
    happened.  The specific occurrence must be defined in the
    description of any schema node defined using this type.  When
    the specific occurrence occurred prior to the last time the
    associated timeticks attribute was zero, then the timestamp
    value is zero.  Note that this requires all timestamp values
    to be reset to zero when the value of the associated timeticks
    attribute reaches 497+ days and wraps around to zero.

    The associated timeticks schema node must be specified
    in the description of any schema node using this type.

    In the value set and its semantics, this type is equivalent
    to the TimeStamp textual convention of the SMIv2.";
  reference
   "RFC 2579: Textual Conventions for SMIv2";
}

/*** collection of generic address types ***/

typedef phys-address {
  type string {
    pattern '([0-9a-fA-F]{2}(:[0-9a-fA-F]{2})*)?';
  }

  description
   "Represents media- or physical-level addresses represented
    as a sequence octets, each octet represented by two hexadecimal
    numbers.  Octets are separated by colons.  The canonical
    representation uses lowercase characters.

    In the value set and its semantics, this type is equivalent
    to the PhysAddress textual convention of the SMIv2.";
  reference
   "RFC 2579: Textual Conventions for SMIv2";
}

typedef mac-address {
  type string {
    pattern '[0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){5}';
  }
  description
   "The mac-address type represents an IEEE 802 MAC address.
    The canonical representation uses lowercase characters.

    In the value set and its semantics, this type is equivalent
    to the MacAddress textual convention of the SMIv2.";
  reference
   "IEEE 802: IEEE Standard for Local and Metropolitan Area
              Networks: Overview and Architecture
    RFC 2579: Textual Conventions for SMIv2";
}
```


Copyright© 2022 Ciena® Corporation

```
  /*** collection of XML-specific types ***/

  typedef xpath1.0 {
    type string;
    description
     "This type represents an XPATH 1.0 expression.

      When a schema node is defined that uses this type, the
      description of the schema node MUST specify the XPath
      context in which the XPath expression is evaluated.";
    reference
     "XPATH: XML Path Language (XPath) Version 1.0";
  }

  /*** collection of string types ***/

  typedef hex-string {
    type string {
      pattern '([0-9a-fA-F]{2}(:[0-9a-fA-F]{2})*)?';
    }

    description
     "A hexadecimal string with octets represented as hex digits
      separated by colons.  The canonical representation uses
      lowercase characters.";
  }

  typedef uuid {
    type string {
      pattern '[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-'
            + '[0-9a-fA-F]{4}-[0-9a-fA-F]{12}';
    }
    description
     "A Universally Unique IDentifier in the string representation
      defined in RFC 4122.  The canonical representation uses
      lowercase characters.

      The following is an example of a UUID in string representation:
      f81d4fae-7dec-11d0-a765-00a0c91e6bf6
      ";
    reference
     "RFC 4122: A Universally Unique IDentifier (UUID) URN
                Namespace";
  }

  typedef dotted-quad {
    type string {
      pattern
        '(([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.){3}'
      + '([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])';
    }
    description
      "An unsigned 32-bit number expressed in the dotted-quad
       notation, i.e., four octets written as decimal numbers
       and separated with the '.' (full stop) character.";
  }
}
```


Copyright© 2022 Ciena® Corporation

ietf-yang-types@2010-09-24.yang

```
module ietf-yang-types {

   namespace "urn:ietf:params:xml:ns:yang:ietf-yang-types";
   prefix "yang";

   organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

   contact
    "WG Web:   <http://tools.ietf.org/wg/netmod/>
     WG List:  <mailto:netmod@ietf.org>

     WG Chair: David Partain
               <mailto:david.partain@ericsson.com>

     WG Chair: David Kessens
               <mailto:david.kessens@nsn.com>

     Editor:   Juergen Schoenwaelder
               <mailto:j.schoenwaelder@jacobs-university.de>";

   description
    "This module contains a collection of generally useful derived
     YANG data types.

     Copyright (c) 2010 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or without
     modification, is permitted pursuant to, and subject to the license
     terms contained in, the Simplified BSD License set forth in Section
     4.c of the IETF Trust's Legal Provisions Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC 6021; see
     the RFC itself for full legal notices.";

   revision 2010-09-24 {
     description
      "Initial revision.";
     reference
      "RFC 6021: Common YANG Data Types";
   }

   /*** collection of counter and gauge types ***/

   typedef counter32 {
     type uint32;
     description
      "The counter32 type represents a non-negative integer
       that monotonically increases until it reaches a
       maximum value of 2^32-1 (4294967295 decimal), when it
       wraps around and starts increasing again from zero.

       Counters have no defined 'initial' value, and thus, a
       single value of a counter has (in general) no information
       content.  Discontinuities in the monotonically increasing
       value normally occur at re-initialization of the
```

```
      management system, and at other times as specified in the
      description of a schema node using this type.  If such
      other times can occur, for example, the creation of
      a schema node of type counter32 at times other than
      re-initialization, then a corresponding schema node
      should be defined, with an appropriate type, to indicate
      the last discontinuity.

      The counter32 type should not be used for configuration
      schema nodes.  A default statement SHOULD NOT be used in
      combination with the type counter32.

      In the value set and its semantics, this type is equivalent
      to the Counter32 type of the SMIv2.";
    reference
     "RFC 2578: Structure of Management Information Version 2 (SMIv2)";
  }

  typedef zero-based-counter32 {
    type yang:counter32;
    default "0";
    description
     "The zero-based-counter32 type represents a counter32
      that has the defined 'initial' value zero.

      A schema node of this type will be set to zero (0) on creation
      and will thereafter increase monotonically until it reaches
      a maximum value of 2^32-1 (4294967295 decimal), when it
      wraps around and starts increasing again from zero.

      Provided that an application discovers a new schema node
      of this type within the minimum time to wrap, it can use the
      'initial' value as a delta.  It is important for a management
      station to be aware of this minimum time and the actual time
      between polls, and to discard data if the actual time is too
      long or there is no defined minimum time.

      In the value set and its semantics, this type is equivalent
      to the ZeroBasedCounter32 textual convention of the SMIv2.";
    reference
      "RFC 4502: Remote Network Monitoring Management Information
                 Base Version 2";
  }

  typedef counter64 {
    type uint64;
    description
     "The counter64 type represents a non-negative integer
      that monotonically increases until it reaches a
      maximum value of 2^64-1 (18446744073709551615 decimal),
      when it wraps around and starts increasing again from zero.

      Counters have no defined 'initial' value, and thus, a
      single value of a counter has (in general) no information
      content.  Discontinuities in the monotonically increasing
      value normally occur at re-initialization of the
      management system, and at other times as specified in the
      description of a schema node using this type.  If such
      other times can occur, for example, the creation of
      a schema node of type counter64 at times other than
```


Copyright© 2022 Ciena® Corporation

```
      re-initialization, then a corresponding schema node
      should be defined, with an appropriate type, to indicate
      the last discontinuity.

      The counter64 type should not be used for configuration
      schema nodes.  A default statement SHOULD NOT be used in
      combination with the type counter64.

      In the value set and its semantics, this type is equivalent
      to the Counter64 type of the SMIv2.";
    reference
     "RFC 2578: Structure of Management Information Version 2 (SMIv2)";
  }

  typedef zero-based-counter64 {
    type yang:counter64;
    default "0";
    description
     "The zero-based-counter64 type represents a counter64 that
      has the defined 'initial' value zero.

      A schema node of this type will be set to zero (0) on creation
      and will thereafter increase monotonically until it reaches
      a maximum value of 2^64-1 (18446744073709551615 decimal),
      when it wraps around and starts increasing again from zero.

      Provided that an application discovers a new schema node
      of this type within the minimum time to wrap, it can use the
      'initial' value as a delta.  It is important for a management
      station to be aware of this minimum time and the actual time
      between polls, and to discard data if the actual time is too
      long or there is no defined minimum time.

      In the value set and its semantics, this type is equivalent
      to the ZeroBasedCounter64 textual convention of the SMIv2.";
    reference
     "RFC 2856: Textual Conventions for Additional High Capacity
                Data Types";
  }

  typedef gauge32 {
    type uint32;
    description
     "The gauge32 type represents a non-negative integer, which
      may increase or decrease, but shall never exceed a maximum
      value, nor fall below a minimum value.  The maximum value
      cannot be greater than 2^32-1 (4294967295 decimal), and
      the minimum value cannot be smaller than 0.  The value of
      a gauge32 has its maximum value whenever the information
      being modeled is greater than or equal to its maximum
      value, and has its minimum value whenever the information
      being modeled is smaller than or equal to its minimum value.
      If the information being modeled subsequently decreases
      below (increases above) the maximum (minimum) value, the
      gauge32 also decreases (increases).

      In the value set and its semantics, this type is equivalent
      to the Gauge32 type of the SMIv2.";
    reference
     "RFC 2578: Structure of Management Information Version 2 (SMIv2)";
```


Copyright© 2022 Ciena® Corporation

```
}

typedef gauge64 {
  type uint64;
  description
   "The gauge64 type represents a non-negative integer, which
    may increase or decrease, but shall never exceed a maximum
    value, nor fall below a minimum value.  The maximum value
    cannot be greater than 2^64-1 (18446744073709551615), and
    the minimum value cannot be smaller than 0.  The value of
    a gauge64 has its maximum value whenever the information
    being modeled is greater than or equal to its maximum
    value, and has its minimum value whenever the information
    being modeled is smaller than or equal to its minimum value.
    If the information being modeled subsequently decreases
    below (increases above) the maximum (minimum) value, the
    gauge64 also decreases (increases).

    In the value set and its semantics, this type is equivalent
    to the CounterBasedGauge64 SMIv2 textual convention defined
    in RFC 2856";
  reference
   "RFC 2856: Textual Conventions for Additional High Capacity
              Data Types";
}

/*** collection of identifier related types ***/

typedef object-identifier {
  type string {
    pattern '(([0-1](\.[1-3]?[0-9]))|(2\.(0|([1-9]\d*))))'
          + '(\.(0|([1-9]\d*)))*';
  }
  description
   "The object-identifier type represents administratively
    assigned names in a registration-hierarchical-name tree.

    Values of this type are denoted as a sequence of numerical
    non-negative sub-identifier values.  Each sub-identifier
    value MUST NOT exceed 2^32-1 (4294967295).  Sub-identifiers
    are separated by single dots and without any intermediate
    whitespace.

    The ASN.1 standard restricts the value space of the first
    sub-identifier to 0, 1, or 2.  Furthermore, the value space
    of the second sub-identifier is restricted to the range
    0 to 39 if the first sub-identifier is 0 or 1.  Finally,
    the ASN.1 standard requires that an object identifier
    has always at least two sub-identifier.  The pattern
    captures these restrictions.

    Although the number of sub-identifiers is not limited,
    module designers should realize that there may be
    implementations that stick with the SMIv2 limit of 128
    sub-identifiers.

    This type is a superset of the SMIv2 OBJECT IDENTIFIER type
    since it is not restricted to 128 sub-identifiers.  Hence,
    this type SHOULD NOT be used to represent the SMIv2 OBJECT
    IDENTIFIER type, the object-identifier-128 type SHOULD be
```


Copyright© 2022 Ciena® Corporation

```
      used instead.";
    reference
     "ISO9834-1: Information technology -- Open Systems
      Interconnection -- Procedures for the operation of OSI
      Registration Authorities: General procedures and top
      arcs of the ASN.1 Object Identifier tree";
  }

  typedef object-identifier-128 {
    type object-identifier {
      pattern '\d*(\.\d*){1,127}';
    }
    description
     "This type represents object-identifiers restricted to 128
      sub-identifiers.

      In the value set and its semantics, this type is equivalent
      to the OBJECT IDENTIFIER type of the SMIv2.";
    reference
     "RFC 2578: Structure of Management Information Version 2 (SMIv2)";
  }

  /*** collection of date and time related types ***/

  typedef date-and-time {
    type string {
      pattern '\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}(\.\d+)?'
            + '(Z|[\+\-]\d{2}:\d{2})';
    }
    description
     "The date-and-time type is a profile of the ISO 8601
      standard for representation of dates and times using the
      Gregorian calendar.  The profile is defined by the
      date-time production in Section 5.6 of RFC 3339.

      The date-and-time type is compatible with the dateTime XML
      schema type with the following notable exceptions:

      (a) The date-and-time type does not allow negative years.

      (b) The date-and-time time-offset -00:00 indicates an unknown
          time zone (see RFC 3339) while -00:00 and +00:00 and Z all
          represent the same time zone in dateTime.

      (c) The canonical format (see below) of data-and-time values
          differs from the canonical format used by the dateTime XML
          schema type, which requires all times to be in UTC using the
          time-offset 'Z'.

      This type is not equivalent to the DateAndTime textual
      convention of the SMIv2 since RFC 3339 uses a different
      separator between full-date and full-time and provides
      higher resolution of time-secfrac.

      The canonical format for date-and-time values with a known time
      zone uses a numeric time zone offset that is calculated using
      the device's configured known offset to UTC time.  A change of
      the device's offset to UTC time will cause date-and-time values
      to change accordingly.  Such changes might happen periodically
      in case a server follows automatically daylight saving time
```


Copyright© 2022 Ciena® Corporation

```
      (DST) time zone offset changes.  The canonical format for
      date-and-time values with an unknown time zone (usually referring
      to the notion of local time) uses the time-offset -00:00.";
    reference
     "RFC 3339: Date and Time on the Internet: Timestamps
      RFC 2579: Textual Conventions for SMIv2
      XSD-TYPES: XML Schema Part 2: Datatypes Second Edition";
  }

  typedef timeticks {
    type uint32;
    description
     "The timeticks type represents a non-negative integer that
      represents the time, modulo 2^32 (4294967296 decimal), in
      hundredths of a second between two epochs.  When a schema
      node is defined that uses this type, the description of
      the schema node identifies both of the reference epochs.

      In the value set and its semantics, this type is equivalent
      to the TimeTicks type of the SMIv2.";
    reference
     "RFC 2578: Structure of Management Information Version 2 (SMIv2)";
  }

  typedef timestamp {
    type yang:timeticks;
    description
     "The timestamp type represents the value of an associated
      timeticks schema node at which a specific occurrence happened.
      The specific occurrence must be defined in the description
      of any schema node defined using this type.  When the specific
      occurrence occurred prior to the last time the associated
      timeticks attribute was zero, then the timestamp value is
      zero.  Note that this requires all timestamp values to be
      reset to zero when the value of the associated timeticks
      attribute reaches 497+ days and wraps around to zero.

      The associated timeticks schema node must be specified
      in the description of any schema node using this type.

      In the value set and its semantics, this type is equivalent
      to the TimeStamp textual convention of the SMIv2.";

    reference
     "RFC 2579: Textual Conventions for SMIv2";
  }

  /*** collection of generic address types ***/

  typedef phys-address {
    type string {
      pattern '([0-9a-fA-F]{2}(:[0-9a-fA-F]{2})*)?';
    }
    description
     "Represents media- or physical-level addresses represented
      as a sequence octets, each octet represented by two hexadecimal
      numbers.  Octets are separated by colons.  The canonical
      representation uses lowercase characters.

      In the value set and its semantics, this type is equivalent
```


Copyright© 2022 Ciena® Corporation

```
      to the PhysAddress textual convention of the SMIv2.";
    reference
     "RFC 2579: Textual Conventions for SMIv2";
  }

  typedef mac-address {
    type string {
      pattern '[0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){5}';
    }
    description
     "The mac-address type represents an IEEE 802 MAC address.
      The canonical representation uses lowercase characters.

      In the value set and its semantics, this type is equivalent
      to the MacAddress textual convention of the SMIv2.";
    reference
     "IEEE 802: IEEE Standard for Local and Metropolitan Area
                Networks: Overview and Architecture
      RFC 2579: Textual Conventions for SMIv2";
  }

  /*** collection of XML specific types ***/

  typedef xpath1.0 {
    type string;
    description
     "This type represents an XPATH 1.0 expression.

      When a schema node is defined that uses this type, the
      description of the schema node MUST specify the XPath
      context in which the XPath expression is evaluated.";

    reference
     "XPATH: XML Path Language (XPath) Version 1.0";
  }

}
```


Copyright© 2022 Ciena® Corporation

mef-cfm-deviation.yang

```
module mef-cfm-deviation {

    namespace "urn:ciena:params:xml:ns:yang:ciena-pn:mef-cfm-dev";
    prefix "mef-cfm-dev";

    import mef-cfm { prefix mef-cfm; }

    revision 2017-06-16 {
        description
          "Updated the model with the below changes :
            - Added vlan-id-wildcard-type typedef definition.
            - Replaced mef-cfm:maintenance-association-end-point/mef-
cfm:primary-vid
              field with type vlan-id-wildcard-type";
    }

    revision 2015-04-02 {
        description "Initial revision.";
    }

    typedef vlan-id-wildcard-type {
      type uint16 {
        range "1..4095";
      }

      description
        "This vlan-id type introduces a wildcard vlan id type.
         The value 0 is not permitted, while the value 4095
         represents a 'wildcard' value. If the value is between 1 and
         4094 inclusive, it represents an IEEE 802.1Q VLAN-ID.";
    }

    //
    //  Defaults
    //

    deviation /mef-cfm:maintenance-domain/mef-cfm:md-level {
        deviate add {
            default 3;
        }
    }

    deviation /mef-cfm:maintenance-domain/mef-cfm:maintenance-association/
mef-cfm:maintenance-association-end-point/mef-cfm:continuity-check/mef-
cfm:lowest-fault-priority-defect {
        deviate add {
            default invalid-ccm;
        }
    }

    deviation /mef-cfm:maintenance-domain/mef-cfm:maintenance-association/
mef-cfm:maintenance-association-end-point/mef-cfm:continuity-check/mef-
cfm:fng-alarm-time {
        deviate replace {
            default 300;
        }
    }
```


Copyright© 2022 Ciena® Corporation

```
    deviation /mef-cfm:maintenance-domain/mef-cfm:maintenance-association/
mef-cfm:maintenance-association-end-point/mef-cfm:continuity-check/mef-
cfm:fng-reset-time {
        deviate replace {
            default 300;
        }
    }

    //
    //  Other changes to supported config data.
    //

    // This list must have exactly one element. Its key, component-id, must
be 1.
    deviation /mef-cfm:maintenance-domain/mef-cfm:maintenance-association/
mef-cfm:component-list {
        deviate add {
            min-elements 1;
            max-elements 1;
        }
    }

    deviation /mef-cfm:maintenance-domain/mef-cfm:maintenance-association/
mef-cfm:component-list/mef-cfm:component-id {
        deviate replace {
            type uint32 {
                range "1";
            }
        }
    }

    deviation /mef-cfm:maintenance-domain/mef-cfm:maintenance-association/
mef-cfm:component-list/mef-cfm:vid {
        deviate not-supported;
    }

    // Reorder union so that string is last. The "string" type matches
    // everything, so types appearing after string will never be checked.
    deviation /mef-cfm:maintenance-domain/mef-cfm:maintenance-association/
mef-cfm:name {
        deviate replace {
            type union {
             // type mef-cfm:vlan-id-type;   // subsumed by uint16 on next line
                type uint16;
                type binary {
                    length 7;
                }
                type string;
            }
        }
    }

    deviation /mef-cfm:maintenance-domain/mef-cfm:maintenance-association/
mef-cfm:maintenance-association-end-point/mef-cfm:primary-vid {
        deviate replace {
            type vlan-id-wildcard-type;
        }
        description
          "The Primary VID of the MEP. The value 4095 indicates that the
           MEP's MA is associated with no VID";
```


Copyright© 2022 Ciena® Corporation

```
    }

    //
    //  Config data that is not supported in SAOS 6.14
    //
    deviation /mef-cfm:default-md-levels {
        deviate not-supported;
    }

    //
    //  Read-only data is not supported in SAOS 6.14
    //
    deviation /mef-cfm:configuration-error-list {
        deviate not-supported;
    }

    deviation /mef-cfm:maintenance-domain/mef-cfm:maintenance-association/
mef-cfm:maintenance-association-end-point/mef-cfm:mac-address {
        deviate not-supported;
    }

    deviation /mef-cfm:maintenance-domain/mef-cfm:maintenance-association/
mef-cfm:maintenance-association-end-point/mef-cfm:continuity-check/mef-
cfm:fng-state {
        deviate not-supported;
    }

    deviation /mef-cfm:maintenance-domain/mef-cfm:maintenance-association/
mef-cfm:maintenance-association-end-point/mef-cfm:continuity-check/mef-
cfm:active-defects {
        deviate not-supported;
    }

    deviation /mef-cfm:maintenance-domain/mef-cfm:maintenance-association/
mef-cfm:maintenance-association-end-point/mef-cfm:continuity-check/mef-
cfm:last-error-ccm {
        deviate not-supported;
    }

    deviation /mef-cfm:maintenance-domain/mef-cfm:maintenance-association/
mef-cfm:maintenance-association-end-point/mef-cfm:continuity-check/mef-
cfm:last-cross-connect-ccm {
        deviate not-supported;
    }

    deviation /mef-cfm:maintenance-domain/mef-cfm:maintenance-association/
mef-cfm:maintenance-association-end-point/mef-cfm:continuity-check/mef-
cfm:ccm-sequence-error-count {
        deviate not-supported;
    }

    deviation /mef-cfm:maintenance-domain/mef-cfm:maintenance-association/
mef-cfm:maintenance-association-end-point/mef-cfm:continuity-check/mef-
cfm:sent-ccms {
        deviate not-supported;
    }

    deviation /mef-cfm:maintenance-domain/mef-cfm:maintenance-association/
mef-cfm:maintenance-association-end-point/mef-cfm:loopback {
        deviate not-supported;
```


Copyright© 2022 Ciena® Corporation

```
    }

    deviation /mef-cfm:maintenance-domain/mef-cfm:maintenance-association/
mef-cfm:maintenance-association-end-point/mef-cfm:linktrace {
        deviate not-supported;
    }

    deviation /mef-cfm:maintenance-domain/mef-cfm:maintenance-association/
mef-cfm:maintenance-association-end-point/mef-cfm:remote-mep-database {
        deviate not-supported;
    }
}
```


Copyright© 2022 Ciena® Corporation

mef-cfm.yang

```
module mef-cfm {

  namespace "http://metroethernetforum.org/ns/yang/mef-cfm";
  prefix "mef-cfm";

  import ietf-yang-types {
    prefix yang;
    revision-date 2010-09-24;
  }
  import ietf-inet-types {
    prefix inet;
    revision-date 2010-09-24;
  }

  organization "Metro Ethernet Forum";
  contact
    "Web URL: http://metroethernetforum.org/
     E-mail:  mibs@metroethernetforum.org
     Postal:  Metro Ethernet Forum
              6033 W. Century Boulevard, Suite 830
              Los Angeles, CA 90045
              U.S.A.
     Phone:   +1 310-642-2800
     Fax:     +1 310-642-2808";

  description
    "This YANG module contains the management data definitions for the
     management of Ethernet Services Operations, Administration and
     Maintenance for Connectivity Fault Management.

     Reference Overview:
      A number of base documents have been used to create the CFM YANG. The
      following are the abbreviations for the baseline documents:
      [802.1AB] refers to 'Station and Media Access Control Connectivity
          Discovery', IEEE 802.1AB-2009, September 2009
      [802.1q] refers to IEEE 802.1Q-2011 'IEEE Standard for Local and
metropolitan
          area networks --Media Access Control (MAC) Bridges and Virtual
          Bridged Local Area Networks, August 2011
      [802-2001] refers to 'IEEE Standard for Local and Metropolitan Area
          Networks: Overview and Architecture', IEEE 802-2001, February 2002
      [MEF30] refers to MEF 30 'Service OAM Fault Management Implementation
          Agreement', January 2011
     [RFC 2737] refers to IETF RFC 2737 'Entity MIB (Version 2)', December 1999
      [RFC 2863] refers to IETF RFC 2863 'The Interfaces Group MIB', June 2000
      [RFC 3419] refers to IETF RFC 3419 'Textual Conventions for Transport
          Addresses', December 2002
    [Y.1731] refers to ITU-T Y.1731 'OAM functions and mechanisms for Ethernet
          based networks', July 2011";

  revision "2012-04-16" {
    description
      "Initial revision";
    reference
      "Service OAM Fault Management YANG Modules (MEF XX), April 2012";
  }

  //
```


Copyright© 2022 Ciena® Corporation

```
// Type definitions related to CFM
//

typedef mep-id-type {
  type uint16 {
    range "1..8191";
  }

  description
    "Maintenance association End Point Identifier (MEPID): A small integer,
     unique over a given Maintenance Association, identifying a
     specific MEP.";
  reference
    "[802.1q] 3.19 and 19.2.1";
}

typedef vlan-id-type {
  type uint16 {
    range "1..4094";
  }

  description
   "The VLAN-ID that uniquely identifies a VLAN.  This is the 12-bit VLAN-ID
    used in the VLAN Tag header.";

  reference
    "[802.1q] 9.6";
}

typedef port-status-type {
  type enumeration {
    enum no-status-tlv {
       description "Indicates either that no CCM has been received or that
                    no port status TLV was present in the last CCM received.";
    }
    enum blocked {
        description "Ordinary data cannot pass freely through the port on
                     which the remote MEP resides. Value of enableRmepDefect
                       is equal to false.";
    }
    enum up {
       description "Ordinary data can pass freely through the port on which
                       the remote MEP resides. Value of enableRmepDefect is
                       equal to true.";
    }
  }

  description
    "The set of values available from the Port Status TLV in CCM PDUs
     including the default no-status-tlv";

  reference
    "[802.1q] 20.19.3, 12.14.7.6.3:f
     IEEE8021-CFM-MIB.Dot1agCfmPortStatus";
}

typedef interface-status-type {
  type enumeration {
    enum no-status-tlv {
        description "Indicates either that no CCM has been received or that
```


Copyright© 2022 Ciena® Corporation

```
                        no interface status TLV was present in the last CCM
received.";
      }
      enum up {
          description "The interface is ready to pass packets.";
      }
      enum down {
          description "The interface cannot pass packets.";
      }
      enum testing {
          description "The interface is in some test mode.";
      }
      enum unknown {
          description "The interface status cannot be determined for some
reason.";
      }
      enum dormant {
          description "The interface is not in a state to pass
packets but is in
                       a pending state, waiting
for some external event.";
      }
      enum not-present {
          description "Some component of the interface is missing.";
      }
      enum lower-layer-down {
         description "The interface is down due to state of the lower layer
                       interfaces.";
      }
    }

    description
      "The set of values available from the Interface Status TLV in CCM PDUs
       including the default no-status-tlv";
    reference
      "[802.1q] 20.19.4, 12.14.7.6.3:g
       IEEE8021-CFM-MIB.Dot1agCfmInterfaceStatus";
  }

  typedef mhf-creation-type {
    type enumeration {
      enum none {
        description
          "No MHFs can be created for this VID(s).";
      }
      enum default {
        description
         "MHFs can be created for this VID(s) on any Bridge Port through which
the
            VID(s) can pass where:
             - There are no lower active MD levels; or
             - There is a MEP at the next lower active MD-level on the port.";
      }
      enum explicit {
        description
        "MHFs can be created for this VID(s) only on Bridge Ports through which
           this VID(s) can pass, and only if there is a MEP at the next
           lower active MD-level on the port.";
      }
```


Copyright© 2022 Ciena® Corporation

```
      enum defer {
        description
        "In the Maintenance Association managed object only, the control of MHF
           creation is deferred to the corresponding variable in the
           enclosing Maintenance Domain";
      }
    }

    description
     "An enumerated value indicating whether the management entity can create
      MHFs for this VID(s)";
    reference
      "[802.1q] 22.2.3, 12.14.3.1.3:d";
  }

  typedef id-permission-type {
    type enumeration {
      enum none {
        description
          "The Sender ID TLV is not to be sent.";
      }
      enum chassis {
        description
          "The Chassis ID Length, Chassis ID Subtype, and Chassis ID fields
of the
           Sender ID TLV are to be sent, but not the Management Address
           Length or Management Address fields.";
      }
      enum manage {
        description
        "The Management Address Length and Management Address of the Sender ID
           TLV are to be sent, but the Chassis ID Length is to be
           transmitted with a 0 value, and the Chassis ID Subtype and
           Chassis ID fields not sent;";
      }
      enum chassis-manage {
        description
          "The Chassis ID Length, Chassis ID Subtype, Chassis ID, Management
           Address Length, and Management Address fields are all to be
           sent.";
      }
      enum defer {
        description
         "The contents of the Sender ID TLV are determined by the Maintenance
           Domain managed object.";
      }
    }

    description
      "An enumerated value indicating what, if anything, is to be included in
       the Sender ID TLV transmitted by maintenance-points configured in the
       default Maintenance Domain";
    reference
      "[802.1q] 21.5.3, 12.14.3.1.3:e";
  }

  typedef mac-address-and-uint-type {
    type binary {
      length "8";
    }
```


Copyright© 2022 Ciena® Corporation

```
    description
      "A MAC address and a two-octet unsigned integer";
    reference
      "[802.1q] IEEE8021-CFM-MIB.Dot1agCfmMaintDomainNameType";
  }

  typedef fault-alarm-defect-type {
    type enumeration {
      enum remote-rdi {
          description "Indicates the aggregate health of the remote MEPs.";
      }
      enum remote-mac-error {
          description "Indicates that one or more of the remote MEPs is
                       reporting a failure in its Port Status TLV or
                       Interface Status TLV.";
      }
      enum remote-invalid-ccm {
          description "Indicates that at least one of the Remote MEP
                       state machines is not receiving valid CCMs
                       from its remote MEP.";
      }
      enum invalid-ccm {
          description "Indicates that one or more invalid CCMs has been
                       received and that 3.5 times that CCMs transmission
                       interval has not yet expired.";
      }
      enum cross-connect-ccm {
         description "Indicates that one or more cross connect CCMs has been
                       received and that 3.5 times of at least one of those
                       CCMs transmission interval has not yet expired.";
      }
}

    description
      "An enumerated value indicating the highest priority defect.";
    reference
      "[802.1q] 20.33.9";
  }

  typedef fault-alarm-defect-bits-type {
    type bits {
      bit remote-rdi;
      bit remote-mac-error;
      bit remote-invalid-ccm;
      bit invalid-ccm;
      bit cross-connect-ccm;
    }

    description
      "A set of bits indicating the the current defects:
        - cross-connect-ccm One or more cross connect CCMs has been received
        - invalid-ccm One or more invalid CCMs has been received
      - remote-invalid-ccm At least one of the Remote MEP state machines is not
          receiving valid CCMs from its remote MEP
       - remote-mac-error One or more of the remote MEPs is reporting a failure
          in its Port Status TLV or Interface Status
        - remote-rdi Indicates that at least one of the Remote MEP state
machines is
         receiving valid CCMs from its remote MEP that has the RDI bit set.";
    reference
```


Copyright© 2022 Ciena® Corporation

```
      "[802.1q] 20.33.9";
  }

  typedef lbm-transaction-id-type {
    type uint32;

    description
      "A loopback transaction identifier";
    reference
      "[802.1q] 21.7.3";
  }

  typedef ltm-transaction-id-type {
    type uint32;

    description
      "A linktrace transaction identifier";
    reference
      "[802.1q] 21.8.3";
  }

  typedef md-level-type {
    type int32 {
      range "0..7";
    }

    description
      "Maintenance Domain Level (MD Level) identifier.  Higher numbers
       correspond to higher Maintenance Domains, those with the greatest
       physical reach, with the highest values for customers' CFM PDUs.
       Lower numbers correspond to lower Maintenance Domains, those with
       more limited physical reach, with the lowest values for CFM PDUs
       protecting single bridges or physical links.";

    reference
      "[802.1q] 18.3, 21.4.1, IEEE8021-CFM-MIB.Dot1agCfmMDLevel";
  }

  typedef error-conditions-type {
    type bits {
      bit cfm-leak;
      bit conflicting-vids;
      bit excessive-levels;
      bit overlapped-levels;
    }

    description
      "A list of errors that may occur on creation or deletion of a MEP";

    reference
      "[802.1q] 22.2.4";
  }

  typedef priority-type {
    type uint32 {
      range "0..7";
    }

    description
      "A 3 bit priority value to be used in the VLAN tag, if present
```


Copyright© 2022 Ciena® Corporation

```
       in the transmitted frame.";
    reference
      "[802.1q] 12.14.7.3.2:e";
  }

  typedef remote-mep-state-type {
    type enumeration {
      enum idle {
        description
          "Indicates momentary state during reset.";
      }
      enum start {
        description
          "Indicates the timer has not expired since the state machine was
           reset, and no valid CCM has yet been received.";
      }
      enum failed {
        description
          "Indicates The timer has expired, both since the state machine
           was reset, and since a valid CCM was received.";
      }
      enum ok {
        description
          "Indicates The timer has not expired since a valid CCM was
           received.";
      }
    }
    description
      "An enumerated value indicating the operational state of a Remote
       MEP state machine for a remote MEP.";
    reference
      "[802.1q] 12.14.7.6.3:b
       IEEE8021-CFM-MIB.Dot1agCfmRemoteMepState";
  }

  typedef component-id-type {
    type uint32 {
      range "1..4294967295";
    }

    description
      "A Provider Backbone Bridge (PBB) can comprise a number of components,
       each of which can be managed in a manner essentially equivalent
       to an 802.1Q bridge.  In order to access these components easily,
       an index is used in a number of places. If any two lists are
       indexed by component-identifier, then entries in those
       tables indexed by the same value correspond to the same component";

    reference
      "IEEE8021-CFM-MIB.Dot1agCfmPbbComponentIdentifier";
  }

  //
  // Groupings related to CFM
  //

  grouping target-address-group {
    description
      "An indication of a destination MEP, either:
        1) The MEPID of a MEP; or
```


Copyright© 2022 Ciena® Corporation

```
        2) An Individual destination MAC address";

    reference "[802.1q] 12.14.7.3.2:b";

    choice address-type {
      case mac-address {
        leaf mac-address {
          type yang:mac-address;
          description
            "Target MAC address";
        }
      }
      case mep-id {
        leaf mep-id {
          type mep-id-type;
          description
            "Target MEP ID";
        }
      }
    }
  }

  grouping loopback-parameters-group {
    description
     "This is the group of parameters associated with Loopback sessions. It is
      used for loopback RPC input.";

    container target-address {
      description
        "Target MAC address or MEP ID for the Loopback session.";
      uses target-address-group;
    }

    leaf number-of-messages {
      type uint32 {
        range "1..1024";
      }
      default 1;

      description
        "The number of LBM transmissions in a session.";
      reference
        "[802.1q] 12.14.7.3.2:c, [MEF30] R39";
    }


    leaf data-tlv {
      type binary;

      description
        "An arbitrary amount of data to be included in a Data TLV.";
      reference
        "[802.1q] 12.14.7.3.d, IEEE8021-CFM-
MIB.dot1agCfmMepTransmitLbmDataTlv";
    }

    leaf vlan-priority {
      type priority-type;

      description
```


Copyright© 2022 Ciena® Corporation

```
        "The priority parameter to be used in the transmitted LBMs";
      reference
        "[802.1q] 12.14.7.3.2:e";
    }

    leaf vlan-drop-eligible {
      type boolean;
      default true;

      description
        "The drop eligible parameter to be used in the transmitted LBMs";
      reference
        "[802.1q] 12.14.7.3.2:e";
    }
  }

  grouping linktrace-parameters-group {
    description
     "This is the group of parameters associated with linktrace sessions. It is
       used for linktrace RPC input as well as linktrace database entries.";

    container target-address {
      description
        "Target MAC address or MEP ID for the Linktrace session.";
      uses target-address-group;
    }

    leaf transmit-ltm-flags {
      type bits {
        bit use-fdb-only;
      }

      description
        "The Flags field for LTMs transmitted by the MEP";
      reference
        "[802.1q] 12.14.7.4.2:b";
    }

    leaf default-ttl {
      type uint8;
      default 64;

      description
        "An initial value for the LTM TTL field";
      reference
        "[802.1q] 12.14.7.4.2:d";
    }
  }

  grouping md-level-group {
    description
      "Data definitions related to a default MD level";

    leaf md-level {
      type int32 {
        range "-1 | 0..7";
      }
      mandatory true;

      description
```


Copyright© 2022 Ciena® Corporation

```
        "The MD Level at which MHFs are to be created and Sender ID TLV
         transmission by those MHFs is to be controlled, for the VLAN to
         which this entry's definitions apply. If this leaf has the value
         -1, the MD Level for MHF creation for this VLAN is controlled
         by the content of the default-md-levels container. -1 is not a
         valid value for this parameter when used in the
         default-md-levels.";
      reference
        "[802.1q] 12.14.3.1.3:c";
    }

    leaf mhf-creation {
      type mhf-creation-type;
      mandatory true;

      description
       "This parameter indicates whether the management entity can create MHFs
          for this VID(s).

          The value 'defer' has different meanings depending on where the
grouping is
          used:
          - The value 'defer' is not allowed when this grouping is used in the
              'default-md-levels' container
           - When used in a member of the 'default-md-level' list the value
'defer'
             means that MHF creation for the VLAN is controlled by the
corresponding
             'mhf-creation' leaf in the 'default-md-levels' container.";

      reference
        "[802.1q] 12.14.3.1.3:d";
    }

    leaf default-id-permission {
      type id-permission-type;
      mandatory true;

      description
        "This parameter indicates what, if anything, is to be included in the
         Sender ID TLV transmitted by MPs configured in a default MD Level";
      reference
        "[802.1q] 12.14.3.1.3:e";
    }
  }

  grouping port-id-tlv-group {
    description
      "Data definitions associated with the Port ID TLV";

    reference
      "[802.1AB] 9.5.3";

    choice port-id-subtype {
      leaf interface-alias {
        type string {
          length "0..64";
        }
        description
          "The ifAlias field from the Interfaces Group MIB";
```


Copyright© 2022 Ciena® Corporation

```
        reference
          "[RFC2863]";
      }

      leaf port-component {
        type string {
          length "0..32";
        }
        description
          "EntPhysicalAlias when entPhysClass has a value of port(10) or
           backplane(4)";
        reference
          "[RFC2737]";
      }
      leaf mac-address {
        type yang:mac-address;
        description
          "A MAC address";
      }
      leaf network-address {
        type string;
        description
          "network-address is an octet string that identifies a particular
network
           address family and an associated network address that are
           encoded in network octet order. An IP address, for example,
           would be encoded with the first octet containing the IANA
           Address Family Numbers enumeration value for the specific
           address type and octets 2 through n containing the address
           value";
        reference
          "[802.1AB] Table 9.2";
      }

      leaf interface-name {
        type string {
          length "0..64";
        }
        description
          "The ifName field from the Interfaces Group MIB";
        reference
          "[RFC2863]";
      }
      leaf agent-circuit-id {
        type string;
        description
          "Agent circuit ID";
        reference
          "[RFC3046]";
      }
      leaf local {
        type string;
        description
          "A locally defined identifier";
        reference
          "[802.1AB] Table 9.3";
      }
    }
  }
```


Copyright© 2022 Ciena® Corporation

```
  grouping sender-id-tlv-group {
    description
      "Data definitions associated with the Sender ID TLV";

    reference
      "[802.1q] 21.5.3";

    choice chassis-id-subtype {
      description
       "The chassis-id-subtype contains the chassis ID entity that is listed in
        the chassis ID field. This is a combination of the 'Chassis ID Subtype'
          and 'chsssis ID' fields";

      reference
        "[802.1AB] 9.5.2.2";

      leaf chassis-component {
        type string {
          length "0..32";
        }
        description
        "This leaf contains the content of EntPhysicalAlias when entPhysClass
has
           a value of chassis(3)'";
        reference
          "[RFC2737]";
      }

      leaf interface-alias {
        type string {
          length "0..64";
        }
        description
          "The IfAlias field from the Interfaces Group MIB";
        reference
          "[RFC2863]";
      }

      leaf port-component {
        type string {
          length "0..32";
        }
        description
        "The EntPhysicalAlias from the Interfaces Group MIB when entPhysClass
has
           a value of port(10) or backplane(4)";
        reference
          "[RFC2737]";
      }

      leaf mac-address-type {
        type yang:mac-address;

        description
          "An IEEE 802-2001 Ethernet MAC Address";
        reference
          "[802-2001]";
      }

      leaf network-address {
```


Copyright© 2022 Ciena® Corporation

```
        type string;
        description
          "network-address is an octet string that identifies a particular
network
           address family and an associated network address that are encoded in
          network octet order. An IP address, for example, would be encoded with
          the first octet containing the IANA Address Family Numbers enumeration
           value for the specific address type and octets 2 through n containing
             the address value";
        reference
          "[802.1AB] Table 9.2";
      }

      leaf interface-name {
        type string {
          length "0..64";
        }
        description
          "IfName";
        reference
          "[RFC2863]";
      }

      leaf local {
        type string;
        description
          "A locally assigned alpha-numeric string.";
        reference
          "[802.1AB] Table 9.2";
      }
    }

    container management-address {
      description
        "Data definitions related to the management address fields defined
         in a Sender ID TLV";
      reference
        "[802.1] Section 21.5.3";
      choice management-address {
        case udp-ipv4 {
          description
            "Represents an IPv4 UDP transport address consisting of an IPv4
address,
             and a port number.";
          reference
            "[RFC3419] TRANSPORT-ADDRESS-MIB.transportDomainUdpIpv4";
          leaf udp-ipv4-address {
            type inet:ipv4-address;
          }
          leaf udp-ipv4-port {
            type inet:port-number;
          }
        }

        case udp-ipv6 {
          description
            "Represents an IPv6 UDP transport address consisting of an IPv6
address,
             and a port number.";
          reference
```


Copyright© 2022 Ciena® Corporation

```
            "[RFC3419] TRANSPORT-ADDRESS-MIB.transportDomainUdpIpv6";
          leaf udp-ipv6-address {
            type inet:ipv6-address;
          }
          leaf udp-ipv6-port {
            type inet:port-number;
          }
        }

        case udp-ipv4z {
          description
           "Represents a UDP transport address consisting of an IPv4 address,
a zone
             index and a port number.";
          reference
            "[RFC3419] TRANSPORT-ADDRESS-MIB.transportDomainUdpIpv4z";
          leaf udp-ipv4z-address {
            type inet:ipv4-address;
          }
          leaf udp-ipv4z-index {
            type uint32;
          }
          leaf udp-ipv4z-port {
            type inet:port-number;
          }
        }

        case udp-ipv6z {
          description
           "Represents a UDP transport address consisting of an IPv6 address,
a zone
             index and a port number.";
          reference
            "[RFC3419] TRANSPORT-ADDRESS-MIB.transportDomainUdpIpv6z";
          leaf udp-ipv6z-address {
            type inet:ipv6-address;
          }
          leaf udp-ipv6z-index {
            type uint32;
          }
          leaf udp-ipv6z-port {
            type inet:port-number;
          }
        }

        case tcp-ipv4 {
          description
            "Represents an IPv4 TCP transport address consisting of an IPv4
address,
             and a port number.";
          reference
            "[RFC3419] TRANSPORT-ADDRESS-MIB.transportDomainTcpIpv4";
          leaf tcp-ipv4-address {
            type inet:ipv4-address;
          }
          leaf tcp-ipv4-port {
            type inet:port-number;
          }
        }
```


Copyright© 2022 Ciena® Corporation

```
        case tcp-ipv6 {
          description
            "Represents an IPv6 TCP transport address consisting of an IPv6
address,
             and a port number.";
          reference
            "[RFC3419] TRANSPORT-ADDRESS-MIB.transportDomainTcpIpv6";
          leaf tcp-ipv6-address {
            type inet:ipv6-address;
          }
          leaf tcp-ipv6-port {
            type inet:port-number;
          }
        }

        case tcp-ipv4z {
          description
            "Represents a TCP IPv4 transport address consisting of an IPv4
address, a
             zone index and a port number.";
          reference
            "[RFC3419] TRANSPORT-ADDRESS-MIB.transportDomainTcpIpv4z";
          leaf tcp-ipv4z-address {
            type inet:ipv4-address;
          }
          leaf tcp-ipv4z-index {
            type uint32;
          }
          leaf tcp-ipv4z-port {
            type inet:port-number;
          }
        }

        case tcp-ipv6z {
          description
            "Represents a TCP IPv6 transport address consisting of an IPv6
address,
             a zone index and a port number.";
          reference
           "[RFC3419] TRANSPORT-ADDRESS-MIB.transportDomainTcpIpv6z";
          leaf tcp-ipv6z-address {
            type inet:ipv6-address;
          }
          leaf tcp-ipv6z-index {
            type uint32;
          }
          leaf tcp-ipv6z-port {
            type inet:port-number;
          }
        }

        case sctp-ipv4 {
          description
            "Represents an IPv4 SCTP transport address consisting of an IPv4
address,
             and a port number.";
          reference
           "[RFC3419] TRANSPORT-ADDRESS-MIB.transportDomainSctpIpv4";
          leaf sctp-ipv4-address {
            type inet:ipv4-address;
```


Copyright© 2022 Ciena® Corporation

```
          }
          leaf sctp-ipv4-port {
            type inet:port-number;
          }
        }

        case sctp-ipv6 {
          description
            "Represents an IPv6 SCTP transport address consisting of an IPv6
address,
             and a port number.";
          reference
           "[RFC3419] TRANSPORT-ADDRESS-MIB.transportDomainSctpIpv6";
          leaf sctp-ipv6-address {
            type inet:ipv6-address;
          }
          leaf sctp-ipv6-port {
            type inet:port-number;
          }
        }

        case sctp-ipv4z {
          description
            "Represents an SCTP IPv4 transport address consisting of an IPv4
             address, a zone index and a port number.";
          reference
           "[RFC3419] TRANSPORT-ADDRESS-MIB.transportDomainSctpIpv4z";
          leaf sctp-ipv4z-address {
            type inet:ipv4-address;
          }
          leaf sctp-ipv4z-index {
            type uint32;
          }
          leaf sctp-ipv4z-port {
            type inet:port-number;
          }
        }

        case sctp-ipv6z {
          description
            "Represents an SCTP IPv6 transport address consisting of an IPv6
             address, a zone index and a port number.";
          reference
       "[RFC3419] TRANSPORT-ADDRESS-MIB.transportDomainSctpIpv6z";
          leaf sctp-ipv6z-address {
            type inet:ipv6-address;
          }
          leaf sctp-ipv6z-index {
            type uint32;
          }
          leaf sctp-ipv6z-port {
            type inet:port-number;
          }
        }

        case local {
          leaf local-address {
            type string {
              length '1..255';
            }
```


Copyright© 2022 Ciena® Corporation

```
            description
              "Represents a POSIX Local IPC transport address.";
          }
        }

        case udp-dns {
          leaf udp-dns-address {
            type string {
              length "1..255";
            }
            description
              "The UDP transport domain using fully qualified domain names.
Represents
               a DNS domain name followed by a colon ':' (ASCII character
               0x3A) and a port number in ASCII.  The name SHOULD be fully
               qualified whenever possible.";
            reference
          "[RFC3419] TRANSPORT-ADDRESS-MIB.transportDomainUdpDns";
           }
        }
        case tcp-dns {
          leaf tcp-dns-address {
            type string {
              length "1..255";
            }
            description
              "The TCP transport domain using fully qualified domain names.
Represents
               a DNS domain name followed by a colon ':' (ASCII character
               0x3A) and a port number in ASCII.  The name SHOULD be fully
               qualified whenever possible.";
            reference
          "[RFC3419] TRANSPORT-ADDRESS-MIB.transportDomainTcpDns";
           }
        }
        case sctp-dns {
          leaf sctp-dns-address {
            type string {
              length "1..255";
            }
            description
              "The SCTP transport domain using fully qualified domain names.
               Represents a DNS domain name followed by a colon ':' (ASCII
              character 0x3A) and a port number in ASCII.  The name SHOULD be
               fully qualified whenever possible.";
            reference
          "[RFC3419] TRANSPORT-ADDRESS-MIB.transportDomainSctpDns";
           }
        }
      }
    }
  }
  grouping maintenance-domain-reference {
    description
      "This grouping uniquely identifies a maintenance domain.";

    leaf maintenance-domain {
      type leafref {
        path "/mef-cfm:maintenance-domain/mef-cfm:id";
      }
```


Copyright© 2022 Ciena® Corporation

```
      description
        "A reference to a specific Maintenance Domain.";
    }
  }

  grouping maintenance-association-reference {
    description
     "This grouping uniquely identifies a maintenance association. It consists
      of a maintence-domain-reference and a maintenance-association leafref";

    uses maintenance-domain-reference;

    leaf maintenance-association {
      type leafref {
      path "/mef-cfm:maintenance-domain[mef-cfm:id=current()/../maintenance-
domain]/mef-cfm:maintenance-association/mef-cfm:id";
      }
     description
        "A reference to a specific Maintenance Association.";
    }
  }

  grouping maintenance-association-end-point-reference {
    description
     "This grouping uniquely identifies a maintenance association. It consists
      of a maintence-association-reference and a maintenance-association-end-
point
       leafref";

    uses maintenance-association-reference;

    leaf maintenance-association-end-point {
      type leafref {
      path "/mef-cfm:maintenance-domain[mef-cfm:id=current()/../maintenance-
domain]/mef-cfm:maintenance-association[mef-cfm:id=current()/../maintenance-
association]/mef-cfm:maintenance-association-end-point/mef-cfm:mep-
identifier";
      }
     description
        "A reference to a specific Maintenance association End Point.";
    }
  }

  //
  // Data definitions related to CFM
  //

  container default-md-levels {
    description
      "Data definitions related to the global default MD level and the default
       MD levels associated with specific VIDs";

    uses md-level-group;

    list default-md-level {
      description
        "Data definitions related to the default MD level associated with
specific
         VIDs";
```


Copyright© 2022 Ciena® Corporation

```
      key "primary-vid component-id";
      leaf primary-vid {
        type vlan-id-type;

        description
          "The primary VID on the MD";
        reference
          "[802.1q] 12.14.3.1.3:a";
      }

      leaf component-id {
        type component-id-type;

        description
          "The bridge component within the system to which the information
           in this default-md-level applies";

        reference
          "[802.1q] IEEE8021-CFM-V2-MIB.ieee8021CfmDefaultMdComponentId";
      }

      leaf-list vid {
        type vlan-id-type;
        min-elements 1;
        ordered-by user;

        description
          "A list of VIDs associated with the MHF";
        reference
          "[802.1q] 12.14.3.1.3:a";
      }

      leaf status {
        type boolean;

        description
        "State of this list entry. The value 'true' if there is no entry in the
           maintenance-assoication list entry defining an MA for the
           same VLAN ID and MD Level as this table's entry, and on which
           MA an Up MEP is defined, else 'false'.";
        reference
          "[802.1q] 12.14.3.1.3:b";
      }

      uses md-level-group;
    }
  }

  list configuration-error-list {
    key "vlan-identifier interface";
    config false;

    description
      "The Configuration Error List managed object provides a list of Bridge
       Ports, aggregated ports, and VIDs that are incorrectly
       configured.";

    leaf vlan-identifier {
      type vlan-id-type;
```


Copyright© 2022 Ciena® Corporation

```
      description
        "A VLAN identifier specifying which VLAN to check for Bridge Ports or
         aggregated ports in error";
      reference
        "[802.1q] 12.14.4.1.2:a";
    }

    leaf interface {
      type string;

      description
        "An interface, either a Bridge Port or an aggregated IEEE 802.3 port
         within a Bridge Port. The value is expected to uniquely identify a
         single interface on the system. The structure of the identifier is
         implementation specific.";
      reference
        "[802.1q] 12.14.4.1.2:b";
    }

    leaf error-conditions {
      type error-conditions-type;
      mandatory true;

      description
        "A list of possible errors on a Bridge Port.";
      reference
        "[802.1q] 12.14.4.1.3:b";
    }
  }

  list maintenance-domain {
    key id;

    description
     "A Maintenance Domain managed object is required in order to create an MA
       with a MAID that includes that Maintenance Domain's Name. From
       this Maintenance Domain managed object, all Maintenance
       Association managed objects associated with that Maintenance
       Domain managed object can be accessed, and thus controlled.";
    reference
      "[802.1q] 12.14.5";

    leaf id {
      type string;
      description
        "A unique identifier of a Maintenance Domain";
      reference
        "[802.1q] 12.14.5";
    }

    leaf name-type {
      type enumeration {
        enum none {
            description "No format specified.";
        }
        enum domain-name {
            description "Domain Name like string, globally unique
text string
                         derived from a DNS name.";
        }
```


Copyright© 2022 Ciena® Corporation

```
        enum mac-address-and-uint {
            description "MAC address + 2-octet (unsigned) integer.";
        }
        enum character-string {
            description "RFC2579 DisplayString, except that the
character
                         codes 0-31 (decimal) are not used.";
        }
      }

      description
       "Type discriminator for the value union described in the 'name' leaf.";
      reference
        " IEEE8021-CFM-MIB.Dot1agCfmMaintDomainNameType";

    }

    leaf name {
      type union {
        type inet:host;
        type mac-address-and-uint-type;
        type string;
      }

      description
        "The value union for the Maintenance Domain Name. The type must
         correspond to the configuration name-type according to the
         following:
          - If the name-type is 'domain-name ' then the name must be of type
            inet:host
          - If the name-type is 'mac-address-and-uint' then the name must
            be of type mac-address-and-uint-type
          - If the name-type is 'character-string' then the name must be of
            type string
          - If the name-type is 'none' then the name must not be defined";
      reference
        "[802.1q] 21.6.5.1 (Table 21-19), 12.14.5.3.2:a";
    }

    leaf md-level {
      type md-level-type;

      description
        "Integer identifying the Maintenance Domain Level (MD Level).  Higher
        numbers correspond to higher Maintenance Domains, those with the
        greatest physical reach, with the highest values for customers'
        CFM PDUs.  Lower numbers correspond to lower Maintenance
        Domains, those with more limited physical reach, with the lowest
        values for CFM PDUs protecting single bridges or physical links.";

      reference
        "[802.1q] 12.14.5.1.3:b";
    }

    leaf mhf-creation {
      type mhf-creation-type;
      mandatory true;

      description
        "A value indicating if the Management entity can create MHFs
```


Copyright© 2022 Ciena® Corporation

```
          (MIP Half Function) for this VID at this MD Level. The value
          'defer' means that MHF creation for the VLAN is controlled by
          the corresponding 'mhf-creation' leaf in the 'default-md-level'
          list entry.";
        reference
          "[802.1q] 22.2.3, 12.14.5.1.3:c";
      }

      leaf id-permission {
        type id-permission-type;
        mandatory true;

        description
         "This parameter indicates what, if anything, is to be included in the
          Sender ID TLV. Since, in this variable, there is no encompassing
          Maintenance Domain, the value 'defer' is not allowed";
        reference
          "[802.1q] 21.5.3, 12.14.6.1.3:d";
      }

      list maintenance-association {
        key id;

        description
          "This list represents Maintenance Entity Groups (Y.1731) or
           Maintenance Associations (802.1ag). MEGs/MAs are sets of
           MEPs, each configured to the same service inside a common
           OAM domain.";

        leaf id {
          type string;
          description
            "A unique identifier of a Maintenance Association";
          reference
            "[802.1q] 12.14.6";
        }

        leaf name-type {
          type enumeration {
            enum primary-vid {
                description "Specifies the Primary VLAN ID.";
            }
            enum character-string {
                description "RFC2579 DisplayString, except that the
character
                             codes 0-31 (decimal) are not used.";
            }
            enum uint16 {
                description "2-octet integer/big endian.";
            }
            enum rfc2685-vpn-id {
                description "Specifies the RFC 2685 VPN ID.";
            }
          }

          description
            "Name type discriminator for the 'name' leaf.";
          reference
            "[802.1q] 12.14.5.3.2:b, Table 21-20
             IEEE8021-CFM-MIB.Dot1agCfmMaintAssocNameType";
```


Copyright© 2022 Ciena® Corporation

```
      }

      leaf name {
        type union {
          type vlan-id-type;
          type string;
          type uint16;
          type binary {
            length 7;
          }
        }

        description
         "The value union for the Maintenance Association Name. The type must
           correspond to the configuration name-type according to the
           following:
            - If the name-type is 'primary-vid' then the name must be of type
               vlan-id-type
            - If the name-type is 'character-string' then the name must be of
               type string
           - If the name-type is 'uint16' then the name must be of type uint16
          - If the name-type is 'rfc2685-vpn-id' then the name must be of type
             binary";
        reference
          "[802.1q] 12.14.5.3.2:b, Table 21-20";
      }

      list component-list {
        key component-id;

        description
          "A list of components each of which can be managed in a manner
           essentially equivalent to an 802.1Q bridge.";
        reference
            "[802.1q] IEEE8021-CFM-V2-MIB.ieee8021CfmMaCompTable";

        leaf component-id {
          type uint32;

          description
            "The bridge component within the system to which the information
             in this maintenance-association applies";

          reference
            "[802.1q] IEEE8021-CFM-V2-MIB.ieee8021CfmMaComponentId";
        }


        leaf-list vid {
          type vlan-id-type;
          min-elements 1;
          ordered-by user;

          description
           "The VID(s) monitored by this MA, or 0, if the MA is not attached
to any
              VID. The first VID returned is the MA's Primary VID";
          reference
            "[802.1q] 12.14.5.3.2:b";
        }
```


Copyright© 2022 Ciena® Corporation

```
        leaf mhf-creation {
          type mhf-creation-type;
          default defer;
          description
           "An enumerated value indicating whether the management entity can
create
              MHFs for this MA. A value of 'defer' here means that control of
              MHF creation is deferred to the corresponding variable in
              the enclosing maintenance-domain.";
          reference
            "[802.1q] 12.14.6.1.3:c";
        }

        leaf id-permission {
          type id-permission-type;
          default defer;

          description
           "This parameter indicates what, if anything, is to be included in the
              Sender ID TLV transmitted by MPs configured in this MA. A
              value of 'defer' means that the contents of the Sender ID
            TLV are determined by the enclosing maintenance-domain instance,";
          reference
            "[802.1q] 12.14.6.1.3:d";
        }
      }
      leaf ccm-interval {
        type enumeration {
          enum invalid {
              description "No CCMs are sent (disabled).";
          }
          enum 3.3ms {
              description "CCMs are sent every 3 1/3 milliseconds (300Hz).";
          }
          enum 10ms {
              description "CCMs are sent every 10 milliseconds.";
          }
          enum 100ms {
              description "CCMs are sent every 100 milliseconds.";
          }
          enum 1s {
              description "CCMs are sent every 1 second.";
          }
          enum 10s {
              description "CCMs are sent every 10 seconds.";
          }
          enum 1min {
              description "CCMs are sent every minute.";
          }
          enum 10min {
              description "CCMs are sent every 10 minutes.";
          }
        }
        mandatory true;

        description
          "The interval between CCM transmissions to be used by all MEPs in
the MA";
        reference
```


Copyright© 2022 Ciena® Corporation

```
          "[802.1q] 12.14.6.1.3:e
           IEEE8021-CFM-MIB.Dot1agCfmCcmInterval";
      }

      leaf-list remote-meps {
        type mep-id-type;

        description
          "A list of the MEPIDs of the MEPs in the MA.";
        reference
          "[802.1q] 12.14.6.1.3:g";
      }

      list maintenance-association-end-point {
        key "mep-identifier";

        description
       "The list of Maintenance association End Points in a specific Maintance
As-sociation.";

        leaf mep-identifier {
          type mep-id-type;

          description
            "Integer that is unique among all the MEPs in the same MA. Other
             definition is: a small integer, unique over a given
             Maintenance Association, identifying a specific Maintenance
             association End Point.";
          reference
            "[802.1q] 12.14.6.3.2:b";
        }

        leaf interface {
          type string;
          mandatory true;

          description
          "An interface, either a Bridge Port or an aggregated IEEE 802.3 port
          within a Bridge Port, to which the MEP is attached. Each interface in
             the system is uniquely identified by an interface-name. The
structure
          and content of the name is outside the scope of this specification.";
          reference
            "[802.1q] 12.14.7.1.3:b";
        }

        leaf direction {
          type enumeration {
            enum up {
            description "Indicates when CFM frames are transmitted towards and
                              received from the bridging function.";
            }
            enum down {
           description "Indicates when CFM frames are transmitted towards and
                            received from the wire.";
            }
          }
          mandatory true;

          description
```


Copyright© 2022 Ciena® Corporation

```
      "A value indicating the direction in which the MEP faces on the
       interface.";

    reference
      "[802.1q] 12.14.7.1.3:c";
  }

  leaf primary-vid {
    type vlan-id-type;
    mandatory true;

    description
     "The Primary VID of the MEP. The value 0 indicates that either the
       Primary VID is that of the MEP's MA or that the MEP's MA is
       associated with no VID";
    reference
      "[802.1q] 12.14.7.1.3:d";
  }

  leaf administrative-state {
    type boolean;
    mandatory true;

    description
      "The administrative state of the MEP";
    reference
      "[802.1q] 12.14.7.1.3:e";
  }

  leaf mac-address {
    type yang:mac-address;
    config false;

    description
      "The MAC address of the MEP";
    reference
      "[802.1q] 12.14.7.1.3:i";
  }

  leaf ccm-ltm-priority {
    type priority-type;
    mandatory true;

    description
     "The priority parameter for CCMs and LTMs transmitted by the MEP.";
    reference
      "[802.1q] 12.14.7.1.3:h";
  }

  container continuity-check {
    description
      "Data definitions related to the Continuity Check function.";

    leaf cci-enabled {
      type boolean;
      default false;

      description
       "An indicator of whether the MEP is or is not to generate CCMs";
      reference
```


Copyright© 2022 Ciena® Corporation

```
              "[802.1q] 12.14.7.1.3:g";
          }

          leaf fng-state {
            type enumeration {
              enum reset {
                  description "No defect has been present since the
                               fng-reset-time timer expired, or since
                               the state machine
was last reset.";
              }
              enum defect {
                 description "A defect is present, but not for a long enough
                               time to be reported (fng-alarm-time).";
              }
              enum report-defect {
                  description "A momentary state during which the defect is
                               reported by sending a
fault-alarm notification,
                               if that action is enabled.";
              }
              enum defect-reported {
                  description "A defect is present, and some defect
has been
                               reported.";
              }
              enum defect-clearing {
               description "No defect is present, but the fng-reset-time timer
                               has
not yet expired.";
              }
            }
            default reset;
            config false;

            description
             "A value indicating the current state of the MEP Fault Notification
                Generator state machine";
            reference
              "[802.1q] 12.14.7.1.3:f, 20.35
               IEEE8021-CFM-MIB.Dot1agCfmFngState";
          }

          leaf lowest-fault-priority-defect {
            type fault-alarm-defect-type;

            description
              "The lowest priority defect that is allowed to generate a Fault
               Alarm. The non-existence of this leaf means that no
               defects are to be reported";
            reference
              "[802.1q] 12.14.7.1.3:k";
          }

          leaf highest-priority-defect-found {
            type fault-alarm-defect-type;
            config false;

            description
              "Value indicating the highest-priority defect that has been
```


Copyright© 2022 Ciena® Corporation

```
             present since the MEP Fault Notification Generator state machine
                was last in the FNG_RESET state The non-existence of this
                leaf represents that no defect has been present since the
                last FNG_RESET state";
              reference
                "[802.1q] 12.14.7.1.3:n";
            }

            leaf fng-alarm-time {
              type yang:timeticks;
              default 250;

              description
                "The time that defects must be present before a Fault Alarm is
issued";
              reference
                "[802.1q] 12.14.7.1.3:l, 20.33.3";
            }

            leaf fng-reset-time {
              type yang:timeticks;
              default 1000;

              description
                "The time that defects must be absent before resetting a Fault
Alarm";
              reference
                "[802.1q] 12.14.7.1.3:m, 20.33.4";
            }

            leaf active-defects {
              type fault-alarm-defect-bits-type;
              config false;
              mandatory true;

              description
                "A bit field of potential active defects. The values are the
same as the
                 content of the fault-alarm-defect-type.";
              reference
                "[802.1q] 12.14.7.1.3:o, p, q, r, s";
            }

            leaf last-error-ccm {
              type binary {
                length "1..1522";
              }
              config false;

              description
                "The last-received CCM that triggered an invalid-ccm fault";
              reference
                "[802.1q] 12.14.7.1.3:t";
            }

            leaf last-cross-connect-ccm {
              type binary {
                length "1..1522";
              }
              config false;
```


Copyright© 2022 Ciena® Corporation

```
      description
      "The last-received CCM that triggered a cross-connect-ccm fault";
      reference
        "[802.1q] 12.14.7.1.3:u";
    }

    leaf ccm-sequence-error-count {
      type yang:counter32;
      config false;
      mandatory true;

      description
      "The total number of out-of-sequence CCMs received from all remote
         MEPs.";
      reference
        "[802.1q] 12.14.7.1.3:v";
    }

    leaf sent-ccms {
      type yang:counter32;
      config false;
      mandatory true;

      description
        "The total number of CCMs transmitted.";
      reference
        "[802.1q] 12.14.7.1.3:w";
    }
  }

  container loopback {
    description
      "Data definitions related to the Loopback function.";

    config false;

    leaf replies-received {
      type yang:counter32;
      config false;
      mandatory true;

      description
        "The total number of valid, in-order LBRs received.";
      reference
        "[802.1q] 12.14.7.1.3:y";
    }

    leaf replies-transmitted {
      type yang:counter32;
      config false;
      mandatory true;

      description
        "The total number of LBRs transmitted.";
      reference
        "[802.1q] 12.14.7.1.3:ad";
    }

    leaf out-of-order-replies-received {
```


Copyright© 2022 Ciena® Corporation

```
            type yang:counter32;
            config false;
            mandatory true;

            description
              "The total number of valid, out-of-order LBRs received.";
            reference
              "[802.1q] 12.14.7.1.3:z";
          }

          leaf bad-msdu {
            type yang:counter32;
            config false;

            description
              "The total number of LBRs received whose mac_service_data_unit
did not
               match (except for the OpCode) that of the corresponding LBM";
            reference
              "[802.1q] 12.14.7.1.3:aa";
          }
        }

        container linktrace {
          description
            "Data definitions related to the Linktrace function.";

          config false;

          leaf unexpected-replies-received {
            type yang:counter32;
            mandatory true;

            description
              "The total number of unexpected LTRs received.";
            reference
              "[802.1q] 12.14.7.1.3:ac";
          }

          container linktrace-database {
            description
             "Data definitions related to the Linktrace database for a specific
MEP";

            list linktrace {
              key "transaction-id";
              config false;

              description
                "The list of entries in a Linktrace database.";

              leaf transaction-id {
                type ltm-transaction-id-type;

                description
               "The LTM Transaction Identifier to which the LTR entries will be
                    attached";
                reference
                  "[802.1q] 12.14.7.5.2:b";
              }
```


Copyright© 2022 Ciena® Corporation

```
uses linktrace-parameters-group;

list reply {
  key reply-order;

  description
    "The list of LTRs associated with a specific Linktrace
     transaction.";
  leaf reply-order {
    type uint32;

    description
  "An index to distinguish among multiple LTRs with the same LTR
       transaction-id field value. reply-order are
       assigned sequentially from 1, in the order that the
       Linktrace Initiator received the LTR";
    reference "[802.1q] 12.14.7.5.2:c";
  }

  leaf reply-ttl {
    type uint32;

    description
      "The integer Reply TTL field value returned in the LTR.";
    reference
      "[802.1q] 12.14.7.5.3:b";
  }

  leaf forwarded {
    type boolean;

    description
      "A Boolean value stating whether an LTM was forwarded by
       the responding MP.";
    reference
      "[802.1q] 12.14.7.5.3:c";
  }

  leaf terminal-mep {
    type boolean;

    description
    "A Boolean value stating whether the forwarded LTM reached
       a MEP for its MA.";
    reference
      "[802.1q] 12.14.7.5.3:d";
  }

  leaf last-egress-identifier {
    type binary {
      length 8;
    }

    description
     "An octet string holding the Last Egress Identifier field
       returned in the LTR Egress Identifier TLV of the LTR.";
    reference
      "[802.1q] 12.14.7.5.3:e";
  }
```


Copyright© 2022 Ciena® Corporation

```
leaf next-egress-identifier {
  type binary {
    length 8;
  }

  description
   "An octet string holding the Next Egress Identifier field
    returned in the LTR Egress Identifier TLV of the LTR";
  reference
    "[802.1q] 12.14.7.5.3:f";
}

leaf ltr-relay {
  type enumeration {
    enum hit {
       description "Indicates the LTM reached an MP whose MAC
                    address matches the target MAC address.";
    }
    enum filtering-database {
     description "Indicates the Egress Port was determined by
                      consulting the Filtering Database.";
    }
    enum mip-ccm-database {
     description "Indicates the Egress Port was determined by
                      consulting the MIP CCM Database.";
    }
  }

  description
   "An enumerated value indicating the value returned in the
    Relay Action field.";
  reference
    "[802.1q] 12.14.7.5.3:g, Table 21-27
     IEEE8021-CFM-MIB.Dot1agCfmRelayActionFieldValue";
}

uses sender-id-tlv-group;

leaf ingress-action {
  type enumeration {
    enum ok {
       description "Indicates the target data frame would be
                    passed through to the MAC Relay Entity.";
    }
    enum down {
      description "Indicates the Bridge Ports MAC Operational
                      parameter is false.";
    }
    enum blocked {
       description "Indicates the target data frame would not
                    be forwarded if received on this Port due
                      to active topology enforement.";
    }
    enum vid {
       description "Indicates the ingress port is not in the
                      member set of the LTMs VID, and ingress
                    filtering is enabled, so the target data
                frame would be filtered by ingress filtering.";
    }
```


Copyright© 2022 Ciena® Corporation

```
    }

    description
     "An enumerated value indicating the value returned in the
     Ingress Action field. This leaf is not present if no value
       is returned in the LTR.";
    reference
      "[802.1q] 12.14.7.5.3:k, Table 21-30
       IEEE8021-CFM-MIB.Dot1agCfmIngressActionFieldValue";
  }

  leaf ingress-mac {
    type yang:mac-address;

    description
    "The MAC address returned in the Ingress MAC Address field.
   This leaf is not present if no value is returned in the LTR.";
    reference
      "[802.1q] 12.14.7.5.3:l";
  }

  container ingress-port-id {
    uses port-id-tlv-group;

    description
  "The Ingress Port ID field and the corresponding port ID value.
   This leaf is not present if no value is returned in the LTR.";
    reference
      "[802.1q] 12.14.7.5.3:m";
  }

  leaf egress-action {
    type enumeration {
      enum ok {
        description "Indicates the targeted data frame would be
                     forwarded.";
      }
      enum down {
      description "Indicates the Egress Port can be identified,
                     but that Bridge Ports MAC Operational
                     parameter is false.";
      }
      enum blocked {
      description "Indicates the Egress Port can be identified,
                  but the data frame would not pass through the
                     the Egress Port due to active topology
                management, i.e., the Bridge Port is not in the
                     Forwarding state.";
      }
      enum vid {
      description "Indicates the Egress Port can be identified,
                   but the Bridge Port is not in the LTMs VIDs
                   member set, so would be filtered by egress
                     filtering.";
      }
    }

    description
     "An enumerated value indicating the value returned in the
      Egress Action field. This leaf is not present if no value
```


Copyright© 2022 Ciena® Corporation

```
                is returned in the LTR.";
              reference
                "[802.1q] 12.14.7.5.3:o, Table 21-32
                 IEEE8021-CFM-MIB.Dot1agCfmIngressActionFieldValue";
            }

            leaf egress-mac {
              type yang:mac-address;

              description
               "The MAC address returned in the Egress MAC Address field.
             This leaf is not present if no value is returned in the LTR.";
              reference
                "[802.1q] 12.14.7.5.3:p";
            }

            container egress-port-id {
              uses port-id-tlv-group;

              description
            "The Egress Port ID field and the corresponding port ID value.
             This leaf is not present if no value is returned in the LTR.";
              reference
                "[802.1q] 12.14.7.5.3:p, 12.14.7.5.3:q";
            }

            leaf organization-specific-tlv {
              type binary {
                length "0|4 .. 1500";
              }
              description
             "The OUI and contents of any Organization-Specific TLVs. This
                leaf is not present if no value is returned in the LTR.";
              reference
                "[802.1q] 12.14.7.5.3:s";
            }
          }
        }
      }
    }

    container remote-mep-database {
      description
        "The MEP CCM database";

      config false;

      list remote-mep {
        description
          "The list of remote MEPs in the MEP CCM database";
        reference
          "[802.1q] 12.14.7.6";

        key remote-mep-id;

        leaf remote-mep-id {
          type mep-id-type;

          description
            "The MEPID of a remote MEP";
```


Copyright© 2022 Ciena® Corporation

```
            reference
              "[802.1q] 12.14.7.6.2";
          }

          leaf remote-mep-state {
            type remote-mep-state-type;

            description
             "An enumerated value indicating the operational state of the
              Remote MEP state machine for this remote MEP.";
            reference
              "[802.1q] 12.14.7.6.3:b
               IEEE8021-CFM-MIB.Dot1agCfmRemoteMepState";
          }

          leaf failed-ok-time {
            type yang:timeticks;

            description
           "The time (SysUpTime, IETF RFC 3418) at which the Remote MEP state
               machine last entered either the RMEP_FAILED or RMEP_OK
               state, or 0 if it has not yet entered either of those
               states";
            reference
              "[802.1q] 12.14.7.6.3:c";
          }

          leaf mac-address {
            type yang:mac-address;

            description
              "The MAC address of the remote MEP.";
            reference
              "[802.1q] 12.14.7.6.3:d";
          }

          leaf rdi {
            type boolean;

            description
            "An indication of the state of the RDI bit in the last received
CCM,
               true for RDI = 1, or false for RDI = 0 or if none has been
               received.";
            reference
              "[802.1q] 12.14.7.6.3:e";
          }

          leaf port-status-tlv {
            type port-status-type;

            description
             "The enumerated value from the Port Status TLV from the last
              CCM received from the remote MEP.";
            reference
              "[802.1q] 12.14.7.6.3:f";
          }

          leaf interface-status-tlv {
            type interface-status-type;
```

```
              description
                "The enumerated value from the Interface Status TLV from the
                 last CCM received from the remote MEP";
              reference
                "[802.1q] 12.14.7.6.3:g";
            }

            uses sender-id-tlv-group;
          }
        }
      }
    }
  }

  //
  // RPCs related to CFM
  //

  rpc transmit-loopback {
    description
      "Start a loopback session on a specific MEP

       The counters in the loopback container of the associated MEP will be
updated
      with the result of the loopback session until the configured number of
messages
       have been sent.";
    reference
      "[802.1q] 12.14.7.3";

    input {
      leaf mep-id {
        type mep-id-type;

        description
          "The ID of the MEP on which a loopback session will be started";
      }
      uses loopback-parameters-group;
    }
  }

  rpc abort-loopback {
    description
     "Administratively abort a loopback session in progress on a specific MEP";
    reference
      "[MEF30] 8.2, R28";

    input {
      leaf mep-id {
        type mep-id-type;

        description
          "The ID of the MEP on which the loopback session should be
           administratively aborted";
      }
    }
  }

  rpc transmit-linktrace {
```


Copyright© 2022 Ciena® Corporation

```
    description
      "Start a linktrace session on a specific MEP

       A list entry in the linktrace-database of the associated MEP will
       be created for each successfully started linktrace session. The
       RPC output transaction-id corresponds to the transaction-id in the
       database for the specific session.";
    reference
      "[802.1q] 12.14.7.4";

    input {
      leaf mep-id {
        type mep-id-type;

        description
          "The ID of the MEP on which a linktrace session will be started.";
      }
      uses linktrace-parameters-group;
    }

    output {
      leaf transaction-id {
        type ltm-transaction-id-type;
        description
         "The transaction identifier of the newly created linktrace session.";
        reference
          "[802.1q] 12.14.7.4.3:b";
      }
    }
  }

  //
  // Notifications related to CFM
  //

  notification fault-alarm {
    description
      "A notification that a specific MEP has a persistent defect condition.";
    reference
      "[802.1q] 12.14.7.7";

    container alarm {
      description
        "Data definitions related to a persistent defect on a specific MEP.";

      leaf mep-id {
        type mep-id-type;

        description
          "The ID of a MEP on which a persistent defect has been detected";
      }
      leaf active-defects {
        type fault-alarm-defect-bits-type;
        description
          "The currently active defects on the specific MEP.";
      }
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

mef-soam-fm-deviation.yang

```
module mef-soam-fm-deviation {

  namespace "http://www.ciena.com/ns/yang/mef-soam-fm-dev";
  prefix "mef-soam-fm-dev";

  import mef-cfm {
    prefix cfm;
  }
  import mef-soam-fm {
    prefix mef-soam-fm;
  }

  revision 2016-03-09 {
    description "Initial revision";
  }


  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:component-list/mef-soam-fm:connectivity-status-interval {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:component-list/mef-soam-fm:mep-port-status-tlv-included {

    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:component-list/mef-soam-fm:mep-interface-status-tlv-included {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-fm:operational-state {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-fm:connectivity-status {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-fm:port-status {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-fm:interface-status {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-fm:last-defect-sent {
    deviate not-supported;
  }
```


Copyright© 2022 Ciena® Corporation

```
  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-fm:rdi-transmit-status {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-fm:oam-frames-discarded {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/cfm:continuity-check/mef-soam-fm:drop-
eligible {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/cfm:loopback/mef-soam-fm:multicast-
enabled {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/cfm:loopback/mef-soam-fm:frame-size {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/cfm:loopback/mef-soam-fm:data-pattern
{
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/cfm:loopback/mef-soam-fm:test-tlv-
included {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/cfm:loopback/mef-soam-fm:test-tlv-
pattern {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/cfm:loopback/mef-soam-fm:crc-errors-in
{
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/cfm:loopback/mef-soam-fm:responses {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-fm:alarm-indication-signal {
    deviate not-supported;
  }
```


Copyright© 2022 Ciena® Corporation

```
  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-fm:locked-signal {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-fm:test-signal {
    deviate not-supported;
  }

  deviation /mef-soam-fm:create-test-signal {
    deviate not-supported;
  }

  deviation /mef-soam-fm:abort-test-signal {
    deviate not-supported;
  }

  deviation /mef-soam-fm:configuration-error-assert {
    deviate not-supported;
  }

  deviation /mef-soam-fm:configuration-error-clear {
    deviate not-supported;
  }

  deviation /mef-soam-fm:lock-alarm {
    deviate not-supported;
  }

  deviation /mef-soam-fm:ais-alarm {
    deviate not-supported;
  }


}
```


Copyright© 2022 Ciena® Corporation

mef-soam-fm.yang

```
module mef-soam-fm {

  namespace "http://metroethernetforum.org/ns/yang/mef-soam-fm";
  prefix "mef-soam-fm";

  import mef-cfm {
    prefix cfm;
    revision-date 2012-04-16;
  }
  import ietf-yang-types {
    prefix yang;
    revision-date 2010-09-24;
  }

  organization "Metro Ethernet Forum";
  contact
    "Web URL: http://metroethernetforum.org/
     E-mail:  mibs@metroethernetforum.org
     Postal:  Metro Ethernet Forum
              6033 W. Century Boulevard, Suite 830
              Los Angeles, CA 90045
              U.S.A.
     Phone:   +1 310-642-2800
     Fax:     +1 310-642-2808";

  description
    "This YANG module contains the management data definitions for the
     management of Ethernet Services Operations, Administration and
     Maintenance for Fault Management and extends the Connectivity Fault
     Management (CFM) YANG modules.

    Reference Overview:
     A number of base documents have been used to create the SOAM FM YANG. The
      following are the abbreviations for the baseline documents:
      [802.1q] refers to IEEE 802.1Q-2011 'IEEE Standard for Local and
metropolitan
          area networks --Media Access Control (MAC) Bridges and Virtual
          Bridged Local Area Networks, August 2011
      [MEF7.1] refers to MEF 7.1 'Phase 2 EMS-NMS Information Model',
          October 2009
      [MEF30] refers to MEF 30 'Service OAM Fault Management Implementation
          Agreement', January 2011
      [MEF31] refers to MEF 31 'Service OAM Fault Management Definition of
Managed
          Objects', January 2011
    [Y.1731] refers to ITU-T Y.1731 'OAM functions and mechanisms for Ethernet
          based networks', July 2011";


  revision "2012-04-16" {
    description
      "Initial revision";
    reference
      "Service OAM Fault Management YANG Modules (MEF XX), April 2012";
  }

  //
  // Type definitions related to MEF SOAM FM
```


Copyright© 2022 Ciena® Corporation

```
  //

  typedef interval-type-ais-lck-type {
    type  enumeration {
      enum one-second {
        description
          "A one second transmission interval";
      }
      enum one-minute {
        description
          "A one minute transmission interval";
      }
}

    description
      "This enumeration data type defines the AIS/LCK interval (transmission
       period for a frame)";
    reference
      "[Y.1731] sections 7.4, 7.6; [MEF7.1] III.2";
  }

  typedef operational-state-type {
    type enumeration {
      enum enabled {
        description
          "The MEP is able to provide OAM capabilities and has been set to
           active via the 'administrative-state' leaf.";
      }
      enum disabled {
        description
          "The MEP is not able to provide OAM capabilities, for example
           because it has been disabled via the administrative-state leaf,
           has detected an operational failure condition, or has failed an
           internal test.";
      }
      enum testing {
        description
          "The MEP has been placed into a test mode, either a troubleshooting
           mode or ETH-Test 'Out-of-service' mode.";
      }
      enum unknown {
        description
          "The MEP is unable to report the operational state.";
      }
    }
    description
      "This attribute indicates the operational state (current capability) of
       a MEP.";
    reference
      "[MEF7.1] 9.2.5";
  }

  typedef test-pattern-type {
    type enumeration {
      enum null-signal-without-crc-32 {
        description
          "Null signal without CRC-32";
      }
      enum null-signal-with-crc-32 {
        description
```


Copyright© 2022 Ciena® Corporation

```
          "Null signal with CRC-32";
      }
      enum prbs-2311-without-crc-32 {
        description
          "PRBS 2^31-1 without CRC-32";
      }
      enum prbs-2311-with-crc-32 {
        description
          "PRBS 2^31-1 with CRC-32";
      }
    }

    description
      "This enumeration data type indicates the type of test pattern to be
       sent in an OAM PDU Data TLV";
    reference
      "[Y.1731]";
  }

  //
  // Groupings related to MEF SOAM FM
  //

  grouping locked-signal-parameters-group {
    description
     "Data definitions related to the administrative locking state of the MEP";

    leaf state {
      type enumeration {
        enum lock {
        description
          "Indicates the resource is administratively
prohibited from use.";
        }
        enum unlock {
        description
          "Indicates the resource is not
administratively prohibited from use.";
        }
      }


      description
        "This attribute specifies the locking state. If state is 'lock',the
         MEP will be administratively locked. If state is 'unlock',
         the MEP will be administratively unlocked if previously
         locked.";
      reference
        "[MEF7.1] 9.3.4.2";
    }

    leaf interval {
      type interval-type-ais-lck-type;
      default one-second;

      description
       "This attribute specifies the ETH-LCK transmission period. The default
          value is 1 frame per second.";
      reference
        "[MEF7.1] 9.3.4.2";
```

```
    }

    leaf priority {
      type cfm:priority-type;

      description
        "This attribute specifies the priority of frames with ETH-LCK
         information.";
      reference
        "[MEF7.1] 9.3.4.2";
    }

    leaf client-meg-level {
      type cfm:md-level-type;

      description
        "The MEG/Maintenance Domain Level of the Client LCK PDU (transmitted
         level).";
      reference
        "[Y.1731] 7.6";
    }
  }

  grouping test-signal-parameters-group {
    description
      "Data definitions related to generating and receiving test signals
       (ETH-Test) frames";

    leaf is-receiver {
      type boolean;
      default true;

      description
        "Indicates the MEP is acting in the role of a receiver.";
      reference
        "[MEF7.1] 9.3.4.3";
    }

    leaf is-generator {
      type boolean;
      default false;

      description
        "Indicates the MEP is acting in the role of a generator.";
      reference
        "[MEF7.1] 9.3.4.3";
    }

    leaf test-type {
      type enumeration {
        enum in-service {
        description
          "Indicates the ETH-Test is in-service and normal client service
traffic
           is not interrupted.";
        }
        enum out-of-service {
        description
         "Indicates the ETH-Test is out-of-service and normal client service
           traffic is disrupted.";
```


Copyright© 2022 Ciena® Corporation

```
        }
      }
      default in-service;

      must "../is-generator = true" {
        error-message "The test-type parameter is only relevant for the
generator side";
      }

      description
       "This attribute specifies the type of ETH-Test to perform, whether it is
          service interrupting or not. An 'in-service' value indicates
          that the ETH-Test is in service and normal client service
          traffic is not interrupted. A 'out-of-service' value indicates
          that the ETH-Test is out of service and normal client service
          traffic is disrupted.";

      reference "[Y.1731] 7.7";
    }

    container target-address {
      uses cfm:target-address-group;

      must "../is-generator = true" {
        error-message "The target-address parameter is only relevant for the
generator side";
      }

      description
        "Unicast MAC address or MEPID of the peer MEP for which ETH-Test is
         intended. This information is configurable per operation.";
      reference
        "[Y.1731] 7.7";
    }

    leaf interval {
      type uint32 {
        range "0..60000000";
      }
      units microseconds;
      default 1000000;

      must "../is-generator = true" {
        error-message
         "The interval parameter is only relevant for the generator side";
      }

      description
        "This attribute specifies the ETH-Test transmission period between
        consecutive transmitted frames in microseconds. A value of '0'
        indicates that the Test TLVs are sent as quickly as possible
        across the interface.";

      reference "[MEF7.1] 9.3.4.3";
    }

    leaf priority {
      type cfm:priority-type;

      must "../is-generator = true" {
```


Copyright© 2022 Ciena® Corporation

```
      error-message "The priority parameter is only relevant for the generator
side";
      }

      description
        "This attribute specifies the priority of frames with ETH-Test
         information";

      reference "[MEF7.1] 9.3.4.3";
    }

    container measurement-timing {
      description
        "Data definitions related to measurement timing";

      choice measurement-timing {
        mandatory true;

        description
          "On-demand measurements can be scheduled using absolute
           time or relative time
            - absolute-time specifies absolute start and stop, date and time
            - relative-time specifies an offset from current system time and
             a duration";

        case absolute-time {
          leaf start-time {
            type yang:date-and-time;

            description
             "This attribute specifies the scheduled start date/time to perform
the
                on-demand ETH-Test operations. The default value for this
                attribute is the current system date and time which
                represents an immediate time.";
          }

          leaf stop-time {
            type yang:date-and-time;

            description
            "This attribute specifies the scheduled stop date/time to perform
                on-demand ETH-Test operations. The stop date/time value
                should be greater than or equal to the scheduled start
                date/time value.";
          }
        }

        case relative-time {
          choice start-time-type {
            case relative-start-time {
              leaf relative-start-time {
                type uint32;
                units "minutes";
                mandatory true;

                description
                 "This attribute specifies the relative start time, from the
current
                    system time, to perform on-demand ETH-Test. The
```


Copyright© 2022 Ciena® Corporation

```
                    default value for this attribute is zero, indicating
                    the current system time, which represents an
                    immediate start time.";
                }
              }
              case absolute-start-time {
                leaf absolute-start-time {
                  type yang:date-and-time;

                  description
                   "This attribute specifies the scheduled start date/time to
                   perform the on-demand ETH-Test operations. The default value
                     for this attribute is the current system date and time
                     which represents an immediate time.";
                }
              }
            }

            leaf duration {
              type uint32;
              units "minutes";
              mandatory true;

              description
                "This attribute specifies the duration of the Locking. The
duration time
                 can be specified as forever (represented by a zero value)
                 or as relative time (e.g., a given number of hours,
                 minutes, and seconds from the start time). If the
                 duration time is relative time, then the duration time
                 should be equal to or greater than the frame transmission
                 period.";
            }
          }
        }
      }

      leaf drop-eligible {
        type boolean;
        default false;

        must "../is-generator = true" {
          error-message "The drop-eligible parameter is only relevant for the
generator side";
        }

        description
          "This attribute specifies the eligibility of frames with ETH-Test
           information to be discarded when congestion conditions are
           encountered. The value 'true' indicates frames are eligible to
           be discarded. The value 'false' indicates frames are not
           eligible to be discarded. This attribute may be constrained to
           read-only in some implementations.";

        reference "[Y.1731] 7.7";
      }

      leaf frame-size {
        type uint32 {
          range "64..9600";
```


Copyright© 2022 Ciena® Corporation

```
      }
      units bytes;
      default 64;

      must "../is-generator = true" {
        error-message "The frame-size parameter is only relevant for the
generator side";
      }

      description
        "This attribute specifies the ETH-Test Ethernet frame size between 64
         bytes and the maximum transmission unit of the EVC. The
         adjustment to the frame size of the standard Test PDU size is
         accomplished by the addition of a Test TLV.";

      reference "[MEF7.1] 9.3.4.3";
    }

    leaf test-pattern {
      type test-pattern-type;

      description
        "This attribute specifies an arbitrary element to include in the Data
         TLV, if the Data TLV is selected to be sent, whose length and
         contents are configurable at the MEP. The contents can be a
         test pattern and an optional checksum.";
    }
    reference "[Y.1731] Figure 9.3-4";
  }

  grouping test-signal-stats-in-group {
    description
      "Data definitions related to TST frames received";

    leaf number-test-in {
      type yang:zero-based-counter64;

      description
        "This attribute contains the count of the total number of TST frames
         received. The count is incremented when a message is received
         with or without errors. This attribute is only applicable to the
         MEP receiving ETH TST frames.";
      reference
        "[MEF7.1] 9.3.4.4";
    }

    leaf number-test-in-out-of-order {
      type yang:zero-based-counter64;

      description
        "This attribute contains the count of the total number of valid,
         out-of-order TST frameTST frames received. The count is incremented
when
         the sequence number in the TST frame received does not match
         the expected sequence number. This attribute is only applicable
         to the MEP receiving ETH-TST frames.";
      reference
        "[MEF7.1] 9.3.4.4";
    }
```


Copyright© 2022 Ciena® Corporation

```
    leaf number-test-in-crc-errors {
      type yang:zero-based-counter64;

      description
        "This attribute contains the count of the total number of TST frames
         received with CRC errors. This attribute is only applicable to
         the MEP receiving ETH-Test messages that include the test TLV with
         test pattern of null CRC 32 or PBRS CRC 32.

        The CRC is dependent upon the Test TLV only and is independent of BER
         errors, which is used to indicate a pattern error.";
      reference
        "[MEF7.1] 9.3.4.4";
    }

    leaf number-test-in-ber-errors {
      type yang:zero-based-counter64;

      description
        "This attribute contains the count of the total number of TST frames
        received with BER or data errors. The count is incremented when the bit
         pattern in the received TST frame does not match the expected bit
         pattern. This attribute is only applicable to the MEP receiving ETH-
Test
         messages.

         The BER error count is independent of the CRC error count and is used
to
          indicate a data pattern error, while the CRC error is used to
indicate
          a TLV CRC error.";
      reference
        "[MEF7.1] 9.3.4.4";
    }
  }

  grouping test-signal-stats-out-group {
    description
      "Data definitions related to TST frames sent";

    leaf number-test-out {
      type yang:zero-based-counter64;

      description
        "This attribute contains the count of the total number of TST frames
         transmitted. This attribute is only applicable to the MEP sending
         ETH-TST frames. (i.e. The MEP under Test)";
      reference
        "[MEF7.1] 9.3.4.4";
    }
  }

  //
  // Augments into CFM related to MEF SOAM FM
  //
  augment "/cfm:maintenance-domain/cfm:maintenance-association" {
    description
      "This set of data definitions represents the Maintenance Entity Group
(Y.1731)
       configuration that is unique from the Maintenance Association.";
```


Copyright© 2022 Ciena® Corporation

```
    leaf y1731-compliant {
      type boolean;
      description
      "A boolean flag to indicate whether the MEG ID/MAID for this MEG operates
         in conformance with 802.1q (if false) or Y.1731 (if true).
         When set to false:
          - The format of the MAID (Maintenance Association ID) is controlled
            by the name-type and name in the maintenance-association instance,
              and the name-type and name in the maintenance-domain instance.
          - The meg-id-format and meg-level leafs are ignored.
          - The level is controlled by the md-level leaf.
         When set to true:
          - The MEG shall be in a maintenance-domain where name-type has the
value
              none(1).
          - The format of the MEG ID is as defined by meg-id-format.
          - The name-type in the maintenance-association is ignored.
          - The name leaf contains the MEG ID value unless the meg-id-format
            is set to 'icc-based' in which case it contains the value of the
            'icc-name-value' leaf
          - The md-level leaf is ignored, and the level is controlled
            by the meg-level leaf.";
      reference
        "[MEF31], MEF-SOAM-FM-MIB.mefSoamNetCfgY1731Compliant";
    }

    leaf meg-id-format {
      type enumeration {
        enum primary-vid {
          value 1;
          description
            "Primary VLAN ID. 12 bits represented in a 2-octet integer:
              - 4 least significant bits of the first byte contains the
                4 most significant bits of the 12 bits primary VID
              - second byte contains the 8 least significant bits of the
                primary VID";
        }
        enum char-string {
          value 2;
          description
            "RFC2579 DisplayString, except that the character codes 0-31
(decimal)
             are not used. (1..45) octets";
        }
        enum unsigned-int16 {
          value 3;
          description
            "2-octet integer/big endian";
        }
        enum rfc2865-vpn-id {
          value 4;
          description
            "RFC 2685 VPN ID 3 octet VPN authority Organizationally Unique
Identifier
             followed by 4 octet VPN index identifying VPN according to
             the OUI";
        }
        enum icc-based {
          value 32;
```

```
          description
            "ICC-based MEG ID Format, thirteen octet field";
        }
      }

      must "../y1731-compliant = true";

      description
        "This enumeration indicates the format of the MEG ID that is
         sent in the OAM PDUs. The primary-vid, char-string,
         unsigned-int16, and rfc2865-vpn-id types are more fully
         explained in [802.1q] 17.5. The icc-based type is from [Y.1731]
         Annex A.";
      reference
        "[Y.1731] Table A-1, [802.1q] 17.5, 21.6.5.1";
    }

    leaf icc-name-value {
      type string {
        length 13;
      }

      must "../meg-id-format = icc-based";

      description
        "ICC-based MEG ID Format, thirteen octet field. It consists of two
         subfields: the ITU Carrier Code (ICC) followed by a unique MEG
         ID code (UMC). The ITU Carrier Code consists of 1-6
         left-justified characters, alphabetic, or leading alphabetic
         with trailing numeric. The UMC code immediately follows the ICC
         and shall consist of 7-12 characters, with trailing NULLs,
         completing the 13-character MEG ID Value.";
      reference
        "[Y.1731] Annex A";
    }

    leaf meg-level {
      type cfm:md-level-type;

      must "../y1731-compliant = true";

      description
        "This attribute indicates the MEG Level of the MEG";

      reference
        "[MEF31] MEF-SOAM-FM-MIB.mefSoamNetCfgMegLevel";
    }
  }

  augment "/cfm:maintenance-domain/cfm:maintenance-association/
cfm:component-list" {
    description
      "This set of data definitions extends the component list of the
Maintenance
       Entity Group (as defined in Y.1731) or Maintenance Association (as
defined
       in 802.1q)";
    reference
      "[MEF7.1] 9.2.2";
```


Copyright© 2022 Ciena® Corporation

```
    leaf connectivity-status-interval {
      type yang:gauge32 {
        range "1..2100000";
      }
      units "milliseconds";

      description
        " This attribute specifies a configurable time interval to
detect a
         change in Connectivity Status. This is the timer timeout value that
         is used by the Remote Maintenance Endpoint (MEP)
state machine.


         This attribute overrides the standard CCM
loss of connectivity
         time interval which is 3.5 times the CCM interval.";
      reference
        "[MEF17] R2c, [802.1q] 20.1";
    }

    leaf peer-mep-info-aging-time {
      type uint32 {
        range "0..86400";
      }
      default 0;
      units "seconds";

      description
       "This attribute defines a period of time after which an instance in the
         remote-mep-database is removed if a valid CCM has not been
         received by the local MEP, i.e. remote-mep-state is set to
         'failed' for the period of time indicated by this parameter.

        A value of zero indicates no aging will occur.";
      reference
        "[MEF17] R5a and R5b, [802.1q] 20.19, 20.20";
    }
    leaf mep-port-status-tlv-included {
      type boolean;
      default true;

      description
        "Indicates whether a Port Status TLV is included in CCM frame
         transmission. A value of 'true' indicates that the Port Status
         TLV is to be included. A value of 'false' indicates that the
         Port Status TLV is not to be included.";
      reference
        "[MEF7.1] 9.2.2";
    }

    leaf mep-interface-status-tlv-included {
      type boolean;
      default true;

      description
        "Indicates whether a Interface Status TLV is included in CCM frame
       transmission. A value of 'true' indicates that the Interface Status TLV
         is to be included. A value of 'false' indicates that the Interface
```


Copyright© 2022 Ciena® Corporation

```
          Status TLV is not to be included.";
        reference
          "[MEF7.1] 9.2.2";
      }
    }

  augment "/cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point" {
      description
        "This set of data definitions extends the MEG End Point (as described in
         Y.1731) or Maintenance association End Point (as described in 802.1q)
         defined in the MEF CFM module (MEF-CFM)";
      reference
        "[MEF7.1] 9.2.2";

      leaf operational-state {
        type operational-state-type;

        config false;

        description
         "This attribute indicates the operational state (current capability) of
            the MEP.";
        reference
          "[MEF7.1] 9.2.5";
      }

      leaf connectivity-status {
        type enumeration {
          enum active {
          description
           "Refers to the ability to exchange SOAM PDU frames among all the UNIs
              of an EVC.";
          }
          enum partially-active {
          description
           "Refers to the ability to exchange SOAM PDU frames among some but not
              all the UNIs of an EVC.";
          }
          enum inactive {
          description
           "Refers to the ability to exchange SOAM PDU frames among any of the
UNIs
              of an EVC.";
          }
        }
        config false;

        description
          "This attribute indicates the connectivity status for a MEP in an EVC
           MEs. An 'active' MEP Connectivity Status refers to the ability
           to exchange SOAM PDU frames among all the UNIs of an EVC. A
           'partially-active' MEP Connectivity Status refers to the
           ability to exchange SOAM PDU frames among some but not all the
           UNIs of an EVC. An 'inactive' MEP Connectivity Status refers to
           the inability to exchange SOAM PDU frames among any of the UNIs
           of an EVC.";
        reference
          "[MEF7.1] 9.2.5";
      }
```


Copyright© 2022 Ciena® Corporation

```
    leaf port-status {
      type cfm:port-status-type;
      config false;

      description
       "An enumerated value of the Port status TLV sent in the last CCM from the
          local MEP or the default value psNoPortStateTLV indicating no
          CCM has been sent or no Port Status TLV has been sent.";
      reference
        "[802.1q] 17.5";
    }

    leaf interface-status {
      type cfm:interface-status-type;
      config false;

      description
        "An enumerated value of the Interface status TLV sent in the last CCM
         from the local MEP or the default value no-status-tlv
         indicating no CCM has been sent or no Interface Status TLV has
         been sent.";
      reference
        "[802.1q] 17.5";
    }

    leaf last-defect-sent {
      type cfm:fault-alarm-defect-bits-type;
      config false;

      description
        "This attribute indicates the state of the previous MEP defects,
         that was sent with the previous
         mep-defect-alarm notification. It is always some *previous*
         value of cfm:active-defects. Once an mep-defect-alarm is sent
         the mep-defect-alarm that was sent in the notification
         updates the contents of this leaf.

         If no mep-defect-alarm notification has been sent the value of
         this leaf is '0'.";
      reference
        "[MEF31] MEF-SOAM-FM-MIB.mefSoamMepStatusLastDefectSentStatus";
    }

    leaf rdi-transmit-status {
      type boolean;
      config false;

      description
       "Indicates whether the local MEP is generating a RDI bit in the CCM that
          it transmits. A value of 'true' indicates that a RDI bit was
          set in the last CCM that the MEP transmitted. A value of
          'false' indicates that the last CCM transmitted by the MEP did
          not set the RDI bit or that a CCM has never been transmitted by
          the MEP.";
      reference
        "[MEF7.1] 9.2.2";
    }

    leaf oam-frames-discarded {
```


Copyright© 2022 Ciena® Corporation

```
      type yang:counter32;
      config false;

      description
       "This attribute indicates the count of incoming OAM frames discarded at
          the MEP. This count includes frames discarded because they have
          an unknown OpCode, and frames (other than CCMs) discarded
          because they have a level below the level of the MEP.  In other
          words, this attribute counts frames discarded by the MEP Equal
          OpCode Demultiplexer and the MEP Low OpCode Demultiplexer
          described in IEEE 802.1q Sn 19.2.7, Table 19-1 and Figure
          19-2.  This count does not include frames that are malformed,
          or that contain OpCode-specific errors (such as CCM defects or
          LBRs with bad data).";
      reference
        "[802.1q] 19.2.7, [MEF31]
        MEF-SOAM-FM-MIB.mefSoamMepFmStatsInOamFramesDiscarded";
    }
  }

  augment "/cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/cfm:continuity-check" {
    description
      "This set of data definitions extends the proactive Ethernet OAM Fault
       Management and Performance Monitoring Continuity Check function
       (ETH-CC) as described in Y.1731 and 802.1q and defined in
       MEF-CFM";
    reference
      "[MEF7.1] 9.3.1";

    leaf drop-eligible {
      type boolean;
      default false;

      description
        "This attribute specifies the eligibility of frames with ETH-CC and
         ETH-RDI information to be discarded when congestion conditions
         are encountered.

         The value 'true' indicates frames are eligible to be discarded.

         The value 'false' indicates frames are not eligible to be
         discarded.";
      reference
        "[MEF7.1] 9.3.1.1";
    }

    leaf total-ccm-in {
      type yang:counter32;
      config false;

      description
        "This attribute indicates the count of the total number of valid (not
         malformed) CCMs received by the MEP.  In other words, it counts
         the frames received by the MEP Continuity Check Receiver
         described in IEEE 802.1q Sn 19.2.8 and Figure 19-2.  This
         includes CCMs at a lower level, CCMs with defects, CCMs from an
         unexpected peer MEP and out-of-sequence CCMs.  It does not
         include CCMs at a higher level than the MEP.";
      reference
```


Copyright© 2022 Ciena® Corporation

```
        "[802.1q] 19.2.8, [MEF31] MEF-SOAM-FM-
MIB.mefSoamMepFmStatsInCcmTotal";
    }
  }

  augment "/cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/cfm:loopback" {
    description
      "This set of data definitions extends on-demand Ethernet OAM Fault
       Management Loopback function (ETH-LB) as defined in Y.1731 and
       802.1q and defined in MEF-CFM";
    reference
      "[MEF7.1] 9.3.2";

    leaf multicast-enabled {
      type boolean;
      default false;

      description
        "This attribute specifies whether a MEP uses unicast or multicast
         to send the ETH-LB messages (LBM). The 802.1ag standard
only allows
         unicast LBM. ITU-T Y.1731 allows LBM to be multicast.
This attribute
         allows the MEP to send either multicast or unicast
LBM on a per MEP
         basis.

         The value 'true' indicates multicast is enabled.

         The value 'false' indicates unicast is enabled.";
      reference
        "[MEF7.1] 9.3.2.1";
    }

    leaf interval {
      type uint32 {
        range "0..60000";
      }
      default 1000;
      units milliseconds;

      description
       "This attribute specifies the period between LBM transmissions in an LB
          Session. For an LB Session, the period for LBM transmission
         is configurable in the range 0 and sixty seconds (60 s). Granularity
         of 100ms is required.


         The transmission of the next LBM is not dependent upon the
         reception the first LBR. The next LBM is sent out based upon
         the interval count.

         An interval count of '0' indicates that the subsequent LBM is sent
         out with the minimum possible delay.";
      reference
        "[MEF7.1] 9.3.2.1";
    }

    leaf frame-size {
```


Copyright© 2022 Ciena® Corporation

```
      type uint32 {
        range "64..9600";
      }
      units "bytes";

      must "not(../data-tlv)" {
        error-message "Specific frame-size not allowed, frame size is based
on data-tlv";
      }

      description
      "This attribute specifies the LBM frame size. For an LB Session, the size
         of the LBM frame is configurable to any Ethernet frame size
         between 64 Bytes and the maximum transmission unit of the
         EVC.

          The range of frame sizes from 64 through 2000 octets, in 4 octet
increments,
          MUST be supported, and the range of frame sizes from 2004
through 9600
          octets, in 4 octet increments, SHOULD be supported.

          The adjustment to the frame size of the standard LBM PDU size is
accomplished
          by the addition of a Data TLV or a Test TLV.";
      reference
        "[MEF7.1] 9.3.2.1";
    }

    leaf data-pattern {
      type enumeration {
        enum zeroes {
            description "Indicates the Data TLV contains all zeros.";
                }
        enum ones {
            description "Indicates the Data TLV contains all ones.";
                }
      }
      default zeroes;
      must "not(../data-tlv)" {
       error-message "Specific data-pattern not allowed, pattern is based on
data-tlv";
      }

      description
        "This attribute specifies the LBM data pattern included in a Data TLV
         when the size of the LBM frame is determined by the frame-size
         leaf and test-tlv-included is 'false'.";
      reference
        "[MEF31] MEF-SOAM-FM-MIB.mefSoamLbCfgDataPattern";
    }

    leaf test-tlv-included {
      type boolean;
      default false;
      must "not(../data-tlv)" {
       error-message "Test-tlv-included does not apply when data-tlv exists";
      }

      description
```


Copyright© 2022 Ciena® Corporation

```
        "Indicates whether a Test TLV or Data TLV is included when the size of
          the LBM frame is determined by the frame-size leaf.

          A value of 'true' indicates that the Test TLV is to be
          included.

          A value of 'false' indicates that the Data TLV is to be
          included.";
        reference
          "[Y.1731] 9.3, [MEF31] MEF-SOAM-FM-MIB.mefSoamLbCfgDataPattern";
      }

      leaf test-tlv-pattern {
        type test-pattern-type;
        default "null-signal-without-crc-32";

        must "not(../data-tlv)" {
          error-message "Test-tlv-pattern does not apply when data-tlv exists";
        }

        description
         "This attribute specifies the type of test pattern to be sent in the LBM
            frame Test TLV when the size of LBM PDU is determined by the
            frame-size leaf and test-tlv-included is 'true'.";
        reference
          "[MEF7.1] 9.3.2.1, [MEF31] MEF-SOAM-FM-
MIB.mefSoamLbCfgTestTlvPattern";
      }

      leaf crc-errors-in {
        type yang:counter32;
        config false;

        description
          "This attribute contains the count of the total number of LBR messages
            received with CRC errors. This is only applicable when the
            ETH-LB includes the test TLV with a test pattern of null-crc-32
            or pbrs-crc-32.";
        reference
          "[MEF7.1] 9.3.2.2";
      }

      leaf timeout {
        type yang:gauge32 {
          range "1..10000";
        }
        default 5000;

        description
        "This attribute specifies the maximum amount of time to receive an LBR in
            response to a LBM. If a LBR is not received within the timeout
            value it is considered lost.";
        reference
          "[MEF31] MEF-SOAM-FM-MIB.mefSoamLbCfgTimeout";
      }

      list responses {
        key "transaction-id receive-order";
        unique "transaction-id receive-order";
        config false;
```


Copyright© 2022 Ciena® Corporation

```
      description
        "A list of responses from a Multicast Loopback Message";
      reference
        "[MEF31] MEF-SOAM-FM-MIB.mefSoamLbrMulticastTable";

      leaf transaction-id {
        type cfm:lbm-transaction-id-type;

        description
          "Loopback transaction identifier returned by a previous loopback
message
           command, indicating which loopback request is returned.";
        reference
          "[MEF31] MEF-SOAM-FM-MIB.mefSoamLbrMulticastTransId";
      }
      leaf receive-order {
        type uint32 {
          range "0..2147483647";
        }

        description
          "An index to distinguish among multiple LBRs with the same LBR
           Transaction Identifier field value. Value is assigned
           sequentially from 1, in the order that the Loopback Initiator
           received the LBR";
        reference
          "[MEF31] MEF-SOAM-FM-MIB.mefSoamLbrMulticastReceiveOrder";
      }

      leaf multicast-reply-mac {
        type yang:mac-address;

        description
          "Source MAC address returned in the LBR Ethernet frame";
        reference
          "[802.1q] 21.7, [Y.1731] 7.2,
           [MEF31] MEF-SOAM-FM-MIB.mefSoamLbrMulticastReplyMac";
      }
    }
  }

  augment "/cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/cfm:linktrace" {
    description
      "Augments to support the enhanced CFM Linktrace functionality";
    reference
      "[MEF30] 8.4";

    leaf ltm-msgs-transmitted {
      type yang:zero-based-counter32;
      config false;

      description
       "This attribute contains the count of the total number of LTM messages
          transmitted by the MEP";
      reference
        "[MEF31] MEF-SOAM-FM-MIB.mefSoamLtLtmTransmitted";
    }
```


Copyright© 2022 Ciena® Corporation

```
    leaf ltr-msgs-received {
      type yang:zero-based-counter32;
      config false;

      description
       "This attribute contains the count of the total number of LTR messages
          received by the MEP";
      reference
        "[MEF31] MEF-SOAM-FM-MIB.mefSoamLtLtrReceived";
    }

    leaf ltm-msgs-received {
      type yang:zero-based-counter32;
      config false;

      description
       "This attribute contains the count of the total number of LTM messages
          received by the MEP";
      reference
        "[MEF31] MEF-SOAM-FM-MIB.mefSoamLtLtmReceived";
    }

    leaf ltr-msgs-transmitted {
      type yang:zero-based-counter32;
      config false;

      description
       "This attribute contains the count of the total number of LTR messages
          transmitted by the MEP";
      reference
        "[MEF31] MEF-SOAM-FM-MIB.mefSoamLtLtrTransmitted";
    }
  }

  augment "/cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point" {
    description
      "Augments to support the AIS functionality";
    reference
      "[Y.1731] 7.4, [MEF30] 8.5";

    container alarm-indication-signal {
      description
       "This set of data definitions covers the proactive Ethernet OAM Fault
          Management Alarm Indication Signal function (ETH-AIS) as
          defined in Y.1731 and extended in MEF30";
      reference
        "[Y.1731] 7.4, [MEF30] 8.5";

      leaf enabled {
        type boolean;
        default false;

        description
        "This attribute specifies whether ETH-AIS transmission is enabled. The
           value 'true' indicates ETH-AIS transmission is enabled. The
           value 'false' indicates ETH-AIS transmission is disabled.";
        reference
          "[MEF7.1] 9.3.4.1, [MEF30] 8.4";
      }
```


Copyright© 2022 Ciena® Corporation

```
      leaf interval {
        type interval-type-ais-lck-type;
        default one-second;

        description
          "This attribute specifies the ETH-AIS transmission period.";
        reference
          "[MEF7.1] 9.3.4.1";
      }

      leaf priority {
        type cfm:priority-type;

        description
          "This attribute specifies the priority of frames with ETH-AIS
           information. If ETH-AIS is supported, the default value for
           the CoS for AIS frame MUST be the value which yields the
           lowest frame loss performance for this EVC.";
        reference
          "[MEF7.1] 9.3.4.1";
      }

      leaf client-meg-level {
        type cfm:md-level-type;
        default 0;

        description
          "The MEG/Maintenance Domain Level of the Client ETH-AIS PDU
(transmitted
           level).";
        reference
          "[Y.1731] 7.4";
      }

      leaf drop-eligible {
        type boolean;
        default false;

        description
          "This attribute specifies the eligibility of frames with ETH-AIS
           information to be discarded when congestion conditions are
           encountered.

           The value 'true' indicates frames are eligible to be
           discarded.

           The value 'false' indicates frames are not eligible to be
           discarded. This attribute may be constrained to read-only in
           some implementations";
        reference
          "[MEF7.1] 9.3.4.1";
      }

      leaf transmit-status {
        type boolean;
        config false;

        description
         "This attribute specifies the current AIS transmission status of the
```


Copyright© 2022 Ciena® Corporation

```
          MEP.

          The value 'true' indicates AIS frames are currently being
          transmitted by the MEP.

          The value 'false' indicates AIS frames are not currently
          being transmitted by the MEP.";
        reference
          "[Y.1731] 7.4";
      }

      leaf receive-status {
        type boolean;
        config false;

        description
          "This attribute specifies the current AIS receive status of the
          MEP. The value 'true' indicates an AIS PDU has been received
          and 3.5 times the interval defined in the PDU has not yet
          passed, otherwise it is 'false'.";
        reference
          "[Y.1731] 7.4";
      }

      leaf transmit-counter {
        type yang:zero-based-counter32;
        config false;

        description
        "This attribute contains the count of the total number of AIS messages
           sent by the MEP. The count is incremented every time an AIS
           PDU is transmitted by the MEP.";
        reference
          "[MEF7.1] 9.3.2.2";
      }

      leaf receive-counter {
        type yang:counter32;
        config false;

        description
        "This attribute contains the count of the total number of AIS messages
           received by the MEP. The count is incremented every time an
           AIS PDU is received by the MEP.";
        reference
          "[MEF7.1] 9.3.2.2";
      }

      leaf last-received-mac-address {
        type yang:mac-address;
        config false;

        description
          "The source MAC Address Field of last AIS received by the MEP.
          If no AIS PDU has been received by the NE the MAC address is
          set to all zeros.";
        reference
          "[Y.1731] 7.7";
      }
    }
```


Copyright© 2022 Ciena® Corporation

```
  }

  augment "/cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point" {
    description
      "Augments to support the Locked Signal functionality";
    reference
      "[Y.1731] 8.6, [MEF30] 8.4";

    container locked-signal {
      description
        "This set of data definitions covers the on-demand Ethernet OAM Fault
         Management Locked Signal function (ETH-LCK) as defined in
         Y.1731.";
      reference
        "[Y.1731] 8.6, [MEF7.1] 9.3.4.2";

      uses locked-signal-parameters-group;

      leaf receive-status {
        type boolean;
        config false;

        description
         "This attribute specifies the current LCK receive status of the MEP.

           The value 'true' indicates LCK frames are currently being
           received by the MEP.

           The value 'false' indicates LCK frames are not currently
           being received by the MEP at the specified interval in the
           LCK PDU.

           If no LCK frames are received within an interval of 3.5 times
           the LCK transmission period indicated in the last LCK frame
           received, the MEP clears the LCK condition by setting
           mefSoamLckInStatus to 'false'.";
        reference
          "[Y.1731] 7.6";
      }

      leaf messages-received {
        type yang:zero-based-counter32;
        config false;

        description
        "This attribute contains the count of the total number of LCK messages
           received. The count is incremented when a ETH-LCK message is
           received.  This attribute is only applicable to the MEP
           receiving ETH-LCK messages.

           The initial value of this leaf is
           zero.";
        reference
          "[Y.1731] 7.6";
      }

      leaf transmit-status {
        type boolean;
        config false;
```


Copyright© 2022 Ciena® Corporation

```
        description
         "This attribute specifies the current LCK transmission status of the
MEP.

            The value 'true' indicates LCK frames are currently being
            transmitted by the MEP.

            The value 'false' indicates LCK frames are not currently being
            transmitted by the MEP.";
        reference
          "[Y.1731] 7.6";
      }

      leaf messages-transmitted {
        type yang:zero-based-counter32;
        config false;

        description
        "This attribute contains the count of the total number of LCK messages
           transmitted. This attribute is only applicable to the MEP
           sending ETH-LCK messages.

           The initial value of this leaf is zero.";
        reference
          "[Y.1731] 7.6";
      }
    }
  }

  augment "/cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point" {
    description
      "Augments to support the Test Signal functionality";
    reference
      "[Y.1731] 7.7, [MEF30] 8.7";

    container test-signal {
      description
        "This container augments cfm:maintenance-association-end-point with
data
       definitions related to the on-demand OAM Fault Management Test function
         (ETH-Test) defined in [Y.1731].";
      reference
        "[Y.1731] 7.7, [MEF7.1] 9.3.4.3";

      leaf transmit-enabled {
        type boolean;
        default false;

        description
          " This attribute specifies the enabling of the ETH-Test
           transmit function. A value of 'true' indicates that the
           ETH-Test transmit function is enabled. A value of of
           'false' indicates that ETH-Test transmit function is
           disabled.";
        reference
          "[Y.1731] 7.7";
      }
```


Copyright© 2022 Ciena® Corporation

```
      uses test-signal-parameters-group;
      container test-signal-stats-in {
        description
          "Data definitions related to received Test Signal PDUs statistics";

        config false;
        must "../is-receiver = true";
        uses test-signal-stats-in-group;
      }
      container test-signal-stats-out {
        description
        "Data definitions related to transmitted Test Signal PDUs statistics";

        config false;
        must "../is-generator = true";
        uses test-signal-stats-out-group;
      }
    }
  }

  //
  // Data definitions related to MEF SOAM FM
  //

  container notification-configuration {
    description
      "A set of data definitions related to FM event notifications.";
    reference
      "[MEF31] MEF-SOAM-FM-MIB.mefSoamFmNotificationCfg";

    leaf alarm-interval {
      type uint32 {
        range "0..60";
      }
      default 5;
      units seconds;

      description
        " A value indicating the shortest time interval in seconds between
          the generation of the same notification type per MEP to the list of
          notification destinations. A server shall generate the first
        notification of given type for a given MEP immediately. A server shall
          not generate a second specific notification of the same type for the
          same MEP until the time interval has expired. A value of zero
          indicates that all notifications are sent immediately upon detection
          of the condition.";
      reference
        "[MEF31] MEF-SOAM-FM-MIB.mefSoamAlarmInterval";
    }
  }

  //
  // RPCs related to MEF SOAM FM
  //

  rpc create-test-signal {
    description
      "Start a test-signal session on a specific MEP according to the
       content of the test signal parameters.
```


Copyright© 2022 Ciena® Corporation

```
      The content of the test-signal container in the associated MEP
      will be updated throughout the session upon successful creation.";

    input {
      leaf mep-id {
        type cfm:mep-id-type;

        description
          "The ID of a MEP on which a test session will be started";
      }
      uses test-signal-parameters-group;
    }
  }

  rpc abort-test-signal {
    description
      "Administratively abort a test-signal session on a specific MEP";

    input {
      leaf mep-id {
        type cfm:mep-id-type;

        description
          "The ID of the MEP on which the ETH Test session should
           be aborted";
      }
    }
  }

  //
  // Notifications related to MEF SOAM FM
  //

  notification mep-defect-alarm {
    description
      "A mep-defect-alarm notification is generated when the value of
       mep-defects changes. It indicates a persistent defect in
       the MEP. This notification is sent whenever the
       cfm:active-defects of the MEP changes, regardless of the
       cfm:highest-priority-defect-found leaf.

       The inclusion of the cfm:remote-mep-state leaf is
       optional. It shall not be included if the defect is not based
       upon a specific MEP instance, e.g.. bDefErrorCCM.";

    reference
      "[MEF31] MEF-SOAM-FM-MIB.mefSoamMepDefectAlarm";

    leaf mep-id {
      type cfm:mep-id-type;

      description
        "A reference to the specific MEP on which the persistent defect is
         present";
    }

    leaf last-defect-sent {
      type cfm:fault-alarm-defect-bits-type;

      description
```


Copyright© 2022 Ciena® Corporation

```
        "The last defect sent on the specific MEP";
    }
    leaf active-defects {
      type cfm:fault-alarm-defect-bits-type;
      description
        "The currently active defects on the specific MEP.";
    }

    leaf remote-mep-state {
      type cfm:remote-mep-state-type;

      description
        "The value of the remote MEP state on a specific MEP";
      reference
        "[802.1q] 12.14.7.6.3:b";
    }
  }

  notification configuration-error-assert {
    description
     "A configuration-error-assert notification is generated when an entry is
     added to the configuration-error-list. The vlan-identifier and interface
      pair uniquely identifies the configuration-error-list entry that was
      added.";
    reference
      "[MEF31] MEF-SOAM-FM-MIB.mefSoamConfigErrorAssertAlarm";

    leaf vlan-identifier {
      type leafref {
        path "/cfm:configuration-error-list/cfm:vlan-identifier";
      }
      description
        "Reference to a specific vlan-identifier in the configuration-error-
list";
    }
    leaf interface {
      type leafref {
        path "/cfm:configuration-error-list/cfm:interface";
      }
      description
        "Reference to a specific interface in the configuration-error-list";
    }
  }

  notification configuration-error-clear {
    description
      "A configuration-error-clear notification is generated when an entry is
       removed to the configuration-error-list. The vlan-identifier and
interface
       pair uniquely identifies the configuration-error-list entry that was
       removed.";
    reference
      "[MEF31] MEF-SOAM-FM-MIB.mefSoamConfigErrorClearAlarm";

    leaf vlan-identifier {
      type leafref {
        path "/cfm:configuration-error-list/cfm:vlan-identifier";
      }
      description
```


Copyright© 2022 Ciena® Corporation

```
        "Reference to a specific vlan-identifier in the configuration-error-
list";
    }
    leaf interface {
      type leafref {
        path "/cfm:configuration-error-list/cfm:interface";
      }
      description
        "Reference to a specific interface in the configuration-error-list";
    }
  }

  notification mep-operational-state-changed {
    description
     "A mep-operational-state-changed notification is sent when the value of a
       MEP's operational-state changes. It indicates an operational
       state change in the MEP. This notification is sent whenever the
       operational status of the MEP changes.";
    reference
      "[MEF31] MEF-SOAM-FM-MIB.mefSoamMepOperStatusAlarm";

    leaf mep-id {
      type cfm:mep-id-type;

      description
        "The ID of the MEP on which the operational-state changed";
    }

    leaf operational-state {
      type operational-state-type;

      description
        "The operational-state leaf of the affected MEP";
    }
  }

  notification lock-alarm {
    description
      "A lock-alarm notification is sent when either receive-status or
       transmit-status changes. Reception of the LCK PDU causes the MEP
       to enter Lock State.  This notification is sent whenever the
       operational lock status of the MEP changes.";
    reference
      "[MEF31] MEF-SOAM-FM-MIB.mefSoamLckAlarm";

    leaf mep-id {
      type cfm:mep-id-type;

      description
        "The ID of the MEP on which the lock alarm is present.";
    }

    leaf receive-status {
      type boolean;

      description
        "The value of the locked signal receive-status of the affected MEP";
    }

    leaf transmit-status {
```


Copyright© 2022 Ciena® Corporation

```
      type boolean;

      description
        "The value of the locked signal transmit-status of the affected MEP";
    }
  }

  notification ais-alarm {
    description
      "An ais-alarm notification is sent when the state of either
       transmit-status or receive-status changes. Transmit-status is set
       to 'true' when AIS frames are sent by the MEP and set to 'false'
       when the MEP stops sending AIS frames. Receive-status is set to
       'true' when AIS PDUs are received and is set to 'false' when AIS
       PDUs stop being received.";
    reference
      "[MEF31] MEF-SOAM-FM-MIB.mefSoamAisAlarm";

    leaf mep-id {
      type cfm:mep-id-type;

      description
        "The ID of the MEP on which the AIS alarm is present.";
    }

    leaf receive-status {
      type boolean;

      description
        "The value of the AIS receive-status of the affected MEP";
    }

    leaf transmit-status {
      type boolean;

      description
        "The value of the AIS transmit-status of the affected MEP";
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

mef-soam-pm-deviation.yang

```
module mef-soam-pm-deviation {

  namespace "http://www.ciena.com/ns/yang/mef-soam-pm-dev";
  prefix "mef-soam-pm-dev";

  import ietf-yang-types {
    prefix yang;
    revision-date 2010-09-24;
  }
  import mef-cfm {
    prefix cfm;
  }
  import mef-soam-pm {
    prefix mef-soam-pm;
  }

  revision 2017-08-31 {
    description "Added deviations to mef-soam-pm.yang model.";
  }

  revision 2016-03-09 {
    description "Initial revision";
  }



  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-single-ended-
responder {
    deviate replace {
      config false;
    }
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:synthetic-loss-single-
ended-responder {
    deviate replace {
      config false;
    }
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-single-ended-
responder {
    deviate replace {
      config false;
    }
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:measurement-type {
    deviate replace {
      type enumeration {
        enum lmm {
          description
            "LMM SOAM PDU generated and received LMR responses tracked.";
```


Copyright© 2022 Ciena® Corporation

```
        }
        enum slm {
          description
            "SLM SOAM PDU generated and received SLR responses tracked.";
        }
      }
    }
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:version {

    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:enabled-counters {
    deviate replace {
      type bits {
        bit forward-transmitted-frames;
        bit forward-received-frames;
        bit forward-min-flr;
        bit forward-max-flr;
        bit forward-average-flr;
        bit backward-transmitted-frames;
        bit backward-received-frames;
        bit backward-min-flr;
        bit backward-max-flr;
        bit backward-average-flr;
        bit soam-pdus-sent;
        bit soam-pdus-received;
        bit measured-stats-forward-measured-flr;
        bit measured-stats-backward-measured-flr;
      }
    }
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:data-pattern {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:test-tlv-included {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:test-tlv-pattern {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:number-intervals-stored {
```


Copyright© 2022 Ciena® Corporation

```
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:align-measurement-intervals {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:align-measurement-offset {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:availability-measurement-interval {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:availability-number-consecutive-flr-
measurements {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:availability-flr-threshold {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:availability-number-consecutive-
intervals {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:availability-number-consecutive-high-
flr {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:session-type {
    deviate not-supported;
    description
     "Only 'on demand' is supported";

  }
```


Copyright© 2022 Ciena® Corporation

```
  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:thresholds/mef-soam-pm:measured-flr-
forward {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:thresholds/mef-soam-pm:measured-flr-
backward {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:thresholds/mef-soam-pm:forward-high-
loss {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:thresholds/mef-soam-pm:forward-
consecutive-high-loss {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:thresholds/mef-soam-pm:backward-high-
loss {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:thresholds/mef-soam-pm:backward-
consecutive-high-loss {
    deviate not-supported;
  }


  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:thresholds/mef-soam-pm:forward-
unavailable-count {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:thresholds/mef-soam-pm:forward-
available-ratio {
    deviate not-supported;
  }
```


Copyright© 2022 Ciena® Corporation

```
  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:thresholds/mef-soam-pm:backward-
unavailable-count {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:thresholds/mef-soam-pm:backward-
available-ratio {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:start-time {

    deviate not-supported;
    description
     "Only start-time of immediate is supported";
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:stop-time/mef-soam-pm:stop-time/mef-
soam-pm:absolute {

    deviate not-supported;
    description
     "Only relative time or 'none' is supported";

  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:measured-availability-forward-status {

    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:measured-availability-backward-status {

    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:measured-forward-last-transition-time {

    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:measured-backward-last-transition-time
{
```


Copyright© 2022 Ciena® Corporation

```
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:current-availability-stats {

    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:current-measurement-stats {

    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:history-availability-stats {

    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:history-measurement-stats/mef-soam-
pm:elapsed-time {

    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:loss-measurements/mef-
soam-pm:loss-measurement/mef-soam-pm:history-measurement-stats/mef-soam-
pm:suspect-status {

    deviate not-supported;
  }

  // Deviation for delay-measurement

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:measurement-type {

    deviate not-supported;
    description
     "Only type dmm is supported and hence no config is possible";
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:version {

    deviate not-supported;
  }
```


Copyright© 2022 Ciena® Corporation

```
  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:measurement-enable {
    deviate replace {
      type bits {
        bit soam-pdus-sent;
        bit soam-pdus-received;
        bit frame-delay-two-way-bins;
        bit frame-delay-two-way-min;
        bit frame-delay-two-way-max;
        bit frame-delay-two-way-average;
        bit inter-frame-delay-variation-two-way-bins;
        bit inter-frame-delay-variation-two-way-min;
        bit inter-frame-delay-variation-two-way-max;
        bit inter-frame-delay-variation-two-way-average;
        bit measured-stats-frame-delay-two-way;
        bit measured-stats-inter-frame-delay-variation-two-way;
     }
    }
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:data-pattern {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:test-tlv-included {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:test-tlv-pattern {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:number-intervals-stored {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:align-measurement-intervals {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:align-measurement-offset {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:session-type {
```


Copyright© 2022 Ciena® Corporation

```
    deviate not-supported;
    description
     "we only support 'on-demand' sessions";
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:ifdv-selection-offset {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:bins-per-fdr-interval {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:thresholds/mef-soam-pm:enabled-
thresholds {
    deviate replace {
      type bits {
        bit measured-frame-delay-two-way;
        bit max-frame-delay-two-way;
        bit measured-inter-frame-delay-variation-two-way;
        bit max-inter-frame-delay-variation-two-way;
      }
    }
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:thresholds/mef-soam-pm:average-frame-
delay-two-way {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:thresholds/mef-soam-pm:average-inter-
frame-delay-variation-two-way {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:thresholds/mef-soam-pm:max-frame-
delay-range-two-way {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:thresholds/mef-soam-pm:average-frame-
delay-range-two-way {
    deviate not-supported;
  }
```


Copyright© 2022 Ciena® Corporation

```
  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:thresholds/mef-soam-pm:measured-frame-
delay-forward {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:thresholds/mef-soam-pm:max-frame-
delay-forward {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:thresholds/mef-soam-pm:average-frame-
delay-forward {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:thresholds/mef-soam-pm:measured-inter-
frame-delay-variation-forward {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:thresholds/mef-soam-pm:max-inter-
frame-delay-variation-forward {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:thresholds/mef-soam-pm:average-inter-
frame-delay-variation-forward {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:thresholds/mef-soam-pm:max-frame-
delay-range-forward {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:thresholds/mef-soam-pm:average-frame-
delay-range-forward {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:thresholds/mef-soam-pm:measured-frame-
delay-backward {
```


Copyright© 2022 Ciena® Corporation

```
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:thresholds/mef-soam-pm:max-frame-
delay-backward {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:thresholds/mef-soam-pm:average-frame-
delay-backward {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:thresholds/mef-soam-pm:measured-inter-
frame-delay-variation-backward {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:thresholds/mef-soam-pm:max-inter-
frame-delay-variation-backward {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:thresholds/mef-soam-pm:average-inter-
frame-delay-variation-backward {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:thresholds/mef-soam-pm:max-frame-
delay-range-backward {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:thresholds/mef-soam-pm:average-frame-
delay-range-backward {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:stop-time/mef-soam-pm:stop-time/mef-
soam-pm:absolute {

    deviate not-supported;
    description
     "Only relative time is supported";
```


Copyright© 2022 Ciena® Corporation

```
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:source-mac-address {
    deviate replace {
      config false;
    }
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:frame-delay-two-way {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:frame-delay-forward {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:frame-delay-backward {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:inter-frame-delay-variation-two-way {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:inter-frame-delay-variation-forward {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:inter-frame-delay-variation-backward {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:current-stats {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:history-stats/mef-soam-pm:elapsed-time
{
    deviate not-supported;
  }
```


Copyright© 2022 Ciena® Corporation

```
  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:history-stats/mef-soam-pm:suspect-
status {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:history-stats/mef-soam-pm:frame-delay-
forward-min {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:history-stats/mef-soam-pm:frame-delay-
forward-max {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:history-stats/mef-soam-pm:frame-delay-
forward-average {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:history-stats/mef-soam-pm:frame-delay-
backward-min {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:history-stats/mef-soam-pm:frame-delay-
backward-max {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:history-stats/mef-soam-pm:frame-delay-
backward-average {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:history-stats/mef-soam-pm:inter-frame-
delay-variation-forward-min {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:history-stats/mef-soam-pm:inter-frame-
delay-variation-forward-max {
```


Copyright© 2022 Ciena® Corporation

```
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:history-stats/mef-soam-pm:inter-frame-
delay-variation-forward-average {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:history-stats/mef-soam-pm:inter-frame-
delay-variation-backward-min {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:history-stats/mef-soam-pm:inter-frame-
delay-variation-backward-max {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:history-stats/mef-soam-pm:inter-frame-
delay-variation-backward-average {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:history-stats/mef-soam-pm:frame-delay-
range-forward-max {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:history-stats/mef-soam-pm:frame-delay-
range-forward-average {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:history-stats/mef-soam-pm:frame-delay-
range-backward-max {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:history-stats/mef-soam-pm:frame-delay-
range-backward-average {
    deviate not-supported;
  }
```


Copyright© 2022 Ciena® Corporation

```
  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:history-stats/mef-soam-pm:frame-delay-
range-two-way-max {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:history-stats/mef-soam-pm:frame-delay-
range-two-way-average {
    deviate not-supported;
  }

  deviation /cfm:maintenance-domain/cfm:maintenance-association/
cfm:maintenance-association-end-point/mef-soam-pm:delay-measurements/mef-
soam-pm:delay-measurement/mef-soam-pm:history-stats/mef-soam-pm:bins/mef-
soam-pm:frame-delay-range {
    deviate not-supported;
  }

  // Deviations for rpcs related to frame-loss-measurement

  deviation /mef-soam-pm:create-loss-measurement/mef-soam-pm:input/mef-soam-
pm:version {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-loss-measurement/mef-soam-pm:input/mef-soam-
pm:measurement-type {
    deviate not-supported;
    description
      "Only type slm is supported and hence no config is possible";
  }

  deviation /mef-soam-pm:create-loss-measurement/mef-soam-pm:input/mef-soam-
pm:enabled-counters {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-loss-measurement/mef-soam-pm:input/mef-soam-
pm:data-pattern {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-loss-measurement/mef-soam-pm:input/mef-soam-
pm:test-tlv-included {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-loss-measurement/mef-soam-pm:input/mef-soam-
pm:test-tlv-pattern {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-loss-measurement/mef-soam-pm:input/mef-soam-
pm:measurement-interval {
    deviate replace {
      type uint32 {
        range "1..43200";
```


Copyright© 2022 Ciena® Corporation

```
      }
    }
  }

  deviation /mef-soam-pm:create-loss-measurement/mef-soam-pm:input/mef-soam-
pm:number-intervals-stored {
    deviate replace {
      type uint32 {
        range "2..96";
      }
    }
  }

  deviation /mef-soam-pm:create-loss-measurement/mef-soam-pm:input/mef-soam-
pm:align-measurement-intervals {
    deviate replace {
      default false;
    }
  }

  deviation /mef-soam-pm:create-loss-measurement/mef-soam-pm:input/mef-soam-
pm:align-measurement-offset {
    deviate replace {
      type uint32 {
         range "0..1440";
      }
    }
  }

  deviation /mef-soam-pm:create-loss-measurement/mef-soam-pm:input/mef-soam-
pm:availability-number-consecutive-flr-measurements {
    deviate replace {
      type uint32 {
        range "10..1000000";
      }
    }
  }

  deviation /mef-soam-pm:create-loss-measurement/mef-soam-pm:input/mef-soam-
pm:availability-measurement-interval {
    deviate replace {
      type uint32 {
        range "1..43200";
      }
    }
  }

  deviation /mef-soam-pm:create-loss-measurement/mef-soam-pm:input/mef-soam-
pm:thresholds/mef-soam-pm:measured-flr-forward {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-loss-measurement/mef-soam-pm:input/mef-soam-
pm:thresholds/mef-soam-pm:measured-flr-backward {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-loss-measurement/mef-soam-pm:input/mef-soam-
pm:thresholds/mef-soam-pm:forward-high-loss {
    deviate not-supported;
```


Copyright© 2022 Ciena® Corporation

```
  }

  deviation /mef-soam-pm:create-loss-measurement/mef-soam-pm:input/mef-soam-
pm:thresholds/mef-soam-pm:forward-consecutive-high-loss {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-loss-measurement/mef-soam-pm:input/mef-soam-
pm:thresholds/mef-soam-pm:backward-high-loss {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-loss-measurement/mef-soam-pm:input/mef-soam-
pm:thresholds/mef-soam-pm:backward-consecutive-high-loss {
    deviate not-supported;
  }


  deviation /mef-soam-pm:create-loss-measurement/mef-soam-pm:input/mef-soam-
pm:thresholds/mef-soam-pm:forward-unavailable-count {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-loss-measurement/mef-soam-pm:input/mef-soam-
pm:thresholds/mef-soam-pm:forward-available-ratio {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-loss-measurement/mef-soam-pm:input/mef-soam-
pm:thresholds/mef-soam-pm:backward-unavailable-count {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-loss-measurement/mef-soam-pm:input/mef-soam-
pm:thresholds/mef-soam-pm:backward-available-ratio {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-loss-measurement/mef-soam-pm:input/mef-soam-
pm:repetition-period {
    deviate replace {
      type yang:timeticks {
        range "0..259200000";
      }
    }
  }

  // Deviations for rpcs related to delay measurement

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:measurement-type {
    deviate replace {
      type enumeration {
        enum dmm {
          description
           "DMM SOAM PDU generated, DMR responses received (one-way or two-way
              measurements).";
        }
      }
    }
```


Copyright© 2022 Ciena® Corporation

```
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:version {

    deviate not-supported;
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:measurement-enable {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:data-pattern {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:test-tlv-included {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:test-tlv-pattern {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:number-intervals-stored {
    deviate replace {
      type uint32 {
        range "2..96";
      }
    }
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:align-measurement-intervals {
    deviate replace {
      default false;
    }
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:align-measurement-offset {
    deviate replace {
      type uint32 {
        range "0..60";
      }
    }
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:bins-per-fd-interval {
    deviate replace {
      type uint32 {
        range "2..10";
      }
    }
```


Copyright© 2022 Ciena® Corporation

```
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:bins-per-ifdv-interval {
    deviate replace {
      type uint32 {
        range "2..10";
      }
    }
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:ifdv-selection-offset {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:bins-per-fdr-interval {
    deviate replace {
      type uint32 {
        range "2..10";
      }
    }
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:thresholds/mef-soam-pm:enabled-thresholds {
    deviate replace {
      type bits {
        bit measured-frame-delay-two-way;
        bit max-frame-delay-two-way;
        bit measured-inter-frame-delay-variation-two-way;
        bit max-inter-frame-delay-variation-two-way;
      }
    }
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:thresholds/mef-soam-pm:average-frame-delay-two-way {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:thresholds/mef-soam-pm:average-inter-frame-delay-variation-two-way {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:thresholds/mef-soam-pm:max-frame-delay-range-two-way {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:thresholds/mef-soam-pm:average-frame-delay-range-two-way {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:thresholds/mef-soam-pm:measured-frame-delay-forward {
    deviate not-supported;
```


Copyright© 2022 Ciena® Corporation

```
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:thresholds/mef-soam-pm:max-frame-delay-forward {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:thresholds/mef-soam-pm:average-frame-delay-forward {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:thresholds/mef-soam-pm:measured-inter-frame-delay-variation-forward
{
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:thresholds/mef-soam-pm:max-inter-frame-delay-variation-forward {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:thresholds/mef-soam-pm:average-inter-frame-delay-variation-forward {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:thresholds/mef-soam-pm:max-frame-delay-range-forward {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:thresholds/mef-soam-pm:average-frame-delay-range-forward {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:thresholds/mef-soam-pm:measured-frame-delay-backward {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:thresholds/mef-soam-pm:max-frame-delay-backward {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:thresholds/mef-soam-pm:average-frame-delay-backward {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:thresholds/mef-soam-pm:measured-inter-frame-delay-variation-backward
{
    deviate not-supported;
  }
```


Copyright© 2022 Ciena® Corporation

```
  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:thresholds/mef-soam-pm:max-inter-frame-delay-variation-backward {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:thresholds/mef-soam-pm:average-inter-frame-delay-variation-backward
{
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:thresholds/mef-soam-pm:max-frame-delay-range-backward {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:thresholds/mef-soam-pm:average-frame-delay-range-backward {
    deviate not-supported;
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:repetition-period {
    deviate replace {
      type yang:timeticks {
        range "0..259200000";
      }
    }
  }

  // Deviations for notifications related to mef-soam-pm

  deviation /mef-soam-pm:availability-change-alarm {

    deviate not-supported;
  }

  deviation /mef-soam-pm:threshold-crossing-alarm/mef-soam-pm:suspect {

    deviate not-supported;
  }
}
```


Copyright© 2022 Ciena® Corporation

mef-soam-pm-sds-deviation.yang

```
module mef-soam-pm-sds-deviation {
  namespace "http://www.ciena.com/ns/yang/mef-soam-pm-sds-deviation";
  prefix "mef-soam-pm-sds-deviation";

  import mef-soam-pm {
    prefix "mef-soam-pm";
    revision-date "2012-04-16";
  }

  organization
    "Ciena Corporation";

  contact
    "Web URL: http://www.ciena.com/
     E-mail:  yang@ciena.com
     Postal:  7035 Ridge Road
              Hanover, Maryland 21076
              U.S.A.
     Phone:   +1 800-921-1144
     Fax:     +1 410-694-5750";

  description
    "This YANG module contains the SAOS SDS platform specific deviations for
mef-soam-pm.yang model.";

  revision 2017-09-01 {
    description
      "Initial version";
  }

  deviation /mef-soam-pm:create-loss-measurement/mef-soam-pm:input/mef-soam-
pm:frame-size {
    deviate replace {
      type uint32 {
        range "64..2000";
      }
    }
    description
     "This object specifies the Loss Measurement frame size between 64 bytes
      and the maximum transmission unit of the EVC.

      The range of frame sizes from 64 through 2000 octets need to be
      supported.

      The adjustment to the frame size of the standard frame size is
      accomplished by the addition of a Data or Test TLV. A Data or
      Test TLV is only added to the frame if the frame size is greater
      than 64 bytes.";
  }

  deviation /mef-soam-pm:create-delay-measurement/mef-soam-pm:input/mef-
soam-pm:frame-size {
    deviate replace {
      type uint32 {
        range "140..2000";
      }
    }
    deviate delete {
```


Copyright© 2022 Ciena® Corporation

```
      default 64;
    }
    description
     "This object specifies the Delay Measurement frame size between 140 bytes
      and the maximum transmission unit of the EVC.

      The range of frame sizes from 140 through 2000 octets need to be
      supported.

      The adjustment to the frame size of the standard frame size is
      accomplished by the addition of a Data or Test TLV. A Data or
      Test TLV is only added to the frame if the frame size is greater
      than 140 bytes.";
  }
}
```


Copyright© 2022 Ciena® Corporation

mef-soam-pm.yang

```
module mef-soam-pm {
  namespace "http://metroethernetforum.org/ns/yang/mef-soam-pm";
  prefix mef-soam-pm;

  import mef-cfm {
    prefix cfm;
    revision-date 2012-04-16;
  }

  import ietf-yang-types {
    prefix yang;
    revision-date 2010-09-24;
  }

  organization "Metro Ethernet Forum";
  contact
    "Web URL: http://metroethernetforum.org/
         E-mail:  mibs@metroethernetforum.org
         Postal:  Metro Ethernet Forum
                  6033 W. Century Boulevard, Suite 830
                  Los Angeles, CA 90045
                  U.S.A.
         Phone:   +1 310-642-2800
         Fax:     +1 310-642-2808";

  description
    "This YANG module contains the management data definitions for the
         management of Ethernet Services Operations, Administration and
         Maintenance for Performance Monitoring and extends the Connectivity
         Fault Management (CFM) YANG modules.

         Reference Overview:
               A number of base documents have been used to create th
               SOAM PM YANG Module.
             The following are the abbreviations for the baseline documents:
             [802.1q] refers to IEEE 802.1Q-2011 'IEEE Standard for Local and
                   metro-politan area networks --Media Access Control (MAC)
                Bridges and Virtual Bridged Local Area Networks, August 2011
              [MEF SOAM PM IA] refers to 'Service OAM Performance Monitoring
                   Implementation Agreement', January 2012
             [MEF SOAM PM MIB] refers to 'Service OAM Performance Monitoring
                   Definition of Managed Objects', January 2012
              [Y.1731] refers to ITU-T Y.1731 'OAM functions and mechanisms
                   for Ethernet based networks', July 2011";

  revision 2012-04-16 {
    description
      "Initial revision";
    reference
      "Service OAM Performance Monitoring YANG Module (MEF XX), April 2012";
  }

  //
  // Type definitions related to MEF SOAM PM
  //

  typedef suspect-status-type {
    type boolean;
```


Copyright© 2022 Ciena® Corporation

```
    description
      "This boolean data type indicates whether the measurement interval
             has been marked as suspect.

             The object is set to false at the start of a measurement
             interval. It is set to true when there is a discontinuity in the
             performance measurements during the measurement interval.
             Conditions for a discontinuity include, but are not limited to
             the following:

               1 - The local time-of-day clock is adjusted by at least 10
                   seconds
             2 - The conducting of a performance measurement is halted before
                   the current measurement interval is completed
               3 - A local test, failure, or reconfiguration that disrupts
                   service";
    reference
      "[MEF SOAM PM IA] R39, R40, 41 and R42";
  }

  typedef performance-monitoring-interval-type {
    type uint32 {
      range "3..3600000";
    }
    units ms;
    description
      "This integer data type indicates the transmission time between the
       SOAM PM frames for session, in ms.";
  }
  typedef test-pattern-type {
    type enumeration {
      enum null-signal-without-crc-32 {
        description
          "This test pattern is a Null signal without CRC-32.";
      }
      enum null-signal-with-crc-32 {
        description
          "This test pattern is a Null signal with CRC-32.";
      }
      enum prbs-2311-without-crc-32 {
        description "This test pattern is a PRBS 2^31-1 without CRC-32.";
      }
      enum prbs-2311-with-crc-32 {
        description "This test pattern is a PRBS 2^31-1 with CRC-32.";
      }
    }

    description
     "This enumeration data type indicates the type of test pattern to be sent
       in an OAM PDU Data TLV.";
    reference
      "MEF-SOAM-TC-MIB.MefSoamTcTestPatternType";
  }

  typedef session-status-type {
    type enumeration {
      enum active {
        description
          "Indicates the measurement instance is active.";
      }
```


Copyright© 2022 Ciena® Corporation

```
      enum not-active {
        description
          "Indicates the measurement instance is not active.";
      }
    }

    description
     "This enumeration data type defines the status of PM session of a MEP.";
    reference
      "MEF-SOAM-TC-MIB.MefSoamTcStatusType";
  }

  typedef availability-type {
    type enumeration {
      enum available {
        description
          "Indicates the MEP is available.";
      }
      enum unavailable {
        description
          "Indicates the MEP is unavailable.";
      }
      enum unknown {
        description
          "Indicates the availability is not known, for instance because
           insufficient time has passed to make an availability
           calculation, the time has been excluded because of a
           maintenance interval, or because availability measurement is
           not enabled.";
      }
    }

    description
      "This enumeration data type defines the availability of a session,
       measured by a loss measurement session.";
    reference
      "MEF-SOAM-TC-MIB.MefSoamTcAvailabilityType";
  }

  typedef measurement-bin-type {
    type enumeration {
      enum two-way-frame-delay {
        description
          "Indicates a measurement bin for two-way Frame Delay.";
      }
      enum forward-frame-delay {
        description
         "Indicates a measurement bin for one-way Frame Delay in the forward
           direction.";
      }
      enum backward-frame-delay {
        description
         "Indicates a measurement bin for one-way Frame Delay in the backward
           direction.";
      }
      enum two-way-inter-frame-delay-variation {
        description
          "Indicates a measurement bin for two-way Inter-frame Delay
           Variation.";
      }
```


Copyright© 2022 Ciena® Corporation

```
      enum forward-inter-frame-delay-variation {
        description
          "Indicates a measurement bin for one-way Inter-frame Delay
           Variation in the forward direction.";
      }
      enum backward-inter-frame-delay-variation {
        description
         "Indicates a measurement bin for one-way Inter-frame Delay Variation
           in the backward direction.";
      }
      enum two-way-frame-delay-range {
        description
          "Indicates a measurement bin for two-way Frame Delay Range.";
      }
      enum forward-frame-delay-range {
        description
          "Indicates a measurement bin for one-way Frame Delay Range in the
           forward direction.";
      }
      enum backward-frame-delay-range {
        description
          "Indicates a measurement bin for one-way Frame Delay Range in the
           backward direction.";
      }
    }

    description
     "This enumeration data type indicates whether the bin number is for Frame
      Delay, Inter-Frame Delay Variation or Frame Delay Range.";
    reference
      "MEF-SOAM-TC-MIB.MefSoamTcDelayMeasurementBinType";
  }

  //
  // Groupings related to MEF SOAM PM
  //

  grouping remote-mep-group {
    description
      "This grouping includes objects which identify a remote MEP.";
    choice remote-mep {
      mandatory true;
      description
       "The remote MEP can be identified by either a MAC address or a MEP ID";

      case mac-address {
        leaf mac-address {
          type yang:mac-address;

          description
            "The Target MAC Address Field to be transmitted: A unicast
             destination MAC address.

             This object is only valid for the entity transmitting the
             SOAM Loss and Delay Measurement frames and is ignored by
             the entity receiving SOAM Loss and Delay Measurement
             frames.";
        }
      }
      case mep-id {
```


Copyright© 2022 Ciena® Corporation

```
        leaf mep-id {
          type string;

          description
           "The Maintenance Association End Point Identifier of another MEP in
              the same Maintenance Association to which the SOAM Loss or Delay
              Measurement frame is to be sent.

              This object is only valid for the entity transmitting the
              SOAM Loss Measurement or Delay Measurement frames and is
              ignored by the entity receiving SOAM Loss Measurement or
              Delay Measurement frames.";
        }
      }
    }
  }

  grouping measurement-timing-group {
    description
      "This grouping includes objects used for proactive and on-demand
       scheduling of PM measurement sessions.";

    container start-time {
      description
        "This container defines the session start time";

      choice start-time {
        mandatory true;
        description
          "Measurement session start time can be immediate, relative or
           absolute.";

        container immediate {
          presence "Start the measurement session immediately.";

          description
           "This object specifies the start time to be immediately at the time
              of session creation.";
        }
        leaf relative {
          type yang:timeticks;

          description
            "This object specifies the relative start time, from the current
             system time, to perform on-demand ETH-DM.  This attribute has no
              meaning for proactive ETH-DM.";
        }
        leaf absolute {
          type yang:date-and-time;

          description
           "This object specifies the scheduled start date/time to perform the
              on-demand Performance Monitoring OAM operations. This attribute
              has no meaning for proactive Performance Monitoring OAM
              operations.";
        }
      }
    }

    container stop-time {
```


Copyright© 2022 Ciena® Corporation

```
    description
      "This container defines the session stop time";

    choice stop-time {
      mandatory true;
      description
        "Measurement session stop time can be none, relative or
         absolute.";

      container none {
        presence "Never end the measurement session.";

        description
          "This object specifies the measurement  session to never end.";
      }

      leaf relative {
        type yang:timeticks;
        mandatory true;

        description
          "This object specifies the duration of the Delay Measurement PM
           Session.  The duration time can be specified as forever
           (represented by a zero value) or as relative time (e.g., a
           given number of hours, minutes, and seconds from the start
           time).  If the duration time is relative time, then the
           duration time should be equal to or greater than the frame
           transmission period of the PM function(s) comprising the PM
           Solution";

      }

      leaf absolute {
        type yang:date-and-time;

        description
         "This object specifies the scheduled stop date and time to perform
           on-demand Performance Monitoring OAM operations.  This
           attribute has no meaning for proactive Performance
           Monitoring OAM operations.  The stop date and time value should
           be greater than or equal to the scheduled start date and time
           value.";
      }
    }
  }


  leaf repetition-period {
    type yang:timeticks;
    default 0;

    description
      "This object specifies a configurable repetition time between
       Measurement Intervals in a Measurement session in hundredths of a
       second.

       If the value is less than or equal to one Measurement Interval
       there is no time gap between the end of one Measurement
       Interval and the start of a new Measurement Interval. This is
       the normal usage case.
```


Copyright© 2022 Ciena® Corporation

```
        If the value is greater than one Measurement Interval there is
        time gap between the end of one Measurement Interval and the
        start of the next Measurement Interval.";
    reference
      "[MEF SOAM PM IA] R18, D3, R19, R20";
  }
}

grouping loss-measurement-configuration-group {
  description
    "This grouping includes configuration objects for the Frame Loss
     Measurement function defined in [Y.1731] and [MEF SOAM PM IA].";
  reference
    "MEF-SOAM-PM-MIB.mefSoamLmCfgTable, [Y.1731] and [MEF SOAM PM IA]";

  leaf measurement-type {
    type enumeration {
      enum lmm {
        description
          "LMM SOAM PDU generated and received LMR responses tracked.";
      }
      enum slm {
        description
          "SLM SOAM PDU generated and received SLR responses tracked.";
      }
      enum ccm {
        description
          "CCM SOAM PDU generated and received CCM PDUs tracked.";
      }
    }
    default slm;

    description
      "This object specifies what type of Loss Measurement will be
       performed.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmCfgType";
  }

  leaf version {
    type enumeration {
      enum Y.1731-2008 {
        description
          "Indicates the PDU formats defined in Y.1731-2008";
      }
      enum Y.1731-2011 {
        description
          "Indicates the PDU formats defined in Y.1731-2011.";
      }
    }
    default Y.1731-2008;

    description
      "This object indicates the version of the PDUs used to perform Loss
       Measurement.

       The exact PDUs to use are specified by this object in
       combination with measurement-type.";
    reference
```


Copyright© 2022 Ciena® Corporation

```
        "MEF-SOAM-PM-MIB.mefSoamLmCfgVersion";
    }

    leaf enabled-counters {
      type bits {
        bit forward-transmitted-frames;
        bit forward-received-frames;
        bit forward-min-flr;
        bit forward-max-flr;
        bit forward-average-flr;
        bit backward-transmitted-frames;
        bit backward-received-frames;
        bit backward-min-flr;
        bit backward-max-flr;
        bit backward-average-flr;
        bit soam-pdus-sent;
        bit soam-pdus-received;
        bit availability-forward-high-loss;
        bit availability-forward-consecutive-high-loss;
        bit availability-forward-available;
        bit availability-forward-unavailable;
        bit availabilility-forward-min-flr;
        bit availability-forward-max-flr;
        bit availability-forward-average-flr;
        bit availability-backward-high-loss;
        bit availability-backward-consecutive-high-loss;
        bit availability-backward-available;
        bit available-backward-unavailable;
        bit available-backward-min-flr;
        bit available-backward-max-flr;
        bit available-backward-average-flr;
        bit measured-stats-forward-measured-flr;
        bit measured-stats-backward-measured-flr;
        bit measured-stats-availability-forward-status;
        bit measured-stats-availability-backward-status;
      }
      default " ";

      description
       "A vector of bits that indicates the type of SOAM LM counters found in
         the current-stats and history-stats that are enabled.

         A present bit enables the specific SOAM LM counter. A not present
         bit disables the SOAM LM counter.

         If a particular SOAM LM counter is not supported the BIT value
         is not present.

         Not all SOAM LM counters are supported for all SOAM LM types. ";
      reference
        "MEF-SOAM-PM-MIB.mefSoamLmCfgMeasurementEnable";
    }

    leaf message-period {
      type performance-monitoring-interval-type;
      units ms;
      default 1000;

      description
      "This object specifies the interval between Loss Measurement OAM message
```

```
        transmission. For Loss Measurement monitoring applications the
        default value is 1 sec.

        This object is not applicable if measurement-type is set to
        'ccm' and is ignored for that Loss Measurement Type.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmCfgMessagePeriod";
  }

  leaf priority {
    type cfm:priority-type;
    mandatory true;

    description
      "This object specifies the priority of frames with Performance
       Monitoring OAM message information.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmCfgPriority";
  }

  leaf frame-size {
    type uint32 {
      range "64 .. 9600";
    }
    units bytes;
    default 64;

    description
     "This object specifies the Loss Measurement frame size between 64 bytes
        and the maximum transmission unit of the EVC.

        The range of frame sizes from 64 through 2000 octets need to be
        supported, and the range of frame sizes from 2001 through 9600
        octets is suggested be supported.

      The adjustment to the frame size of the standard frame size is
      accomplished by the addition of a Data or Test TLV. A Data or
      Test TLV is only added to the frame if the frame size is greater
      than 64 bytes.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmCfgFrameSize";
  }

  leaf data-pattern {
    type enumeration {
      enum zeroes {
        description
          "Indicates the Data TLV contains all zeros.";
      }
      enum ones {
        description
          "Indicates the Data TLV contains all ones.";
      }
    }
    default zeroes;

    description
      "This object specifies the LM data pattern included in a Data TLV when
        the size of the LM frame is determined by the frame-size object
        and test-tlv-included is 'false'.
```


Copyright© 2022 Ciena® Corporation

```
        If the frame size object does not define the LM frame size or
        test-tlv-included is 'true' the value of this object is ignored.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmCfgDataPattern";
  }

  leaf test-tlv-included {
    type boolean;
    default false;

    description
     "This object indicates whether a Test TLV or Data TLV is included when
        the size of the LM frame is determined by the
        frame-size object.

        A value of 'true' indicates that the Test TLV is to be
        included. A value of 'false' indicates that the Data TLV is to
        be included.

        If the frame-size object does not define the LM frame size the
        value of this object is ignored.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmCfgTestTlvIncluded";
  }

  leaf test-tlv-pattern {
    type test-pattern-type;
    default null-signal-without-crc-32;

    description
      "This object specifies the type of test pattern to be sent in the LM
        frame Test TLV when the size of LM PDU is determined by the
        frame-size object and test-tlv-included is 'true'. If the frame
        size object does not define the LM frame size or
        test-tlv-included is 'false' the value of this object is
        ignored.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmCfgTestTlvPattern";
  }

  leaf measurement-interval {
    type uint32 {
      range "1..525600";
    }
    units minutes;
    default 15;

    description
     "This object specifies the Measurement Interval for FLR statistics, in
        minutes.

        A Measurement Interval of 15 minutes needs to be supported,
        other intervals may be supported.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmCfgMeasurementInterval";
  }

  leaf number-intervals-stored {
    type uint32 {
```


Copyright© 2022 Ciena® Corporation

```
      range "2..1000";
    }
    default 32;
    description
     "This object specifies the number of completed measurement intervals to
        store in the history statistic table.

        At least 32 completed measurement intervals are to be
        stored. 96 measurement intervals are recommended to be stored.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmCfgNumIntervalsStored";
  }

  leaf align-measurement-intervals {
    type boolean;
    default true;

    description
      "This object specifies whether the measurement intervals for the Loss
        Measurement session are aligned with a zero offset to real
        time.

        The value 'true' indicates that each Measurement Interval
        starts at a time which is aligned to NE time source hour if the
        interval is a factor of an hour, i.e. 60min/15min = 4. For
        instance, a measurement time interval of 15 minutes would
        stop/start the measurement interval at 0, 15, 30, and 45
        minutes of an hour. A measurement interval of 7 minutes would
        not align to the hour since 7 minutes is NOT a factor of an
        hour, i.e.  60min/7min = 8.6, and the behavior is the same as
        if the object is set to 'false'.

        The value 'false' indicates that each Measurement Interval
        starts at a time which is indicated by repetition-period.

        One side effect of the usage of this parameter is that if the
        value is true and the repetition-period is not a factor of an
        hour then the start of the next Measurement Interval will be
        delayed until the next factor of an hour.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmCfgAlignMeasurementIntervals";
  }

  leaf align-measurement-offset {
    type uint32 {
      range "0..525600";
    }
    units minutes;
    default 0;

    description
     "This object specifies the offset in minutes from the time of day value
        if align-measurement-intervals is 'true' and the repetition
        time is a factor of 60 minutes. If not, the value of this
        object is ignored.

        If the Measurement Interval is 15 minutes and
        align-measurement-intervals is true and if this object was set
        to 5 minutes, the Measurement Intervals would start at 5, 20,
        35, 50 minutes past each hour.";
```


Copyright© 2022 Ciena® Corporation

```
      reference
        "MEF-SOAM-PM-MIB.mefSoamLmCfgAlignMeasurementOffset";
    }

    leaf availability-measurement-interval {
      type uint32 {
        range "1..525600";
      }
      units minutes;
      default 15;

      description
      "This object specifies the availability measurement interval in minutes.

         A measurement interval of 15 minutes is to be supported, other
         intervals can be supported.";
      reference
        "MEF-SOAM-PM-MIB.mefSoamLmCfgAvailabilityMeasurementInterval";
    }
    leaf availability-number-consecutive-flr-measurements {
      type uint32 {
        range "1..1000000";
      }
      default 10;

      description
        "This object specifies a configurable number of consecutive loss
         measurement PDUs to be used in evaluating the
         availability/unavailability status of each availability
         indicator per MEF 10.2.1.  Loss Measurement PDUs (LMMs, CCMs or
         SLMs) are sent regularly with a period defined by
         message-period.  Therefore, this object, when multiplied by
         message-period, is equivalent to the Availability parameter of
         'delta_t' as specified by MEF 10.2.1.

         If the measurement-type is lmm or ccm, this object defines the
         number of LMM or CCM PDUs transmitted during each 'delta_t'
         period.  The Availability flr for a given 'delta_t' can be
         calculated based on the counters in the last LMM/R or CCM
         during this 'delta_t' and the last LMM/R or CCM in the previous
         'delta_t'.

          If the measurement-type is slm, this object defines the number
          of SLM PDUs transmitted during each 'delta_t' period.  The
          Availability flr for a given 'delta_t' is calculated based on
          the number of those SLM PDUs that are lost.

          If the measurement-type is lmm or ccm, the number range of 1
          through 10 must be supported. The number range of 10 through
          1000000 may be supported, but is not mandatory.

          If the measurement-type is slm, the number range of 10 through
          100 must be supported. The number range of 100 through 1000000
          may be supported, but is not mandatory.";
      reference
        "MEF-SOAM-PM-MIB.mefSoamLmCfgAvailabilityNumConsecutiveMeasPdus";
    }
    leaf availability-flr-threshold {
      type uint32 {
        range "0..100000";
```


Copyright© 2022 Ciena® Corporation

```
    }
    units milli-percent;
    default 50000;

    description
     "This object specifies a configurable availability threshold to be used
        in evaluating the availability/unavailability status of an
        availability indicator per MEF 10.2.1. The availability
        threshold range of 0.00 (0) through 1.00 (100000) is
        supported. This parameter is equivalent to the Availability
        parameter of 'C' as specified by MEF 10.2.1.

        Units are in milli-percent, where 1 indicates 0.001 percent.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmCfgAvailabilityFlrThreshold";
  }

  leaf availability-number-consecutive-intervals {
    type uint32 {
      range "1..1000";
    }
    default 10;

    description
    "This object specifies a configurable number of consecutive availability
        indicators to be used to determine a change in the availability
        status as indicated by MEF 10.2.1. This parameter is equivalent
        to the Availability parameter of 'n' as specified by MEF
        10.2.1.

        The number range of 1 through 10 must be supported. The number
        range of 1 through 1000 may be supported, but is not mandatory.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmCfgAvailabilityNumConsecutiveIntervals";
  }

  leaf availability-number-consecutive-high-flr {
    type uint32 {
      range "1..1000";
    }
    default 5;

    description
    "This object specifies a configurable number of consecutive availability
        indicators to be used for assessing CHLI.  This parameter is
        equivalent to the Resiliency parameter of 'p' as specified by
        MEF 10.2.1.

        Availability-consecutive-high-flr must be strictly less than
        availability-number-consecutive-intervals. If not, the count of
        high loss intervals over time, and the count of consecutive
        high loss levels, is disabled.

        The number range of 1 through 10 must be supported. The number
        range of 1 through 1000 may be supported, but is not mandatory.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmCfgAvailabilityNumConsecutiveHighFlr";
  }

  leaf session-type {
```


Copyright© 2022 Ciena® Corporation

```
      type enumeration {
        enum proactive {
          description
            "The current session is 'proactive'";
        }
        enum on-demand {
          description
            "The current session is 'on-demand'";
        }
      }

      description
        "This object indicates whether the current session is defined to be
         'proactive' or 'on-demand'.";
      reference
        "[MEF SOAM IA] R3";
    }

    list thresholds {
      key id;
      description
      "This list contains the list of Loss Measurement configuration threshold
         values for LM Performance Monitoring.

         The main purpose of the threshold configuration list is to
         configure threshold alarm notifications indicating that a
         specific performance metric is not being met.";

      leaf id {
        type uint32;
        description
          "The index of the threshold number for the specific LM threshold
           entry.

           An index value of '1' needs to be supported. Other index
           values can also be supported.";
      }

      leaf enabled-thresholds {
        type bits {
          bit measured-flr-forward;
          bit max-flr-forward;
          bit average-flr-forward;
          bit measured-flr-backward;
          bit max-flr-backward;
          bit average-flr-backward;
          bit forward-high-loss;
          bit forward-consecutive-high-loss;
          bit backward-high-loss;
          bit backward-consecutive-high-loss;
          bit forward-unavailable-count;
          bit forward-available-ratio;
          bit backward-unavailable-count;
          bit backward-available-ratio;
        }
        default " ";
        description
          "A vector of bits that indicates the type of SOAM LM thresholds
           notifications that are enabled.
```

```
      A present but enables the specific SOAM LM threshold
      notifica-tion and when the specific counter is enabled and
      the threshold is crossed a notification is generated.

      A not present bit disables the specific SOAM LM threshold
      notification.

      If a particular SOAM LM threshold is not supported the BIT
      value is not present.";
}

leaf measured-flr-forward {
  type uint32 {
    range "0..100000";
  }
  units milli-percent;
  default 100000;

  description
    "This object is used to set the measured forward frame loss ratio
     threshold value that will be used to determine if a threshold
     notification is generated.";
  reference
   "MEF-SOAM-PM-MIB.mefSoamLmThresholdCfgMeasuredFlrForwardThreshold";
}

leaf max-flr-forward {
  type uint32 {
    range "0..100000";
  }
  units milli-percent;
  default 100000;

  description
    "This object is used to set the maximum forward frame loss ratio
     threshold value that will be used to determine if a threshold
     notification is generated.";
  reference
    "MEF-SOAM-PM-MIB.mefSoamLmThresholdCfgMaxFlrForwardThreshold";
}

leaf average-flr-forward {
  type uint32 {
    range "0..100000";
  }
  units milli-percent;
  default 100000;

  description
    "This object is used to set the average forward frame loss ratio
     threshold value that will be used to determine if a threshold
     notification is generated.";
  reference
    "MEF-SOAM-PM-MIB.mefSoamLmThresholdCfgAvgFlrForwardThreshold";
}

leaf measured-flr-backward {
  type uint32 {
    range "0..100000";
  }
```


Copyright© 2022 Ciena® Corporation

```
      units milli-percent;
      default 100000;

      description
        "This object is used to set the measured backward frame loss ratio
         threshold value that will be used to determine if a threshold
         notification is generated.";
      reference
      "MEF-SOAM-PM-MIB.mefSoamLmThresholdCfgMeasuredFlrBackwardThreshold";
    }

    leaf max-flr-backward {
      type uint32 {
        range "0..100000";
      }
      units milli-percent;
      default 100000;

      description
        "This object is used to set the maximum backward frame loss ratio
         threshold value that will be used to determine if a threshold
         notification is generated.";
      reference
        "MEF-SOAM-PM-MIB.mefSoamLmThresholdCfgMaxFlrBackwardThreshold";
    }
    leaf average-flr-backward {
      type uint32 {
        range "0..100000";
      }
      units milli-percent;
      default 100000;

      description
        "This object is used to set the average backward frame loss ratio
         threshold value that will be used to determine if a threshold
         notification is generated.";
      reference
        "MEF-SOAM-PM-MIB.mefSoamLmThresholdCfgAvgFlrBackwardThreshold";
    }

    leaf forward-high-loss {
      type uint32;
      default 2147483647;

      description
      "This object is used to set the forward high loss threshold value that
         will be used to determine if a threshold notification is
         generated.";
      reference
        "MEF-SOAM-PM-MIB.mefSoamLmThresholdCfgForwardHighLossThreshold";
    }
    leaf forward-consecutive-high-loss {
      type uint32;
      default 2147483647;

      description
        "This object is used to set the consecutive forward high loss
         threshold value that will be used to determine if a threshold
         notification is generated.";
      reference
```


Copyright© 2022 Ciena® Corporation

```
      "MEF-SOAM-PM-MIB." +
      "mefSoamLmThresholdCfgForwardConsecutiveHighLossThreshold";
  }
  leaf backward-high-loss {
    type uint32;
    default 2147483647;

    description
      "This object is used to set the backward high loss threshold value
       that will be used to determine if a threshold notification is
       generated.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmThresholdCfgBackwardHighLossThreshold";
  }
  leaf backward-consecutive-high-loss {
    type uint32;
    default 2147483647;

    description
      "This object is used to set the consecutive backward high loss
       threshold value that will be used to determine if a threshold
       notification is generated.";
    reference
      "MEF-SOAM-PM-MIB." +
      "mefSoamLmThresholdCfgBackwardConsecutiveHighLossThreshold";
  }

  leaf forward-unavailable-count {
    type uint32;
    default 2147483647;

    description
    "This object is used to set the forward unavailability threshold value
       that will be used to determine if a threshold notification is
       generated.";
    reference
    "MEF-SOAM-PM-MIB.mefSoamLmThresholdCfgForwardUnavailCountThreshold";
  }

  leaf forward-available-ratio {
    type uint32 {
      range "0..100000";
    }
    units milli-percent;
    default 0;

    description
     "This object is used to set the forward availability/total time ratio
       threshold value that will be used to determine if a
       thresh-old notification is generated if the ratio drops below
       the configured value.

       The ratio value is expressed as a percent with a value of 0
       (ra-tio 0.00) through 100000 (ratio 1.00)

       Units are in milli-percent, where 1 indicates 0.001 percent.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmThresholdCfgForwardAvailRatioThreshold";
  }
```


Copyright© 2022 Ciena® Corporation

```
      leaf backward-unavailable-count {
        type uint32;
        default 2147483647;
        description
          "This object is used to set the backward unavailability threshold
          value that will be used to determine if a threshold notification is
           generated.";
        reference
          "MEF-SOAM-PM-
MIB.mefSoamLmThresholdCfgBackwardUnavailCountThreshold";
      }

      leaf backward-available-ratio {
        type uint32 {
          range "0..100000";
        }
        units milli-percent;
        default 0;

        description
        "This object is used to set the backward availability/total time ratio
           threshold value that will be used to determine if a
           thresh-old notification is generated if the ratio drops below
           the configured value.

           The ratio value is expressed as a percent with a value of 0
           (ratio 0.00) through 100000 (ratio 1.00)

           Units are in milli-percent, where 1 indicates 0.001 percent.";
        reference
         "MEF-SOAM-PM-MIB.mefSoamLmThresholdCfgBackwardAvailRatioThreshold";
      }
    }
  }

  grouping loss-availability-stats-group {
    description
      "This grouping includes availability statistics objects for a SOAM Loss
       Measurement session.";

    leaf suspect-status {
      type suspect-status-type;
      description
       "This object indicates whether the Measurement Interval has been marked
          as suspect.

          The object is set to false at the start of a measurement
          interval. It is set to true when there is a discontinuity in
          the performance measurements during the Measurement Interval.
          Conditions for a discontinuity include, but are not limited to
          the following:

           1 - The local time-of-day clock is adjusted by at least 10 seconds
          2 - The conducting of a performance measurement is halted before the
               current Measurement Interval is completed
         3 - A local test, failure, or reconfiguration that disrupts service";
    }

    leaf forward-high-loss {
      type yang:gauge32;
```


Copyright© 2022 Ciena® Corporation

```
      description
       "This object is the number of high loss intervals (HLI) over time in the
          forward direction.

          The value starts at 0 and increments for every HLI that occurs.
          This parameter is equivalent to 'L Sub T' found in MEF 10.2.1.";
      reference
        "MEF-SOAM-PM-MIB.mefSoamLmCurrentAvailStatsForwardHighLoss";
    }

    leaf backward-high-loss {
      type yang:gauge32;

      description
       "This object is the number of high loss intervals (HLI) over time in the
          backward direction.

          The value starts at 0 and increments for every HLI that occurs.
          This parameter is equivalent to 'L Sub T' found in MEF 10.2.1.";
      reference
        "MEF-SOAM-PM-MIB.mefSoamLmCurrentAvailStatsBackwardHighLoss";
    }

    leaf forward-consecutive-high-loss {
      type yang:gauge32;

      description
        "This object is the number of consecutive high loss intervals (CHLI)
        over time in the forward direction.

        The value starts at 0 and increments for every HLI that occurs.
        This parameter is equivalent to 'B Sub T' found in MEF 10.2.1.";
      reference
        "MEF-SOAM-PM-
MIB.mefSoamLmCurrentAvailStatsForwardConsecutiveHighLoss";
    }

    leaf backward-consecutive-high-loss {
      type yang:gauge32;

      description
        "This object is the number of consecutive high loss intervals (CHLI)
         over time in the backward direction.

          The value starts at 0 and increments for every HLI that occurs.
          This parameter is equivalent to 'B Sub T' found in MEF 10.2.1.";
      reference
        "MEF-SOAM-PM-
MIB.mefSoamLmCurrentAvailStatsBackwardConsecutiveHighLoss";
    }

    leaf forward-available {
      type yang:gauge32;

      description
        "This object contains the number of availability indicators during a
        small time interval evaluated as available (low frame loss) in
        the forward direction by this MEP during this measurement
        interval.";
```


Copyright© 2022 Ciena® Corporation

```
      reference
        "MEF-SOAM-PM-MIB.mefSoamLmCurrentAvailStatsForwardAvailable";
    }

    leaf backward-available {
      type yang:gauge32;

      description
        "This object contains the number of availability indicators during a
         small time interval evaluated as available (low frame loss) in
         the backward direction by this MEP during this measurement
         interval.";
      reference
        "MEF-SOAM-PM-MIB.mefSoamLmCurrentAvailStatsBackwardAvailable";
    }
    leaf forward-unavailable {
      type yang:gauge32;

      description
        "This object contains the number of availability indicators during a
         small time interval evaluated as unavailable (high frame loss)
         in the forward direction by this MEP during this measurement
         interval.";
      reference
        "MEF-SOAM-PM-MIB.mefSoamLmCurrentAvailStatsForwardUnavailable";
    }
    leaf backward-unavailable {
      type yang:gauge32;

      description
        "This object contains the number of availability indicators during a
         small time interval evaluated as unavailable (high frame loss)
         in the backward direction by this MEP during this measurement
         interval.";
      reference
        "MEF-SOAM-PM-MIB.mefSoamLmCurrentAvailStatsBackwardUnavailable";
    }

    leaf forward-min-frame-loss-ratio {
      type uint32 {
        range "0..100000";
      }
      units milli-percent;

      description
        "This object contains the minimum one-way availability flr in the
         forward direction, from among the set of availability flr values
         calculated by the MEP in this Measurement Interval.  There is
         one availability flr value for each 'delta_t' time period
         within the Measurement Interval, as specified in MEF 10.2.1.

         The flr value is a ratio that is expressed as a percent with a
         value of 0 (ratio 0.00) through 100000 (ratio 1.00).

        Units are in milli-percent, where 1 indicates 0.001 percent.";
      reference
        "MEF-SOAM-PM-MIB.mefSoamLmCurrentAvailStatsForwardMinFlr";
    }

    leaf forward-max-frame-loss-ratio {
```


Copyright© 2022 Ciena® Corporation

```
    type uint32 {
      range "0..100000";
    }
    units milli-percent;

    description
      "This object contains the maximum one-way availability flr in the
       forward direction, from among the set of availability flr
       values calculated by the MEP in this Measurement Interval.
       There is one availability flr value for each 'delta_t' time
       period within the Measurement Interval, as specified in MEF
       10.2.1.

       The flr value is a ratio that is expressed as a percent with a
       value of 0 (ratio 0.00) through 100000 (ratio 1.00).

       Units are in milli-percent, where 1 indicates 0.001 percent.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmCurrentAvailStatsForwardMaxFlr";
  }

  leaf forward-average-frame-loss-ratio {
    type uint32 {
      range "0..100000";
    }
    units milli-percent;

    description
      "This object contains the average one-way availability flr in the
       for-ward direction, from among the set of availability flr
       values calculated by the MEP in this Measurement Interval.
       There is one availability flr value for each 'delta_t' time
       period within the Measurement Interval, as specified in MEF
       10.2.1.

       The flr value is a ratio that is expressed as a percent with a
       value of 0 (ratio 0.00) through 100000 (ratio 1.00).

       Units are in milli-percent, where 1 indicates 0.001 percent.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmCurrentAvailStatsForwardAvgFlr";
  }

  leaf backward-min-frame-loss-ratio {
    type uint32 {
      range "0..100000";
    }
    units milli-percent;

    description
      "This object contains the minimum one-way availability flr in the
       backward direction, from among the set of availability flr
       values calculated by the MEP in this Measurement Interval.
       There is one availability flr value for each 'delta_t' time
       period within the Measurement Interval, as specified in MEF
       10.2.1.

      The flr value is a ratio that is expressed as a percent with a
      value of 0 (ratio 0.00) through 100000 (ratio 1.00).
```


Copyright© 2022 Ciena® Corporation

```
        Units are in milli-percent, where 1 indicates 0.001 percent.";
      reference
        "MEF-SOAM-PM-MIB.mefSoamLmCurrentAvailStatsBackwardMinFlr";
    }

    leaf backward-max-frame-loss-ratio {
      type uint32 {
        range "0..100000";
      }
      units milli-percent;

      description
        "This object contains the maximum one-way availability flr in the
         backward direction, from among the set of availability flr
         values calculated by the MEP in this Measurement Interval.
         There is one availability flr value for each 'delta_t' time
         period within the Measurement Interval, as specified in MEF
         10.2.1.

         The flr value is a ratio that is expressed as a percent with a
         value of 0 (ratio 0.00) through 100000 (ratio 1.00).

         Units are in milli-percent, where 1 indicates 0.001 percent.";
      reference
        "MEF-SOAM-PM-MIB.mefSoamLmCurrentAvailStatsBackwardMaxFlr";
    }

    leaf backward-average-frame-loss-ratio {
      type uint32 {
        range "0..100000";
      }
      units milli-percent;

      description
        "This object contains the average one-way availability flr in the
         backward direction, from among the set of availability flr
         values calculated by the MEP in this Measurement Interval.
         There is one availability flr value for each 'delta_t' time
         period within the Measurement Interval, as specified in MEF
         10.2.1.

         The flr value is a ratio that is expressed as a percent with a
         value of 0 (ratio 0.00) through 100000 (ratio 1.00).

         Units are in milli-percent, where 1 indicates 0.001 percent.";
      reference
        "MEF-SOAM-PM-MIB.mefSoamLmCurrentAvailStatsBackwardAvgFlr";
    }
  }

  grouping loss-measurement-stats-group {
    description
      "This grouping includes statistics objects for a SOAM Loss Measurement
       session.";

    leaf suspect-status {
      type suspect-status-type;

      description
        "Whether the Measurement Interval has been marked as suspect.
```


Copyright© 2022 Ciena® Corporation

```
        The object is set to false at the start of a measurement
        interval. It is set to true when there is a discontinuity in
        the performance measurements during the Measurement Interval.
        Conditions for a discontinuity include, but are not limited to
        the following:

         1 - The local time-of-day clock is adjusted by at least 10 seconds
        2 - The conducting of a performance measurement is halted before the
             current Measurement Interval is completed
       3 - A local test, failure, or reconfiguration that disrupts service";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmCurrentStatsSuspect";
  }

  leaf forward-transmitted-frames {
    type yang:gauge32;
    mandatory true;

    description
      "This object contains the number of frames transmitted in the forward
       direction by this MEP.

       For a PM Session of types lmm or ccm this includes Ethernet
       Service Frames and SOAM PDUs that are in a higher MEG level
       only.

       For a PM Session of type slm this includes the count of SOAM
       ETH-SLM frames only.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmCurrentStatsForwardTransmittedFrames";
  }

  leaf forward-received-frames {
    type yang:gauge32;
    mandatory true;

    description
      "This object contains the number of frames received in the forward
       direction by this MEP.

       For a PM Session of types lmm or ccm this includes Ethernet
       Service Frames and SOAM PDUs that are in a higher MEG level only.

       For a PM Session of type slm this includes the count of SOAM
       ETH-SLM frames only.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmCurrentStatsForwardReceivedFrames";
  }

  leaf forward-min-frame-loss-ratio {
    type uint32 {
      range "0..100000";
    }
    units milli-percent;

    description
      "This object contains the minimum one-way frame loss ratio in the
       forward direction calculated by this MEP for this Measurement
       Interval. The FLR value is a ratio that is expressed as a
```


Copyright© 2022 Ciena® Corporation

```
        percent with a value of 0 (ratio 0.00) through 100000 (ratio
        1.00).

        Units are in milli-percent, where 1 indicates 0.001 percent.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmCurrentStatsForwardMinFlr";
  }

  leaf forward-max-frame-loss-ratio {
    type uint32 {
      range "0..100000";
    }
    units milli-percent;

    description
      "This object contains the maximum one-way frame loss ratio in the
       forward direction calculated by this MEP for this Measurement
       Interval. The FLR value is a ratio that is expressed as a
       percent with a value of 0 (ratio 0.00) through 100000 (ratio
       1.00).

       Units are in milli-percent, where 1 indicates 0.001 percent.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmCurrentStatsForwardMaxFlr";
  }

  leaf forward-average-frame-loss-ratio {
    type uint32 {
      range "0..100000";
    }
    units milli-percent;

    description
      "This object contains the average one-way frame loss ratio in the
       forward direction calculated by this MEP for this Measurement
       Interval. The FLR value is a ratio that is expressed as a
       percent with a value of 0 (ratio 0.00) through 100000 (ratio
       1.00).

       Units are in milli-percent, where 1 indicates 0.001 percent.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmCurrentStatsForwardAvgFlr";
  }

  leaf backward-transmitted-frames {
    type yang:gauge32;
    mandatory true;

    description
     "This object contains the number of frames transmitted in the backward
       direction by this MEP.

       For a PM Session of type lmm or ccm this includes Ethernet
       Service Frames and SOAM PDUs that are in a higher MEG level
       only.

       For a PM Session of type slm this includes the count of SOAM
       ETH-SLM frames only.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmCurrentStatsBackwardTransmittedFrames";
```


Copyright© 2022 Ciena® Corporation

```
}

leaf backward-received-frames {
  type yang:gauge32;
  mandatory true;

  description
    "This object contains the number of frames received in the backward
     direction by this MEP.

     For a PM Session of type lmm this includes Ethernet Service
     Frames and SOAM PDUs that are in a higher MEG level only.

     For a PM Session of type slm this includes the count of SOAM
     ETH-SLM frames only.";
  reference
    "MEF-SOAM-PM-MIB.mefSoamLmCurrentStatsBackwardReceivedFrames";
}

leaf backward-min-frame-loss-ratio {
  type uint32 {
    range "0..100000";
  }
  units milli-percent;

  description
    "This object contains the minimum one-way frame loss ratio in the
     backward direction calculated by this MEP for this Measurement
     Interval. The FLR value is a ratio that is expressed as a
     percent with a value of 0 (ratio 0.00) through 100000 (ratio
     1.00).

     Units are in milli-percent, where 1 indicates 0.001 percent.";
  reference
    "MEF-SOAM-PM-MIB.mefSoamLmCurrentStatsBackwardMinFlr";
}

leaf backward-max-frame-loss-ratio {
  type uint32 {
    range "0..100000";
  }
  units milli-percent;

  description
    "This object contains the maximum one-way frame loss ratio in the
     backward direction calculated by this MEP for this Measurement
     Interval. The FLR value is a ratio that is expressed as a
     percent with a value of 0 (ratio 0.00) through 100000 (ratio
     1.00).

     Units are in milli-percent, where 1 indicates 0.001 percent.";
  reference
    "MEF-SOAM-PM-MIB.mefSoamLmCurrentStatsBackwardMaxFlr";
}

leaf backward-average-frame-loss-ratio {
  type uint32 {
    range "0..100000";
  }
  units milli-percent;
```


Copyright© 2022 Ciena® Corporation

```
      description
        "This object contains the average one-way frame loss ratio in the
         backward direction calculated by this MEP for this Measurement
         Interval. The FLR value is a ratio that is expressed as a
         percent with a value of 0 (ratio 0.00) through 100000 (ratio
         1.00).

         Units are in milli-percent, where 1 indicates 0.001 percent.";
      reference
        "MEF-SOAM-PM-MIB.mefSoamLmCurrentStatsBackwardAvgFlr";
    }

    leaf soam-pdus-sent {
      type yang:gauge32;
      description
       "This object contains the count of the number of SOAM PDUs sent during
         this Measurement Interval.

         This object applies when type is lmm, slm or ccm. It indicates
         the number of LMM, CCM, or SLM SOAM frames transmitted.";
      reference
        "MEF-SOAM-PM-MIB.mefSoamLmCurrentStatsSoamPdusSent";
    }

    leaf soam-pdus-received {
      type yang:gauge32;

      description
       "This object contains the count of the number of SOAM PDUs PDUs received
         in this Measurement Interval.

         This object applies when type is lmm, slm, or ccm. This object
         indicates the number of LMR, CCM, or SLR SOAM frames received.";
      reference
        "MEF-SOAM-PM-MIB.mefSoamLmCurrentStatsSoamPdusReceived";
    }
  }

  grouping delay-measurement-configuration-group {
    description
      "This grouping includes configuration objects for the Delay Measurement
       function defined in [Y.1731] and [MEF SOAM PM IA].";
    reference
      "MEF-SOAM-PM-MIB.mefSoamDmCfgTable, [Y.1731], and [MEF SOAM PM IA].";

    leaf measurement-type {
      type enumeration {
        enum dmm {
          description
           "DMM SOAM PDU generated, DMR responses received (one-way or two-way
              measurements).";
        }
        enum dm1-transmitted {
          description
            "1DM SOAM PDU generated (one-way measurements are made by the
             receiver).";
        }
        enum dm1-received {
          description
```


Copyright© 2022 Ciena® Corporation

```
            "1DM SOAM PDU received and tracked (one-way measurements).";
        }
      }
      mandatory true;

      description
        "This object indicates what type of Delay Measurement is to be
         performed.

         The exact PDUs to use are specified by this object in
         combination with version.";
    }

    leaf version {
      type enumeration {
        enum Y.1731-2008 {
          description
            "Indicates the PDU formats defined in Y.1731-2008.";
        }
        enum Y.1731-2011 {
          description
            "Indicates the PDU formats defined in Y.1731-2011.";
        }
      }
      default Y.1731-2008;

      description
        "This object indicates the version of the PDUs used to perform Delay
         Measurement.

         The exact PDUs to use are specified by this object in
         combination with measurement-type.";
      reference
        "[Y.1731]";
    }

    leaf measurement-enable {
      type bits {
        bit soam-pdus-sent;
        bit soam-pdus-received;
        bit frame-delay-two-way-bins;
        bit frame-delay-two-way-min;
        bit frame-delay-two-way-max;
        bit frame-delay-two-way-average;
        bit frame-delay-forward-bins;
        bit frame-delay-forward-min;
        bit frame-delay-forward-max;
        bit frame-delay-forward-average;
        bit frame-delay-backward-bins;
        bit frame-delay-backward-min;
        bit frame-delay-backward-max;
        bit frame-delay-backward-average;
        bit inter-frame-delay-variation-forward-bins;
        bit inter-frame-delay-variation-forward-min;
        bit inter-frame-delay-variation-forward-max;
        bit inter-frame-delay-variation-forward-average;
        bit inter-frame-delay-variation-backward-bins;
        bit inter-frame-delay-variation-backward-min;
        bit inter-frame-delay-variation-backward-max;
        bit inter-frame-delay-variation-backward-average;
```


Copyright© 2022 Ciena® Corporation

```
        bit inter-frame-delay-variation-two-way-bins;
        bit inter-frame-delay-variation-two-way-min;
        bit inter-frame-delay-variation-two-way-max;
        bit inter-frame-delay-variation-two-way-average;
        bit frame-delay-range-forward-bins;
        bit frame-delay-range-forward-max;
        bit frame-delay-range-forward-average;
        bit frame-delay-range-backward-bins;
        bit frame-delay-range-backward-max;
        bit frame-delay-range-backward-average;
        bit frame-delay-range-two-way-bins;
        bit frame-delay-range-two-way-max;
        bit frame-delay-range-two-way-average;
        bit measured-stats-frame-delay-two-way;
        bit measured-stats-frame-delay-forward;
        bit measured-stats-frame-delay-backward;
        bit measured-stats-inter-frame-delay-variation-two-way;
        bit measured-stats-inter-frame-delay-variation-forward;
        bit measured-stats-inter-frame-delay-variation-backward;
      }
      default " ";

      description
        "A vector of bits that indicates the type of SOAM DM counters that
                 are enabled.

                 A present bit enables the specific SOAM DM counter.

                 A not present bit disables the SOAM DM counter.

               If a particular SOAM DM counter is not supported the BIT value
                 is not present.

              Not all SOAM DM counters are supported for all SOAM DM types.";
    }

    leaf message-period {
      type performance-monitoring-interval-type;
      default 100;

      description
        "This object specifies the interval between Delay Measurement OAM
        message transmission. For Delay Measurement monitoring applications,
         the default value is 100ms.";
    }

    leaf priority {
      type cfm:priority-type;
      mandatory true;

      description
        "This object specifies the priority of frames with Performance
         Monitoring OAM message information.";
    }

    leaf frame-size {
      type uint32 {
        range "64 .. 9600";
      }
      default 64;
```


Copyright© 2022 Ciena® Corporation

```
    description
     "This object specifies the Delay Measurement frame size between 64 bytes
        and the maximum transmission unit of the EVC.

        The range of frame sizes from 64 through 2000 octets need to be
        supported, and the range of frame sizes from 2001 through 9600
        octets is suggested to be supported.

        The adjustment to the frame size of the standard frame size is
        accomplished by the addition of a Data or Test TLV. A Data or
        Test TLV is only added to the frame if the frame size is
        greater than 64 bytes.";
  }

  leaf data-pattern {
    type enumeration {
      enum zeroes {
        description "Indicates the Data TLV contains all zeros.";
      }
      enum ones {
        description "Indicates the Data TLV contains all ones.";
      }
    }
    default zeroes;

    description
     "This object specifies the DM data pattern included in a Data TLV when
        the size of the DM frame is determined by the frame-size object
        and test-tlv-included is 'false'.  If the frame size object
        does not define the DM frame size or test-tlv-included is
        'true' the value of this object is ignored.";
  }

  leaf test-tlv-included {
    type boolean;
    default false;

    description
     "Indicates whether a Test TLV or Data TLV is included when the size of
        the LBM frame is determined by the frame-size leaf.

        A value of 'true' indicates that the Test TLV is to be
        included.

        A value of 'false' indicates that the Data TLV is to be
        included.";
    reference
      "[Y.1731] 9.3";
  }

  leaf test-tlv-pattern {
    type test-pattern-type;
    default null-signal-without-crc-32;

    description
      "This object specifies the type of test pattern to be sent in the DM
        frame Test TLV when the size of DM PDU is determined by the
        frame-size object and test-tlv-included is 'true'. If the frame
        size object does not define the DM frame size or
```


Copyright© 2022 Ciena® Corporation

```
        test-tlv-included is 'false' the value of this object can be
        ignored.";
    }

    leaf measurement-interval {
      type uint32 {
        range "1..1440";
      }
      units minutes;
      default 15;

      description
        "This object specifies a Measurement Interval in minutes.

         A Measurement Interval 15 minutes needs to be supported, other
         intervals may be supported.";
      reference
        "MEF-SOAM-PM-MIB.mefSoamDmCfgMeasurementInterval";
    }

    leaf number-intervals-stored {
      type uint32 {
        range "2..1000";
      }
      default 32;

      description
       "This object specifies the number of completed measurement intervals to
        store in the history statistic table.

        At least 32 completed measurement intervals are to be
        stored. 96 measurement intervals are recommended to be stored.";
    }

    leaf align-measurement-intervals {
      type boolean;
      default true;

      description
       "This object specifies whether the Measurement Intervals for the Delay
        Measurement session are aligned with a zero offset to real
        time.

        The value 'true' indicates that each Measurement Interval
        starts at a time which is aligned to NE time source hour, if
        the repetition time (or the Measurement Interval, if the
        repetition time is 0) is a factor of an hour, i.e. 60min/15min
        = 4. For instance, a Measurement Interval/Repetition Time of 15
        minutes would stop/start the Measurement Interval at 0, 15, 30,
        and 45 minutes of an hour. A Measurement Interval/Repetition
        Time of 7 minutes would not align to the hour since 7 minutes
        is NOT a factor of an hour, i.e.  60min/7min = 8.6.  In this
        case the behavior is the same as if the object is set to
        'false'.

        The value 'false' indicates that the first Measurement Interval
        starts at an arbitrary time and each subsequent Measurement
        Interval starts at a time which is determined by
        repetition-time.";
    }
```


Copyright© 2022 Ciena® Corporation

```
leaf align-measurement-offset {
  type uint32 {
    range "0..525600";
  }
  units minutes;
  default 0;

  description
   "This object specifies the offset in minutes from the time of day value
     if align-measurement-intervals is 'true' and the repetition
     time is a factor of 60 minutes. If not, the value of this
     object is ignored.

     If the Measurement Interval is 15 minutes and
     align-measurement-intervals is true and if this object was set
     to 5 minutes, the Measurement Intervals would start at 5, 20,
     35, 50 minutes past each hour.";
}

leaf bins-per-fd-interval {
  type uint32 {
    range "2..100";
  }
  default 3;

  description
    "This object specifies the number of measurement bins per Measurement
     Interval for Frame Delay measurements.

     At least 3 bins are to be supported; at least 10 bins are
     recommended to be supported.";
}

leaf bins-per-ifdv-interval {
  type uint32 {
    range "2..100";
  }
  default 2;

  description
    "This object specifies the number of measurement bins per Measurement
     Interval for Inter-Frame Delay Variation measurements.

     The minimum number of measurement bins to be supported is
     2. The desired number of measurements bins to be supported is
     10.";
}

leaf ifdv-selection-offset {
  type yang:gauge32 {
    range "1..100";
  }
  default 1;

  description
    "This object specifies the selection offset for Inter-Frame Delay
     Variation measurements.  If this value is set to n, then the
     IFDV is calculated by taking the difference in frame delay
     between frame F and frame (F+n).";
```


Copyright© 2022 Ciena® Corporation

```
      reference
      "MEF-SOAM-PM-MIB.mefSoamDmCfgInterFrameDelayVariationSelectionOffset";
    }

    leaf bins-per-fdr-interval {
      type uint32;
      default 2;

      description
        "This object specifies the number of measurement bins per Measurement
         Interval for Frame Delay Range measurements.";
      reference
        "[MEF SOAM PM IA] R31, D15, R32, D16
         MEF-SOAM-PM-MIB.mefSoamDmCfgNumMeasBinsPerFrameDelayRangeInterval";
    }

    leaf session-type {
      type enumeration {
        enum proactive {
          description
            "The current session is 'proactive'";
        }
        enum on-demand {
          description
            "The current session is 'on-demand'";
        }
      }
      mandatory true;

      description
        "This object indicates whether the current session is defined to be
         'proactive' or 'on-demand'.";
      reference
        "[MEF SOAM IA] R3";
    }

    list thresholds {
      key id;
      description
      "This list contains the Delay Measurement threshold configuration values
         for DM Performance Monitoring.

         The main purpose of the threshold configuration list is to
         configure threshold alarm notifications indicating that a
         specific performance metric is not being met.";

      leaf id {
        type uint32 {
          range "1..2147483647";
        }

        description
          "The index of the threshold number for the specific DM threshold
           entry.

           An index value of '1' is to be supported. Other index values
           can be supported.";
      }

      leaf enabled-thresholds {
```


Copyright© 2022 Ciena® Corporation

```
    type bits {
      bit measured-frame-delay-two-way;
      bit max-frame-delay-two-way;
      bit average-frame-delay-two-way;
      bit measured-inter-frame-delay-variation-two-way;
      bit max-inter-frame-delay-variation-two-way;
      bit average-inter-frame-delay-variation-two-way;
      bit max-frame-delay-range-two-way;
      bit average-frame-delay-range-two-way;
      bit measured-frame-delay-forward;
      bit max-frame-delay-forward;
      bit average-frame-delay-forward;
      bit measured-inter-frame-delay-variation-forward;
      bit max-inter-frame-delay-variation-forward;
      bit average-inter-frame-delay-variation-forward;
      bit max-frame-delay-range-forward;
      bit average-frame-delay-range-forward;
      bit measured-frame-delay-backward;
      bit max-frame-delay-backward;
      bit average-frame-delay-backward;
      bit measured-inter-frame-delay-variation-backward;
      bit max-inter-frame-delay-variation-backward;
      bit average-inter-frame-delay-variation-backward;
      bit max-frame-delay-range-backward;
      bit average-frame-delay-range-backward;
    }
    default " ";

    description
      "A vector of bits that indicates the type of SOAM DM threshold
       notifications that are enabled.

       A present bit enables the specific SOAM DM threshold
       notification and when the specific counter is enabled and the
       threshold is crossed a notification is generated.

       A not present bit disables the specific SOAM DM threshold
       notification.

       If a particular SOAM DM threshold is not supported the BIT
       value is not present.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamDmThresholdCfgEnable";
  }

  leaf measured-frame-delay-two-way {
    type yang:gauge32;
    units microseconds;
    default 2147483647;

    description
      "This object is used to set the measurement two-way delay threshold
       value that will be used to determine if a threshold
       notification is generated.";
    reference
      "MEF-SOAM-PM-MIB." +
      "mefSoamDmThresholdCfgMeasuredFrameDelayTwoWayThreshold";
  }

  leaf max-frame-delay-two-way {
```


Copyright© 2022 Ciena® Corporation

```
    type yang:gauge32;
    units microseconds;
    default 2147483647;

    description
     "This object is used to set the maximum two-way delay threshold value
        that will be used to determine if a threshold notification is
        generated.";
    reference
    "MEF-SOAM-PM-MIB.mefSoamDmThresholdCfgMaxFrameDelayTwoWayThreshold";
  }

  leaf average-frame-delay-two-way {
    type yang:gauge32;
    units microseconds;
    default 2147483647;

    description
     "This object is used to set the average two-way delay threshold value
        that will be used to determine if a threshold notification is
        generated.";
    reference
    "MEF-SOAM-PM-MIB.mefSoamDmThresholdCfgAvgFrameDelayTwoWayThreshold";
  }

  leaf measured-inter-frame-delay-variation-two-way {
    type yang:gauge32;
    units microseconds;
    default 2147483647;

    description
      "This object is used to set the measurement two-way IFDV threshold
      value that will be used to determine if a threshold notification is
        generated.";
    reference
     "MEF-SOAM-PM-MIB.mefSoamDmThresholdCfgMeasuredIfdvTwoWayThreshold";
  }

  leaf max-inter-frame-delay-variation-two-way {
    type yang:gauge32;
    units microseconds;
    default 2147483647;

    description
     "This object is used to set the maximum two-way IFDV threshold value
        that will be used to determine if a threshold notification is
        generated.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamDmThresholdCfgMaxIfdvTwoWayThreshold";
  }

  leaf average-inter-frame-delay-variation-two-way {
    type yang:gauge32;
    units microseconds;
    default 2147483647;

    description
     "This object is used to set the average two-way IFDV threshold value
        that will be used to determine if a threshold notification is
        generated.";
```


Copyright© 2022 Ciena® Corporation

```
      reference
        "MEF-SOAM-PM-MIB.mefSoamDmThresholdCfgAvgIfdvTwoWayThreshold";
    }
    leaf max-frame-delay-range-two-way {
      type yang:gauge32;
      units microseconds;
      default 2147483647;

      description
        "This object is used to set the maximum two-way Frame Delay Range
         threshold value that will be used to determine if a threshold
         notification is generated.";
      reference
        "MEF-SOAM-PM-MIB." +
        "mefSoamDmThresholdCfgMaxFrameDelayRangeTwoWayThreshold";
    }
    leaf average-frame-delay-range-two-way {
      type yang:gauge32;
      units microseconds;
      default 2147483647;

      description
        "This object is used to set the average two-way Frame Delay Range
         threshold value that will be used to determine if a threshold
         notification is generated.";
      reference
        "MEF-SOAM-PM-MIB." +
        "mefSoamDmThresholdCfgAvgFrameDelayRangeTwoWayThreshold";
    }
    leaf measured-frame-delay-forward {
      type yang:gauge32;
      units microseconds;
      default 2147483647;

      description
       "This object is used to set the measurement forward delay thresh-old
          value that will be used to determine if a threshold
          notification is generated.";
      reference
        "MEF-SOAM-PM-MIB." +
        "mefSoamDmThresholdCfgMeasuredFrameDelayForwardThreshold";
    }

    leaf max-frame-delay-forward {
      type yang:gauge32;
      units microseconds;
      default 2147483647;

      description
       "This object is used to set the maximum forward delay threshold value
          that will be used to determine if a threshold notification is
          generated.";
      reference
        "MEF-SOAM-PM-MIB." +
        "mefSoamDmThresholdCfgMaxFrameDelayForwardThreshold";
    }

    leaf average-frame-delay-forward {
      type yang:gauge32;
      units microseconds;
```


Copyright© 2022 Ciena® Corporation

```
        default 2147483647;

        description
         "This object is used to set the average forward delay threshold value
            that will be used to determine if a threshold notification is
            generated.";
        reference
          "MEF-SOAM-PM-
MIB.mefSoamDmThresholdCfgAvgFrameDelayForwardThreshold";
      }

      leaf measured-inter-frame-delay-variation-forward {
        type yang:gauge32;
        units microseconds;
        default 2147483647;

        description
         "This object is used to set the measurement IFDV threshold value that
            will be used to determine if a threshold notification is
            generated.";
        reference
        "MEF-SOAM-PM-MIB.mefSoamDmThresholdCfgMeasuredIfdvForwardThreshold";
      }

      leaf max-inter-frame-delay-variation-forward {
        type yang:gauge32;
        units microseconds;
        default 2147483647;

        description
        "This object is used to set the maximum IFDV threshold value that will
           be used to determine if a threshold notification is generated.";
        reference
          "MEF-SOAM-PM-MIB.mefSoamDmThresholdCfgMaxIfdvForwardThreshold";
      }

      leaf average-inter-frame-delay-variation-forward {
        type yang:gauge32;
        units microseconds;
        default 2147483647;

        description
        "This object is used to set the average IFDV threshold value that will
           be used to determine if a threshold notification is generated.";
        reference
          "MEF-SOAM-PM-MIB.mefSoamDmThresholdCfgAvgIfdvForwardThreshold";
      }

      leaf max-frame-delay-range-forward {
        type yang:gauge32;
        units microseconds;
        default 2147483647;

        description
          "This object is used to set the maximum Frame Delay Range threshold
           value that will be used to determine if a threshold
           notification is generated.";
        reference
          "MEF-SOAM-PM-MIB." +
          "mefSoamDmThresholdCfgMaxFrameDelayRangeForwardThreshold";
```


Copyright© 2022 Ciena® Corporation

```
      }

      leaf average-frame-delay-range-forward {
        type yang:gauge32;
        units microseconds;
        default 2147483647;

        description
          "This object is used to set the average Frame Delay Range threshold
           value that will be used to determine if a threshold
           notification is generated.";
        reference
          "MEF-SOAM-PM-MIB." +
          "mefSoamDmThresholdCfgAvgFrameDelayRangeForwardThreshold";
      }

      leaf measured-frame-delay-backward {
        type yang:gauge32;
        units microseconds;
        default 2147483647;

        description
         "This object is used to set the measurement backward delay threshold
           value that will be used to determine if a threshold
           notification is generated.";
        reference
          "MEF-SOAM-PM-MIB." +
          "mefSoamDmThresholdCfgMeasuredFrameDelayBackwardThreshold";
      }

      leaf max-frame-delay-backward {
        type yang:gauge32;
        units microseconds;
        default 2147483647;

        description
        "This object is used to set the maximum backward delay threshold value
           that will be used to determine if a threshold notification is
           generated.";
        reference
          "MEF-SOAM-PM-
MIB.mefSoamDmThresholdCfgMaxFrameDelayBackwardThreshold";
      }

      leaf average-frame-delay-backward {
        type yang:gauge32;
        units microseconds;
        default 2147483647;

        description
        "This object is used to set the average backward delay threshold value
           that will be used to determine if a threshold notification is
           generated.";
        reference
          "MEF-SOAM-PM-
MIB.mefSoamDmThresholdCfgAvgFrameDelayBackwardThreshold";
      }

      leaf measured-inter-frame-delay-variation-backward {
        type yang:gauge32;
```


Copyright© 2022 Ciena® Corporation

```
        units microseconds;
        default 2147483647;

        description
         "This object is used to set the measurement backward IFDV thresh-old
           value that will be used to determine if a threshold
           notification is generated.";
        reference
          "MEF-SOAM-PM-
MIB.mefSoamDmThresholdCfgMeasuredIfdvBackwardThreshold";
      }

      leaf max-inter-frame-delay-variation-backward {
        type yang:gauge32;
        units microseconds;
        default 2147483647;
        description
         "This object is used to set the maximum backward IFDV threshold value
           that will be used to determine if a threshold notification is
           generated.";
        reference
          "MEF-SOAM-PM-MIB.mefSoamDmThresholdCfgMaxIfdvBackwardThreshold";
      }

      leaf average-inter-frame-delay-variation-backward {
        type yang:gauge32;
        units microseconds;
        default 2147483647;
        description
         "This object is used to set the average backward IFDV threshold value
           that will be used to determine if a threshold notification
           is generated.";
        reference
          "MEF-SOAM-PM-MIB.mefSoamDmThresholdCfgAvgIfdvBackwardThreshold";
      }
      leaf max-frame-delay-range-backward {
        type yang:gauge32;
        units microseconds;
        default 2147483647;

        description
          "This object is used to set the maximum backward Frame Delay Range
           threshold value that will be used to determine if a threshold
           notification is generated.";
        reference
          "MEF-SOAM-PM-MIB." +
          "mefSoamDmThresholdCfgMaxFrameDelayRangeBackwardThreshold";
      }

      leaf average-frame-delay-range-backward {
        type yang:gauge32;
        units microseconds;
        default 2147483647;

        description
          "This object is used to set the average backward Frame Delay Range
           threshold value that will be used to determine if a threshold
           notification is generated.";
        reference
          "MEF-SOAM-PM-MIB." +
```


Copyright© 2022 Ciena® Corporation

```
          "mefSoamDmThresholdCfgAvgFrameDelayRangeBackwardThreshold";
      }
    }
  }

  grouping delay-measurement-stats-group {
    description
      "This grouping includes statistics objects for a SOAM Delay Measurement
       session.";

    leaf suspect-status {
      type suspect-status-type;
      description
        "Whether the Measurement Interval has been marked as suspect.

         The object is to be set to false at the start of a measurement
         interval. It is set to true when there is a discontinuity in
         the performance measurements during the Measurement Interval.
         Conditions for a discontinuity include, but are not limited to
         the following:

          1 - The local time-of-day clock is adjusted by at least 10 seconds
         2 - The conducting of a performance measurement is halted before the
              current Measurement Interval is completed
        3 - A local test, failure, or reconfiguration that disrupts service";
    }

    leaf frame-delay-two-way-min {
      type yang:gauge32;
      units microseconds;

      description
      "This object contains the minimum two-way frame delay calculated by this
         MEP for this Measurement Interval.

         This object is undefined if measurement-type is dm1-transmitted
         or dm1-received.";
    }

    leaf frame-delay-two-way-max {
      type yang:gauge32;
      units microseconds;

      description
      "This object contains the maximum two-way frame delay calculated by this
         MEP for this Measurement Interval.

         This object is undefined if measurement-type is dm1DmTx or
         dm1-received.";
    }

    leaf frame-delay-two-way-average {
      type yang:gauge32;
      units microseconds;

      description
      "This object contains the average two-way frame delay calculated by this
         MEP for this Measurement Interval.

         This object is undefined if measurement-type is dm1-transmitted or
```


Copyright© 2022 Ciena® Corporation

```
      dm1-received.";
  }
  leaf frame-delay-forward-min {
    type yang:gauge32;
    units microseconds;

    description
      "This object contains the minimum one-way frame delay in the forward
      direction calculated by this MEP for this Measurement
      Interval. The value of this object may not be accurate in the
      absence of sufficiently precise clock synchronization.

      This object is undefined if measurement-type is dm1-transmitted.";
  }

  leaf frame-delay-forward-max {
    type yang:gauge32;
    units microseconds;

    description
      "This object contains the maximum one-way frame delay in the forward
       direction calculated by this MEP for this Measurement
       Interval. The value of this object may not be accurate in the
       absence of sufficiently precise clock synchronization.

       This object is undefined if measurement-type is dm1-transmitted.";
  }

  leaf frame-delay-forward-average {
    type yang:gauge32;
    units microseconds;

    description
      "This object contains the average one-way frame delay in the forward
       direction calculated by this MEP for this Measurement
       Interval. The value of this object may not be accurate in the
       absence of sufficiently precise clock synchronization.

       This object is undefined if measurement-type is dm1-transmitted.";
  }

  leaf frame-delay-backward-min {
    type yang:gauge32;
    units microseconds;

    description
      "This object contains the minimum one-way frame delay in the backward
       direction calculated by this MEP for this Measurement
       Interval. The value of this object may not be accurate in the
       absence of sufficiently precise clock synchronization.

       This object is undefined if measurement-type is dm1-transmitted
       or dm1-received.";
  }

  leaf frame-delay-backward-max {
    type yang:gauge32;
    units microseconds;

    description
```


Copyright© 2022 Ciena® Corporation

```
      "This object contains the maximum one-way frame delay in the backward
       direction calculated by this MEP for this Measurement
       Interval. The value of this object may not be accurate in the
       absence of sufficiently precise clock synchronization.

       This object is undefined if measurement-type is dm1-transmitted or
       dm1-received.";
    }

    leaf frame-delay-backward-average {
      type yang:gauge32;
      units microseconds;

      description
        "This object contains the average one-way frame delay in the backward
         direction calculated by this MEP for this Measurement
         Interval. The value of this object may not be accurate in the
         absence of sufficiently precise clock synchronization.

         This object is undefined if measurement-type is dm1-transmitted or
         dm1-received.";
    }

    leaf inter-frame-delay-variation-forward-min {
      type yang:gauge32;
      units microseconds;
      description
       "This object contains the minimum one-way inter-frame delay interval in
         the forward direction calculated by this MEP for this
         Measurement Interval.

         The value of this object is undefined when measurement-type is
         dm1-transmitted.";
    }

    leaf inter-frame-delay-variation-forward-max {
      type yang:gauge32;
      units microseconds;

      description
       "This object contains the maximum one-way inter-frame delay interval in
         the forward direction calculated by this MEP for this
         Measurement Interval.

         The value of this object is undefined when measurement-type is
         dm1-transmitted.";
    }

    leaf inter-frame-delay-variation-forward-average {
      type yang:gauge32;
      units microseconds;

      description
       "This object contains the average one-way inter-frame delay interval in
         the forward direction calculated by this MEP for this
         Measurement Interval.

         The value of this object is undefined when measurement-type is
         dm1-transmitted.";
    }
```


Copyright© 2022 Ciena® Corporation

```
leaf inter-frame-delay-variation-backward-min {
  type yang:gauge32;
  units microseconds;

  description
   "This object contains the minimum one-way inter-frame delay interval in
      the backward direction calculated by this MEP for this
      Measurement Interval.

    The value of this object is undefined when measurement-type is
    dm1-transmitted or dm1-received.";
}

leaf inter-frame-delay-variation-backward-max {
  type yang:gauge32;
  units microseconds;

  description
   "This object contains the maximum one-way inter-frame delay interval in
      the backward direction calculated by this MEP for this
      Measurement Interval.

      The value of this object is undefined when measurement-type is
      dm1-transmitted or dm1-received.";
}

leaf inter-frame-delay-variation-backward-average {
  type yang:gauge32;
  units microseconds;

  description
   "This object contains the average one-way inter-frame delay interval in
      the backward direction calculated by this MEP for this
      Measurement Interval.

      The value of this object is undefined when measurement-type is
      dm1-transmitted or dm1-received.";
}

leaf inter-frame-delay-variation-two-way-min {
  type yang:gauge32;
  units microseconds;

  description
    "This object contains the minimum two-way inter-frame delay interval
     calculated by this MEP for this Measurement Interval.

      The value of this object is undefined when measurement-type is
      dm1-transmitted or dm1-received.";
}

leaf inter-frame-delay-variation-two-way-max {
  type yang:gauge32;
  units microseconds;

  description
    "This object contains the maximum two-way inter-frame delay interval
     calculated by this MEP for this Measurement Interval.
```


Copyright© 2022 Ciena® Corporation

```
      The value of this object is undefined when measurement-type is
      dm1-transmitted or dm1-received.";
  }

  leaf inter-frame-delay-variation-two-way-average {
    type yang:gauge32;
    units microseconds;

    description
      "This object contains the average two-way inter-frame delay interval
       calculated by this MEP for this Measurement Interval.

       The value of this object is undefined when measurement-type is
       dm1-transmitted or dm1-received.";
  }

  leaf frame-delay-range-forward-max {
    type yang:gauge32;
    units microseconds;

    description
      "This object contains the maximum one-way Frame Delay Range in the
       forward direction calculated by this MEP for this Measurement
       Interval.

       The value of this object is undefined when measurement-type is
       dm1-transmitted.";
  }

  leaf frame-delay-range-forward-average {
    type yang:gauge32;
    units microseconds;

    description
      "This object contains the average one-way Frame Delay Range in the
       forward direction calculated by this MEP for this Measurement
       Interval.

       The value of this object is undefined when measurement-type is
       dm1-transmitted.";
  }

  leaf frame-delay-range-backward-max {
    type yang:gauge32;
    units microseconds;

    description
      "This object contains the maximum one-way Frame Delay Range in the
       backward direction calculated by this MEP for this Measurement
       Interval.

       The value of this object is undefined when measurement-type is
       dm1-transmitted or dm1-received.";
  }

  leaf frame-delay-range-backward-average {
    type yang:gauge32;
    units microseconds;

    description
```


Copyright© 2022 Ciena® Corporation

```
        "This object contains the average one-way Frame Delay Range in the
         backward direction calculated by this MEP for this Measurement
         Interval.

         The value of this object is undefined when measurement-type is
         dm1-transmitted or dm1-received.";
    }

    leaf frame-delay-range-two-way-max {
      type yang:gauge32;
      units microseconds;

      description
       "This object contains the maximum two-way Frame Delay Range calculated
         by this MEP for this Measurement Interval.

         The value of this object is undefined when measurement-type is
         dm1-transmitted or dm1-received.";
    }

    leaf frame-delay-range-two-way-average {
      type yang:gauge32;
      units microseconds;

      description
       "This object contains the average two-way Frame Delay Range calculated
         by this MEP for this Measurement Interval.

         The value of this object is undefined when measurement-type is
         dm1-transmitted or dm1-received.";
    }
    leaf soam-pdus-sent {
      type yang:gauge32;

      description
       "This object contains the count of the number of SOAM PDUs sent during
         this Measurement Interval.

         This object applies when measurement-type is dmm or
         dm1-transmitted and is undefined if measurement-type is
         dm1-received. It indicates the number of DMM or 1DM SOAM frames
         transmitted.";
    }

    leaf soam-pdus-received {
      type yang:gauge32;
      description
       "This object contains the count of the number of SOAM PDUs received in
         this Measurement Interval.

         This object indicates the number of DMR and 1DM SOAM frames
         received. This object applies when measurement-type is dmm or
         dm1-received and is undefined if measurement-type is
         dm1-transmitted.";
    }
  }

  grouping delay-measurement-bins-content-group {
    description
      "This grouping contains result measurement bin objects for a SOAM Delay
```


Copyright© 2022 Ciena® Corporation

```
     Measurement session.";

  leaf type {
    type measurement-bin-type;

    description
      "This object specifies whether the bin number is for Frame De-lay,
       Inter-Frame Delay Variation or Frame Delay Range.";
  }

  leaf number {
    type uint32;

    description
     "This object specifies the bin number for the configured boundary.  The
       first bin has bin number 1.";
  }

  leaf lower-bound {
    type yang:gauge32;
    units microseconds;

    description
      "This object specifies the lower boundary for a measurement bin. The
       upper boundary is defined by the next bin value or infinite for
       the last bin defined.  The measurement boundary for each
       measurement bin is to be larger than the measurement boundary
       of the preceding measurement bin. By default, the next bin is
       set to 5000us larger than the lower bin boundary.

       The values in a bin boundary object represents the time range
       used to segregate delay data into the appropriate statistical
       data bin. For five bins with default values, each bin has the
       following time range:

        bin 1 =     0, range is 0us
        bin 2 =  5000, range is 5,000us
        bin 3 = 10000, range is 10,000us
        bin 4 = 15000, range is 15,000us
        bin 5 = 20000, range is 20,000us

      The first bin boundary (number set to 1) always contains the
      value of 0. Attempting to write a non-zero value to this bin
      will result in an error.";
  }

  leaf counter {
    type yang:gauge32;
    config false;

    description
     "This object contains the count of the number of completed measurements
       initiated in this Measurement Interval whose value falls within
       the range specified for this bin (that is, greater than or
       equal to the measurement boundary for the bin, and (unless the
       bin is the last bin) less than the measurement boundary for the
       following bin.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamDmCurrentStatsBinsCounter";
  }
```


Copyright© 2022 Ciena® Corporation

```
}

grouping delay-measurement-bins-group {
  description
    "This grouping contains the top-level structure for the three types of
     measurements (frame delay, inter frame delay variation, and frame
     delay range)";

  container bins {
    description
      "This container defines the bins content for FD, IFDR, and FDR
       sessions.";

    list frame-delay {
      key "type number";
      uses delay-measurement-bins-content-group;

      description
        "Data definitions related to frame delay bins content.";
    }

    list inter-frame-delay-variation {
      key "type number";
      uses delay-measurement-bins-content-group;

      description
        "Data definitions related to inter frame delay bins content.";
    }

    list frame-delay-range {
      key "type number";
      uses delay-measurement-bins-content-group;

      description
        "Data definitions related to frame delay range bins content.";
    }
  }
}

//
// Augments into CFM related to MEF SOAM PM
//

augment "/cfm:maintenance-domain/cfm:maintenance-association" +
        "/cfm:maintenance-association-end-point" {
  description
    "This set of data definitions extends the MEG End Point (as described in
     Y.1731) or Maintenance End Point (as described in 802.1q) defined
     in the MEF CFM module (MEF-CFM), specifically with regards to the
     role of the MEP in PM tests.";

  leaf loss-single-ended-responder {
    type boolean;
    default true;

    description
     "This object specifies whether the Loss Measurement (LMM) single-ended
       Responder is enabled.

       The value 'true' indicates the single-ended Loss Measurement
```


Copyright© 2022 Ciena® Corporation

```
        Responder is enabled and if a LMM message is received a LMR
        will be sent in re-ply.

        The value 'false' indicates the single-ended Loss Measurement
        Responder is disabled. If a LMM message is received no
        response will be sent and the message will be discarded.

        This object needs to be persistent upon reboot or restart of a
        device.

        A MEP can be both a single-ended Responder and Controller
        simultaneously.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamPmMepLmSingleEndedResponder";
  }

  leaf synthetic-loss-single-ended-responder {
    type boolean;
    default true;

    description
      "This object specifies whether the Synthetic Loss Measurement (SLM)
       single-ended Responder is enabled.

       The value 'true' indicates the single-ended SLM Responder is
       enabled and if a SLM message is received a SLR will be sent in
       reply.

       The value 'false' indicates the single-ended SLM Responder is
       disabled.  If a SLM message is received no response
       will be sent and the message will be discarded.

       This object needs to be persistent upon reboot or restart of a
       device.

       A MEP can be both a single-ended Responder and Controller
       simultaneously.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamPmMepSlmSingleEndedResponsder";
  }

  leaf delay-single-ended-responder {
    type boolean;
    default true;

    description
     "This object specifies whether the Delay Measurement (DMM) single ended
        Responder is enabled.

        The value 'true' indicates the single-ended Delay Measurement
        Responder is enabled and if a DMM message is received a DMR
        will be sent in re-ply.

        The value 'false' indicates the single-ended Delay Measurement
        Responder is disabled. If a DMM message is received no
        response will be sent and the message will be discarded.

        This object needs to be persistent upon reboot or restart of a
        device.
```


Copyright© 2022 Ciena® Corporation

```
          A MEP can be both a single-ended Responder and Controller
          simultaneously.";
      reference
        "MEF-SOAM-PM-MIB.mefSoamPmMepDmSingleEndedResponder";
    }
  }

  augment "/cfm:maintenance-domain/cfm:maintenance-association" +
          "/cfm:maintenance-association-end-point" {
    description
      "This set of data definitions extends the MEG End Point (as described
       Y.1731) or Maintenance End Point (as described in 802.1q) defined
       in the MEF CFM module (MEF-CFM), specifically with regards to
       Loss Measurement.";

    container loss-measurements {
      description
        "This container contains a collection of data definitions re-lated to
         Loss Measurements.";

      list loss-measurement {
        key session-id;
        config false;
        description
          "A list of Loss Measurement PM Sessions where each in-stance is
uniquely
           identified by an id attribute.";

        leaf session-id {
          type uint32;

          description
            "This object uniquely identifies a scheduled measurement. It is
             automatically generated by the server on creation of a new
             measurement.";
        }

        leaf session-status {
          type session-status-type;
          config false;

          description
          "This object indicates the current status of the LM session. A value
             of 'active' indicates the current LM session is active,
             i.e. the current time lies between the start time and the
             stop time, and enabled is true. A value of 'not-active'
             indicates the current LM session is not active, i.e. it has
             not started yet, has stopped upon reaching the stop time,
             or is disabled.";
        }

        uses loss-measurement-configuration-group;
        uses remote-mep-group;
        uses measurement-timing-group;

        leaf measured-forward-flr {
          type yang:gauge32 {
            range "0..100000";
          }
          units milli-percent;
```


Copyright© 2022 Ciena® Corporation

```
    description
     "This object contains the Frame Loss Ratio in the forward direction
        calculated by this MEP from the last received SOAM PDU. The
        FLR value is a ratio that is expressed as a percent with a
        value of 0 (ratio 0.00) through 100000 (ratio 1.00).

        Units are in milli-percent, where 1 indicates 0.001
        per-cent.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmMeasuredStatsForwardFlr";
  }

  leaf measured-backward-flr {
    type yang:gauge32 {
      range "0..100000";
    }
    units milli-percent;

    description
    "This object contains the Frame Loss Ratio in the backward direction
        calculated by this MEP from the last received SOAM PDU. The
        FLR value is a ratio that is expressed as a percent with a
        value of 0 (ratio 0.00) through 100000 (ratio 1.00).

        Units are in milli-percent, where 1 indicates 0.001
        percent.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmMeasuredStatsBackwardFlr";
  }

  leaf measured-availability-forward-status {
    type availability-type;

    description
     "This object indicates the availability status (the outcome of the
        last availability indicator) in the forward direction based upon
        the last received SOAM PDU.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmMeasuredStatsAvailForwardStatus";
  }

  leaf measured-availability-backward-status {
    type availability-type;

    description
     "This object indicates the availability status (the outcome of the
        last availability indicator) in the backward direction based
        upon the last received SOAM PDU.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmMeasuredStatsAvailBackwardStatus";
  }

  leaf measured-forward-last-transition-time {
    type yang:date-and-time;
    description
      "This object indicates the time of the last transition between
       available and unavailable in the forward direction.

       If there have been no transitions since the Loss
```


Copyright© 2022 Ciena® Corporation

```
       Measurement Session was started, this is set to 0.";
    reference
      "MEF-SOAM-PM-MIB." +
      "mefSoamLmMeasuredStatsAvailForwardLastTransitionTime";
  }

  leaf measured-backward-last-transition-time {
    type yang:date-and-time;
    description
      "This object indicates the time of the last transition between
       available and unavailable in the backward direction.

       If there have been no transitions since the Loss Measurement
       Session was started, this is set to 0.";
    reference
      "MEF-SOAM-PM-MIB." +
      "mefSoamLmMeasuredStatsAvailBackwardLastTransitionTime";
  }
  container current-availability-stats {
    description
      "This container contains the current results for a SOAM Loss
     Measurement session for availability statistics gathered during the
       interval indicated by availability-measurement-interval.";

    leaf id {
      type uint32;
      description
       "The index for the current availability Measurement Interval for
        this PM session. This value will become the value for id of the
         history-availability-stats list once the Measurement
         Interval is completed.  The duration of the Measurement
         Interval is specified by availability-measurement-interval.";
    }

    leaf start-time {
      type yang:date-and-time;
      mandatory true;

      description
        "The time that the current Measurement Interval started.";
    }

    leaf elapsed-time {
      type uint32 {
        range "0..2147483647";
      }
      mandatory true;

      description
       "The time that the current Measurement Interval has been running,
          in 0.01 seconds.";
    }

    uses loss-availability-stats-group;
  }

  container current-measurement-stats {
    description
      "This container contains the results for the current Measurement
       Interval in a SOAM Loss Measurement session gathered during
```


Copyright© 2022 Ciena® Corporation

```
        the interval indicated by measurement-interval.";

      leaf id {
        type uint32;

        description
          "The index for the current Measurement Interval for this PM
           session. This value will become the value for id of the
           history-measurement-stats list once the Measurement
           Interval is completed.";
      }

      leaf start-time {
        type yang:date-and-time;
        mandatory true;

        description
          "The time that the current Measurement Interval started.";
      }

      leaf elapsed-time {
        type uint32 {
          range "0..2147483647";
        }
        mandatory true;

        description
         "The time that the current Measurement Interval has been running,
            in 0.01 seconds.";
      }

      uses loss-measurement-stats-group;
    }

    list history-availability-stats {
      key id;
      description
      "This list contains the results for availability history Measurement
         Intervals in a SOAM Loss Measurement session.";

      leaf id {
        type uint32;
        description
         "The index for the availability Measurement Interval within this
            PM session.";
      }

      leaf end-time {
        type yang:date-and-time;
        mandatory true;

        description
          "The time that the Measurement Interval ended.";
      }

      leaf elapsed-time {
        type uint32 {
          range "0..2147483647";
        }
        mandatory true;
```


Copyright© 2022 Ciena® Corporation

```
            description
              "The length of time that the Measurement Interval ran for,
               in 0.01 seconds.";
            }

            uses loss-availability-stats-group;
          }

          list history-measurement-stats {
            key id;
            description
              "This list contains the results for history Measurement
               Intervals in a SOAM Loss Measurement session.";

            leaf id {
              type uint32;

              description
                "The index for the Measurement Interval within this
                 PM session.";
            }

            leaf end-time {
              type yang:date-and-time;
              mandatory true;

              description
                "The time that the Measurement Interval ended.";
            }

            leaf elapsed-time {
              type uint32 {
                range "0..2147483647";
              }
              mandatory true;

              description
                "The length of time that the Measurement Interval ran for,
                in 0.01 seconds.";
            }

            uses loss-measurement-stats-group;
          }
        }
      }
    }

    augment "/cfm:maintenance-domain/cfm:maintenance-association" +
            "/cfm:maintenance-association-end-point" {
      description
        "This set of data definitions extends the MEG End Point (as described in
             Y.1731) or Maintenance End Point (as described in 802.1q)
             defined in the MEF CFM module (MEF-CFM), specifically with
             regards to Delay Measurement.";

      container delay-measurements {
        description
          "This container contains a collection of data definitions related to
           Delay Measurements.";
```


Copyright© 2022 Ciena® Corporation

```
list delay-measurement {
  key id;
  config false;
  description
    "A list of Delay Measurement PM Sessions where each instance is
     uniquely identified by an id attribute.";

  leaf id {
    type string;

    description
      "This object uniquely identifies a scheduled measurement. It is
       automatically generated by the server on creation of a new
       measurement.";
  }

  leaf session-status {
    type session-status-type;
    config false;

    description
    "This object indicates the current status of the DM session. A value
       of 'active' indicates the current DM session is active,
       i.e. the current time lies between the start time and the
       stop time, and enabled is true. A value of 'not-active'
       indicates the current DM session is not active, i.e. it has
       not started yet, has stopped upon reaching the stop time,
       or is disabled.";
  }

  uses delay-measurement-configuration-group;
  uses remote-mep-group;
  uses measurement-timing-group;

  leaf source-mac-address {
    type yang:mac-address;

    description
     "The Source MAC Address Field of the received SOAM DM session PDUs.

       If measurement-type is dm1-received this object indicates
       the source address of the dm1-transmitted DM session.

       This object is only valid for measurement-type set to
       dm1-received. It is ignored for measurement-type set to dmm
       or dm1-transmitted.";
  }

  leaf frame-delay-two-way {
    type yang:gauge32;
    units microseconds;
    description
    "This object contains the two-way frame delay calculated by this MEP
       from the last received SOAM PDU.

       This object is undefined is measurement-type is
       dm1-transmitted or dm1-received.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamDmMeasuredStatsFrameDelayTwoWay";
```


Copyright© 2022 Ciena® Corporation

```
}

leaf frame-delay-forward {
  type yang:gauge32;
  units microseconds;
  description
    "This object contains the frame delay in the forward direction
    calculated by this MEP from the last received SOAM PDU. The value
     of this object may not be accurate in the absence of
     sufficiently precise clock synchronization.

     This object is undefined is measurement-type is
     dm1-transmitted.";
  reference
    "MEF-SOAM-PM-MIB.mefSoamDmMeasuredStatsFrameDelayForward";
}

leaf frame-delay-backward {
  type yang:gauge32;
  units microseconds;

  description
    "This object contains the frame delay in the backward direction
     calculated by this MEP from the last received SOAM PDU. The
     value of this object may not be accurate in the absence of
     sufficiently precise clock synchronization.

     This object is undefined is measurement-type is
     dm1-transmitted or dm1-received.";
  reference
    "MEF-SOAM-PM-MIB.mefSoamDmMeasuredStatsFrameDelayBackward";
}

leaf inter-frame-delay-variation-two-way {
  type yang:gauge32;
  units microseconds;

  description
    "This object contains the last two-way inter-frame delay interval
     calculated by this MEP.

     The value of this object is undefined when measurement-type
     is dm1-transmitted or dm1-received.";
  reference
    "MEF-SOAM-PM-MIB.mefSoamDmMeasuredStatsIfdvTwoWay";
}

leaf inter-frame-delay-variation-forward {
  type yang:gauge32;
  units microseconds;

  description
  "This object contains the last one-way inter-frame delay interval in
     the forward direction calculated by this MEP.

     The value of this object is undefined when measurement-type
     is dm1-transmitted.";
  reference
    "MEF-SOAM-PM-MIB.mefSoamDmMeasuredStatsIfdvForward";
}
```


Copyright© 2022 Ciena® Corporation

```
leaf inter-frame-delay-variation-backward {
  type yang:gauge32;
  units microseconds;

  description
  "This object contains the last one-way inter-frame delay interval in
     the backward direction calculated by this MEP.

     The value of this object is undefined when measurement-type
     is dm1-transmitted or dm1-received.";
  reference
    "MEF-SOAM-PM-MIB.mefSoamDmMeasuredStatsIfdvBackward";
}

container current-stats {
  description
    "This container contains the results for the current Measurement
     Interval in a SOAM Delay Measurement session gathered
     during the interval indicated by measurement-interval.";

 leaf start-time {
    type yang:date-and-time;
    mandatory true;

    description
      "The time that the current Measurement Interval started.";
  }

  leaf elapsed-time {
    type uint32 {
      range "0..2147483647";
    }
    mandatory true;

    description
     "The time that the current Measurement Interval has been running,
        in 0.01 seconds.";
  }

  uses delay-measurement-stats-group;
  uses delay-measurement-bins-group;
}

list history-stats {
  key id;
  description
  "This list contains the results for history Measurement Intervals in
     a SOAM Delay Measurement session.";

  leaf id {
    type uint32;

    description
      "The identifier for the Measurement Interval within this
       PM session.";
  }

  leaf end-time {
    type yang:date-and-time;
```


Copyright© 2022 Ciena® Corporation

```
            mandatory true;

            description
              "The time that the Measurement Interval ended.";
          }

          leaf elapsed-time {
            type uint32 {
              range "0..2147483647";
            }
            mandatory true;

            description
            "The length of time that the Measurement Interval ran for, in 0.01
               seconds.";
          }

          uses delay-measurement-stats-group;
          uses delay-measurement-bins-group;
        }
      }
    }
  }

  //
  // RPCs related to MEF SOAM PM
  //

  rpc create-loss-measurement {
    description
      "Schedule a one-way and two-way on-demand and proactive performance
       monitoring loss measurements on a specific MEP.

       A list entry associated with the newly created session will be
       created in the loss-measurements container and the assigned
       session identifier will be returned in the output parameter.";

    input {
      uses loss-measurement-configuration-group;
      uses remote-mep-group;
      uses measurement-timing-group;
    }

    output {
      leaf session-id {
        type uint32;
        mandatory true;

        description
          "The session identifier of the newly created loss measurement
           session.";
      }
    }
  }

  rpc abort-loss-measurement {
    description
      "Abort a currently running or scheduled single-ended on-demand ETH-LM
       function. This operation is not applicable to dual-ended
       proactive ETH-LM.";
```


Copyright© 2022 Ciena® Corporation

```
    input {
      leaf maintenance-domain-id {
        type string;

        description
          "The ID of a Maintenance Domain of the MEP on which the measurement
           session should be aborted";
      }

      leaf maintenance-association-id {
        type string;

        description
          "The ID of a Maintenance Association of the MEP on which the
measurement
           session should be aborted";
      }


      leaf mep-id {
        type cfm:mep-id-type;
        mandatory true;

        description
         "The ID of the MEP on which a measurement session should be aborted";
      }

      leaf session-id {
        type uint32;
        mandatory true;

        description
          "The session ID of the measurement session to be aborted.";
      }
    }
  }

  rpc clear-loss-history-stats {
    description
     "Clear the Loss Measurement history list (history-availability-stats and
      history-measurement-stats) for a specific session. All instance
      data is deleted.";

    input {
      leaf maintenance-domain-id {
        type string;

        description
          "The ID of a Maintenance Domain of the MEP on which the loss
           measurement history should be cleared";
      }

      leaf maintenance-association-id {
        type string;

        description
          "The ID of a Maintenance Association of the MEP on which the loss
           measurement history should be cleared";
      }
```


Copyright© 2022 Ciena® Corporation

```
      leaf mep-id {
        type cfm:mep-id-type;
        mandatory true;

        description
          "The ID of the MEP on which the loss measurement history should be
           cleared.";
      }

      leaf session-id {
        type uint32;
        mandatory true;

        description
          "The session ID of the loss measurement session of which the loss
           measurement history should be cleared";
      }
    }
  }

  rpc create-delay-measurement {
    description
      "Schedule one-way and two-way on-demand and proactive performance
       monitoring frame delay measurements on a specific MEP.

       A list entry associated with the newly created session will be
       created in the delay-measurements container and the assigned
       session identifier will be returned in the output parameter.";

    input {
      uses delay-measurement-configuration-group;
      uses remote-mep-group;
      uses measurement-timing-group;
    }

    output {
      leaf session-id {
        type uint32;
        mandatory true;

        description
          "The session identifier of the newly created delay measurement
           session.";
      }
    }
  }

  rpc abort-delay-measurement {
    description
      "Abort a pre-scheduled instances of performance monitoring";

    input {
      leaf maintenance-domain-id {
        type string;

        description
          "The ID of a Maintenance Domain of the MEP on which a measurement
           session should be aborted";
      }
```


Copyright© 2022 Ciena® Corporation

```
      leaf maintenance-association-id {
        type string;

        description
        "The ID of a Maintenance Association of the MEP on which a measurement
           session should be aborted";
      }
      leaf mep-id {
        type cfm:mep-id-type;
        mandatory true;

        description
         "The ID of the MEP on which a measurement session should be aborted";
      }

      leaf session-id {
        type uint32;
        mandatory true;

        description
          "The session ID of the measurement session to be aborted.";
      }
    }
  }

  rpc clear-delay-history-stats {
    description
     "Clear the Delay Measurement history list (history-stats) for a specific
      session. All instance data is deleted.";

    input {
      leaf maintenance-domain-id {
        type string;

        description
          "The ID of a Maintenance Domain of the MEP on which the loss
           measurement history should be cleared";
      }

      leaf maintenance-association-id {
        type string;

        description
          "The ID of a Maintenance Association of the MEP on which the
           loss measurement history should be cleared";
      }
      leaf mep-id {
        type cfm:mep-id-type;
        mandatory true;

        description
          "The ID of the MEP on which the loss measurement history should be
           cleared.";
      }

      leaf session-id {
        type uint32;
        mandatory true;
```


Copyright© 2022 Ciena® Corporation

```
        description
          "The session ID of the loss measurement session of which the loss
           measurement history should be cleared";
      }
    }
  }

  //
  // Notifications related to MEF SOAM PM
  //

  notification availability-change-alarm {
    description
      "An availability-change-alarm notification is sent when the state of
       measured-availability-forward-status or
       measured-availability-backward-status associated with a specific
       loss-measurement session changes.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamAvailabilityChangeAlarm";

    leaf maintenance-domain-id {
      type string;

      description
        "The ID of a Maintenance Domain of the MEP on which the availability
         change alarm is present";
    }

    leaf maintenance-association-id {
      type string;

      description
      "The ID of a Maintenance Association of the MEP on which the availability
         change alarm is present";
    }

    leaf mep-id {
      type cfm:mep-id-type;
      mandatory true;

      description
       "A reference to the specific MEP on which the availability change alarm
         is present";
    }

    leaf session-id {
      type uint32;
      mandatory true;

      description
       "A reference to the loss-measurement session in which the availability
         change alarm occurred.";
    }

    leaf measured-availability-forward-status {
      type availability-type;

      description
        "The current value of measured-availability-forward-status of the
         referenced loss-measurement session.";
```


Copyright© 2022 Ciena® Corporation

```
    }

    leaf measured-availability-backward-status {
      type availability-type;

      description
        "The current value of measured-availability-backward-status of the
         referenced loss-measurement session.";
    }

    leaf measured-forward-last-transition-time {
      type yang:date-and-time;

      description
        "The current value of measured-forward-last-transition-time of the
         referenced loss-measurement session.";
    }

    leaf measured-backward-last-transition-time {
      type yang:date-and-time;

      description
        "The current value of measured-backward-last-transition-time of the
         referenced loss-measurement session.";
    }

    leaf forward-available {
      type yang:gauge32;

      description
        "The current value of forward-available of the referenced
         loss-measurement session.";
    }

    leaf backward-available {
      type yang:gauge32;

      description
        "The current value of backward-available of the referenced
         loss-measurement session.";
    }

    leaf forward-unavailable {
      type yang:gauge32;

      description
        "The current value of forward-unavailable of the referenced
         loss-measurement session.";
    }

    leaf backward-unavailable {
      type yang:gauge32;

      description
        "The current value of backward-unavailable of the referenced
         loss-measurement session.";
    }
  }

  notification loss-session-start-stop-alarm {
```


Copyright© 2022 Ciena® Corporation

```
    description
      "A loss-session-start-stop-alarm notification is sent when the state of
       session-status changes for a specific loss-measurement session.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamLmSessionStartStopAlarm";

    leaf maintenance-domain-id {
      type string;

      description
        "The ID of a Maintenance Domain of the MEP on which the loss session
         state change alarm is present";
    }

    leaf maintenance-association-id {
      type string;

      description
        "The ID of a Maintenance Association of the MEP on which the loss
         session state change alarm is present";
    }

    leaf mep-id {
      type cfm:mep-id-type;
      mandatory true;

      description
        "A reference to the specific MEP on which the loss session state
         change alarm occurred.";
    }

    leaf session-id {
      type uint32;
      mandatory true;

      description
       "A reference to the loss-measurement session in which the loss session
         state change alarm occurred.";
    }

    leaf session-status {
      type session-status-type;

      description
       "The current value of session-status of the referenced loss-measurement
         session.";
    }
  }

  notification delay-session-start-stop-alarm {
    description
     "A delay-session-start-stop-alarm notification is sent when the state of
       session-status changes for a specific delay-measurement session.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamDmSessionStartStopAlarm";

    leaf maintenance-domain-id {
      type string;

      description
```


Copyright© 2022 Ciena® Corporation

```
        "The ID of a Maintenance Domain of the MEP on which the delay
         session state change alarm is present";
    }

    leaf maintenance-association-id {
      type string;

      description
        "The ID of a Maintenance Association of the MEP on which the delay
         session state change alarm is present";
    }

    leaf mep-id {
      type cfm:mep-id-type;
      mandatory true;

      description
        "A reference to the specific MEP on which the delay session state
         change alarm occurred.";
    }

    leaf session-id {
      type uint32;
      mandatory true;

      description
       "A reference to the delay-measurement session in which the delay session
          state change alarm occurred.";
    }

    leaf session-status {
      type session-status-type;
      description
      "The current value of session-status of the referenced delay-measurement
         session.";
    }
  }

  notification threshold-crossing-alarm {
    description
      "A threshold-crossing-alarm notification is sent if the following
       conditions are met for a particular type.

       For an above-alarm five conditions need to be met:
        - measurement of the parameter is enabled via enabled-thresholds
          for a LM or a DM crossing; and
        - the parameter threshold is configured in the thresholds list
          for an LM or DM session.
        - the threshold crossing type of threshold-above-alarm is
          enabled in alarm-enable; and
        - the measured value of the parameter exceeds the value
          configured in the thresholds list for an LM or DM crossing
          entry for a type of threshold-above-alarm.
        - no previous threshold-crossing-alarm notifications with type
          above-alarm have been sent relating to the same threshold and
          the same parameter, during this measurement interval.

       For a set-alarm five conditions need to be met:
        - measurement of the parameter is enabled via measurement-enable
          for a LM or a DM crossing; and
```


Copyright© 2022 Ciena® Corporation

```
        - the parameter threshold is configured in the thresholds list
          for an LM or DM session.
        - the threshold crossing type of threshold-set-clear-alarm is
          enabled in alarm-enable; and
        - the measured value of the parameter exceeds the value
          configured in the thresholds list for an LM or DM crossing
          entry for a type of threshold-set-clear-alarm.
        - no previous threshold-crossing-alarm notifications with type
          threshold-set-clear have been sent relating to the same
          threshold and the same parameter, during this measurement
          interval.

        For a clear-alarm five conditions need to be met:
         - measurement of the parameter is enabled via enabled-thresholds
           for a LM or a DM crossing; and
         - the parameter threshold is configured in the thresholds list
           for an LM or DM session.
         - the threshold crossing type of threshold-set-clear-alarm is
           enabled in alarm-enable; and
         - no previous threshold-crossing-alarm notifications with type
           threshold-set-clear have been sent relating to the same
           threshold and the same parameter, during this measurement
           interval.

        In the case of thresholds applied to a maximum or average
        measurement counter, the previous measured value is the value of
        the counter at the end of the preceding measurement interval.
        In the case of thresholds applied to the last measured value, it
        is the previous measured value.

        An agent is not to generate more than one
        threshold-crossing-alarm of a given type per LM or DM session as
        specified by alarm-interval. A 'notification-event' is the
        transmission of a single notification to a list of notification
        destinations.

        If additional threshold crossing events occur within the
        alarm-interval period, then notification generation for these
        changes are suppressed by the agent until the current alarm
        interval expires. At the end of an alarm interval period, one
        notification-event is generated if any threshold crossing events
        occurred since the start of the alarm interval period.  In such
        a case, another alarm interval period is started right away.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamPmThresholdCrossingAlarm";


    leaf maintenance-domain-id {
      type string;

      description
        "The ID of a Maintenance Domain of the MEP on which the threshold
         crossing alarm occurred";
    }

    leaf maintenance-association-id {
      type string;

      description
        "The ID of a Maintenance Association of the MEP on which the threshold
```


Copyright© 2022 Ciena® Corporation

```
      crossing alarm occurred";
  }
  leaf mep-id {
    type cfm:mep-id-type;
    mandatory true;

    description
     "A reference to the specific MEP on which the threshold crossing alarm
      occurred.";
  }

  container measurement-id {
    description
      "This container describes the measurement ID for a loss or delay
       measurement.";

    choice measurement-id {
      mandatory true;
      description
      "The measurement ID can reference either a loss or a delay measurement
         session.";

      leaf loss-measurement-session-id {
        type uint32;

        description
        "The session identifier for the loss-measurement session affected by
           the notification.";
      }

      leaf delay-measurement-session-id {
        type uint32;

        description
         "The session identifier for the delay-measurement session affected
           by the notification.";
      }
    }
  }

  leaf notification-crossing-type {
    type enumeration {
      enum above-alarm {
        description
          "Indicates that the crossing type alarm was an above threshold.";
      }
      enum set-alarm {
        description
          "Indicates that the crossing type alarm was a set threshold.";
      }
      enum clear-alarm {
        description
          "Indicates that the crossing type alarm was a clear threshold.";
      }
    }
    description
     "This object indicates the crossing type that caused the generation of
      the notification.";
    reference
      "MEF-SOAM-PM-MIB.mefSoamPmNotificationObjCrossingType";
```


Copyright© 2022 Ciena® Corporation

```
}

leaf threshold-id {
  type enumeration {
    enum measured-flr-forward {
      description
      "Indicates the measured Frame Loss Ratio forward threshold generated
         the notification.";
    }
    enum max-flr-forward {
      description
      "Indicates the maximum Frame Loss Ration forward threshold generated
         the notification.";
    }
    enum average-flr-forward {
      description
       "Indicates the average Frame Loss Ratio forward threshold generated
         the notification.";
    }
    enum measured-flr-backward {
      description
        "Indicates the measured Frame Loss Ratio backward threshold
         generated the notification.";
    }
    enum max-flr-backward {
      description
      "Indicates the maximum Frame Loss Ratio backward threshold generated
         the notification.";
    }
    enum forward-high-loss {
      description
        "Indicates the forward High Loss threshold generated the
         notification.";
    }
    enum forward-consecutive-high-loss {
      description
      "Indicates the forward Consecutive High Loss threshold generated the
         notification.";
    }
    enum backward-high-loss {
      description
        "Indicates the backward High Loss threshold generated the
         notification.";
    }
    enum backward-consecutive-high-loss {
      description
      "Indicates the backward Consective High Loss threshold generated the
         notification.";
    }
    enum forward-unavailable-count {
      description
        "Indicates the forward Unavailable count threshold generated the
         notification.";
    }
    enum forward-available-ratio {
      description
        "Indicates the forward Available ration threshold generated the
         notification.";
    }
    enum backward-unavailable-count {
```


Copyright© 2022 Ciena® Corporation

```
  description
    "Indicates the backward Unavailable count threshold generated the
     notification.";
}
enum backward-available-ratio {
  description
    "Indicates the backward Available ratio threshold generated the
     notification.";
}
enum measured-frame-delay-two-way {
  description
    "Indicates the measured Frame Delay two-way threshold generated
     the notification.";
}
enum max-frame-delay-two-way {
  description
   "Indicates the maximum Frame Delay two-way threshold generated the
     notification.";
}
enum average-frame-delay-two-way {
  description
   "Indicates the average Frame Delay two-way threshold generated the
     notification.";
}
enum measured-inter-frame-delay-variation-two-way {
  description
    "Indicates the measured IFDV two-way threshold generated the
     notification.";
}
enum max-inter-frame-delay-variation-two-way {
  description
    "Indicates the maximum IFDV two-way threshold generated the
     notification.";
}
enum average-inter-frame-delay-variation-two-way {
  description
    "Indicates the average IFDV two-way threshold generat-ed the
     notification.";
}
enum max-frame-delay-range-two-way {
  description
  "Indicates the maximum Frame Delay Range two-way threshold generated
     the notification.";
}
enum average-frame-delay-range-two-way {
  description
  "Indicates the average Frame Delay Range two-way threshold generated
     the notification.";
}
enum measured-frame-delay-forward {
  description
   "Indicates the measured Frame Delay forward threshold generated the
     notification.";
}
enum max-frame-delay-forward {
  description
   "Indicates the maximum Frame Delay forward threshold generated the
     notification.";
}
enum average-frame-delay-forward {
```


Copyright© 2022 Ciena® Corporation

```
  description
   "Indicates the average Frame Delay forward threshold generated the
     notification.";
}
enum measured-inter-frame-delay-variation-forward {
  description
    "Indicates the measured IFDV forward threshold generated the
     notification.";
}
enum max-inter-frame-delay-variation-forward {
  description
    "Indicates the maximum IFDV forward threshold generated the
     notification.";
}
enum average-inter-frame-delay-variation-forward {
  description
    "Indicates the average IFDV forward threshold generated the
     notification.";
}
enum max-frame-delay-range-forward {
  description
  "Indicates the maximum Frame Delay Range forward threshold generated
     the notification.";
}
enum average-frame-delay-range-forward {
  description
  "Indicates the average Frame Delay Range forward threshold generated
     the notification.";
}
enum measured-frame-delay-backward {
  description
  "Indicates the measured Frame Delay backward threshold generated the
     notification.";
}
enum max-frame-delay-backward {
  description
   "Indicates the maximum Frame Delay backward threshold generated the
     notification.";
}
enum average-frame-delay-backward-threshold {
  description
   "Indicates the average Frame Delay backward threshold generated the
     notification.";
}
enum measured-inter-frame-delay-variation-backward {
  description
    "Indicates the measured IFDV backward threshold gener-ated the
     notification.";
}
enum max-inter-frame-delay-variation-backward {
  description
    "Indicates the maximum IFDV backward threshold gener-ated the
     notification.";
}
enum average-inter-frame-delay-variation-backward {
  description
    "Indicates the average IFDV backward threshold gener-ated the
     notification.";
}
enum max-frame-delay-range-backward {
```


Copyright© 2022 Ciena® Corporation

```
          description
            "Indicates the maximum Frame Delay Range backward threshold
             generated the notification.";
        }
        enum average-frame-delay-range-backward {
          description
            "Indicates the average Frame Delay Range backward threshold
             generated the notification.";
        }
      }

      description
       "This object identifies the threshold that caused the generation of the
          notification.";
      reference
        "MEF-SOAM-PM-MIB.mefSoamPmNotificationObjThresholdId";
    }

    leaf threshold-config-value {
      type uint32;

      description
      "The configured threshold value of the object that caused the generation
         of the notification.";
      reference
        "MEF-SOAM-PM-MIB.mefSoamPmNotificationObjThresholdConfig";
    }

    leaf measured-value {
      type uint32;

      description
       "The measured value of the object at the time of the generation of the
          notification.";
      reference
        "MEF-SOAM-PM-MIB.mefSoamPmNotificationObjThresholdValue";
    }

    leaf suspect {
      type suspect-status-type;

      description
        "The suspect flag for the current measurement interval in which the
         notification was generated.";
      reference
        "MEF-SOAM-PM-MIB.mefSoamPmNotificationObjSuspect";
    }
  }
}
```


Copyright© 2022 Ciena® Corporation

Copyright© 2022 Ciena® Corporation

39XX/51XX Switches and Platforms

YANG Model Reference

Copyright© 2022 Ciena® Corporation. All rights reserved.

SAOS 6.21.2
Publication: 009-3458-047
Document status: Revision
A
Document release date: March 2022

CONTACT CIENA
For additional information, office locations, and phone numbers, please visit the Ciena web site at **www.ciena.com**