

MEF

Technical Specification

MEF x.0

Service OAM Fault Management YANG Modules

April 2012

Disclaimer

The information in this publication is freely available for reproduction and use by any recipient and is believed to be accurate as of its publication date. Such information is subject to change without notice and the Metro Ethernet Forum (MEF) is not responsible for any errors. The MEF does not assume responsibility to update or correct any information in this publication. No representation or warranty, expressed or implied, is made by the MEF concerning the completeness, accuracy, or applicability of any information contained herein and no liability of any kind shall be assumed by the MEF as a result of reliance upon such information.

The information contained herein is intended to be used without modification by the recipient or user of this document. The MEF is not responsible or liable for any modifications to this document made by any other party.

The receipt or any use of this document or its contents does not in any way create, by implication or otherwise:

any express or implied license or right to or under any patent, copyright, trademark or trade secret rights held or claimed by any MEF member company which are or may be associated with the ideas, techniques, concepts or expressions contained herein; nor

any warranty or representation that any MEF member companies will announce any product(s) and/or service(s) related thereto, or if such announcements are made, that such announced product(s) and/or service(s) embody any or all of the ideas, technologies, or concepts contained herein; nor

any form of relationship between any MEF member companies and the recipient or user of this document.

Implementation or use of specific Metro Ethernet standards or recommendations and MEF specifications will be voluntary, and no company shall be obliged to implement them by virtue of participation in the Metro Ethernet Forum. The MEF is a non-profit international organization accelerating industry cooperation on Metro Ethernet technology. The MEF does not, expressly or otherwise, endorse or promote any specific products or services.

© The Metro Ethernet Forum 2012. All Rights Reserved.

Table of Contents

| | | |
|------------|--|-----------|
| 1. | Abstract..... | 1 |
| 2. | Terminology and Acronyms..... | 1 |
| 3. | Scope..... | 3 |
| 4. | Compliance Levels | 3 |
| 5. | Introduction..... | 4 |
| 5.1 | The Basic Need..... | 4 |
| 5.2 | The General Structure..... | 4 |
| 5.3 | The Foundational Elements | 5 |
| 6. | SOAM YANG CFM Overview | 6 |
| 6.1 | Type Definitions | 6 |
| 6.2 | Groupings | 7 |
| 6.3 | Data definitions..... | 8 |
| 6.4 | RPCs | 11 |
| 6.5 | Notifications | 11 |
| 7. | SOAM FM YANG Overview | 11 |
| 7.1 | Type definitions | 11 |
| 7.2 | Groupings | 12 |
| 7.3 | Data definitions..... | 12 |
| 7.4 | RPCs | 14 |
| 7.5 | Notifications | 14 |
| 8. | SOAM CFM YANG Requirements..... | 15 |
| 9. | SOAM FM YANG Requirements..... | 15 |
| 10. | SOAM CFM YANG MODULE..... | 17 |
| 11. | SOAM FM YANG MODULE..... | 52 |
| 12. | References | 81 |

List of Figures

| | |
|--|----|
| Figure 1 – Generalized OSS/BSS-NMS-EMS-NE Model..... | 5 |
| Figure 2 – Relationship between MEF and ITU-T specifications, the YANG language and MEF CFM and SOAM (PM and FM) modules | 6 |
| Figure 3 - CFM YANG Module Structure..... | 9 |
| Figure 4 - Fault Management YANG Module Structure | 13 |

List of Tables

| | |
|--|---|
| Table 1 – Terminology and Acronyms | 3 |
|--|---|

1. Abstract

This document specifies the Fault Management (FM) YANG Modules necessary to implement the Service Operations, Administration, and Maintenance (OAM) that satisfies the Service OAM requirements and framework specified by MEF 17 [9], the Service OAM Fault Management requirements as specified by SOAM-FM [11], and the Service OAM management objects as specified by MEF 7.1 [6] which are applicable to Fault Management functions. One MEF document, the SOAM TC/FM MIB [12] and two non-MEF documents serve as the baseline documents for this work: ITU-T Y.1731 [18] and IEEE 802.1Q [19]. YANG is the data modeling language used to model configuration and state data manipulated by the NETCONF protocol, and NETCONF notifications on the Network Element (NE) [25][26][27].

2. Terminology and Acronyms

| Term | Definition | Source |
|----------|---|----------------------|
| AIS | Alarm Indication Signal | ITU-T Y.1731 [18] |
| API | Application Programming Interface | |
| BSS | Business Support System | |
| CoS | Class of Service | MEF 10.2.1 [7] |
| CFM | Connectivity Fault Management | IEEE Std 802.1Q [19] |
| EMS | Element Management System | MEF 7.1 [6] |
| ENNI | External Network-to-Network Interface | MEF 4 [5] |
| ETH-AIS | Ethernet Alarm Indication Signal function | ITU-T Y.1731 [18] |
| ETH-CC | Ethernet Continuity Check function | ITU-T Y.1731 [18] |
| ETH-LB | Ethernet Loopback function | ITU-T Y.1731 [18] |
| ETH-LCK | Ethernet Lock signal function | ITU-T Y.1731 [18] |
| ETH-LT | Ethernet Linktrace function | ITU-T Y.1731 [18] |
| ETH-RDI | Ethernet Remote Defect Indication function | ITU-T Y.1731 [18] |
| ETH-Test | Ethernet Test function | ITU-T Y.1731 [18] |
| EVC | Ethernet Virtual Connection | MEF 10.2.1 [7] |
| FM | Fault Management | MEF 17 [9] |
| IEEE | Institute of Electrical and Electronics Engineers | |
| IETF | Internet Engineering Task Force | |
| ITU-T | International Telecommunication Union - Telecommunication Standardization Bureau | |
| LAN | Local Area Network | MEF 4 [5] |
| LCK | Locked, used in reference to LCK PDUs | ITU-T Y.1731 [18] |
| LBM | Loopback Message | IEEE Std 802.1Q [19] |
| LBR | Loopback Reply | IEEE Std 802.1Q [19] |
| LTM | Linktrace Message | IEEE Std 802.1Q [19] |
| LTR | Linktrace Reply | IEEE Std 802.1Q [19] |
| MAC | Media Access Control | IEEE Std 802.3 [21] |
| MA | Maintenance Association (equivalent to a MEG) | IEEE Std 802.1Q [19] |
| MAID | Maintenance Association Identifier (equivalent to a MEG ID) | IEEE Std 802.1Q [19] |

| Term | Definition | Source |
|----------------|--|--|
| MD | Maintenance Domain (equivalent to a OAM Domain in MEF 17) | IEEE Std 802.1Q [19] |
| MD Level | Maintenance Domain Level (equivalent to a MEG level) | IEEE Std 802.1Q [19] |
| ME | Maintenance Entity | IEEE Std 802.1Q [19] |
| MEF | Metro Ethernet Forum | |
| MEG | Maintenance Entity Group (equivalent to a MA) | ITU-T Y.1731 [18] |
| MEG ID | Maintenance Entity Group Identifier. Equivalent to Maintenance Association Identifier (MAID). | ITU-T Y.1731 [18] |
| MEG Level | Maintenance Entity Group Level (equivalent to MD Level) | ITU-T Y.1731 [18] |
| MEN | Metro Ethernet Network | MEF 4 [5] |
| MEP | Maintenance association End Point or MEG End Point | IEEE Std 802.1Q [19], ITU-T Y.1731 [18] |
| MHF | MIP Half Function | IEEE Std 802.1Q [19], |
| MIB | Management Information Base | RFC 2578 [2] |
| MIP | Maintenance Domain Intermediate Point or MEG Intermediate Point | IEEE Std 802.1Q [19], ITU-T Y.1731 [18] |
| MP | Maintenance Point. One of either a MEP or a MIP. | IEEE Std 802.1Q [19] |
| MTU | Maximum Transmission Unit | MEF 10.2.1 [7] |
| NE | Network Element | MEF 4 [5] |
| NETCONF | The Network Configuration Protocol | IETF RFC 6241 [25] |
| NETCONF Client | A NETCONF entity that invokes protocol operations on a NETCONF server. In addition, a client can subscribe to receive notifications from a server. | IETF RFC 6241 [25] |
| NETCONF Server | A NETCONF entity that executes protocol operations invoked by a NETCONF client. In addition, a server can send notifications to a client. | IETF RFC 6241 [25] |
| NNI | Network-to-Network Interface | MEF 4 [5] |
| NMS | Network Management System | MEF 7.1 [6] |
| OAM | Operations, Administration, and Maintenance | MEF 17 [9] |
| OSS | Operations Support System | ITU-T Y.1731 [18] |
| OVC | Operator Virtual Connection | MEF 26 [10] |
| PDU | Protocol Data Unit | IEEE Std 802.1Q [19] |
| RDI | Remote Defect Indication | IEEE Std 802.1Q [19] |
| RFC | Request for Comment | |
| RPC | Remote Procedure Call | |
| SOAM | Service OAM | MEF 17 [9] |
| SOAM PDU | Service OAM frame, or Protocol Data Unit. Specifically, those PDUs defined in [IEEE 802.1Q], [ITU-T Y.1731], or MEF specifications. | SOAM-FM [11] |
| Service Frame | An Ethernet frame transmitted across the UNI toward the Service Provider or an Ethernet frame transmitted across the UNI toward the Subscriber | MEF 10.2.1 [7] |

| Term | Definition | Source |
|--------------|--|-------------------------------|
| SNMP | Simple Network Management Protocol | RFC 1157 |
| SNMP Agent | An SNMP entity containing one or more command responder and/or notification originator applications (along with their associated SNMP engine). Typically implemented in an NE. | RFC 3411 [3] |
| SNMP Manager | An SNMP entity containing one or more command generator and/or notification receiver applications (along with their associated SNMP engine). Typically implemented in an EMS or NMS. | RFC 3411 [3] |
| TC | Textual Conventions | RFC 4181 [4] |
| TLV | Type Length Value, a method of encoding Objects | |
| TST | Test PDU | ITU-T Y.1731 [18] |
| UML | Unified Modeling Language | Object Management Group (OMG) |
| UNI | User-to-Network Interface | MEF 4 [5] |
| VID | VLAN Identifier | IEEE Std 802.1Q [19] |
| VLAN | Virtual LAN | IEEE Std 802.1Q [19] |
| YANG | A Data Modeling Language for NETCONF | IETF RFC 6020 [26] |

Table 1 – Terminology and Acronyms

3. Scope

The scope of this document is to provide the YANG modules that support the CFM and Service OAM (SOAM) Fault Management functions that have been defined in MEF 17 [9], the SOAM-FM IA [11], and MEF 7.1 [6], the *EMS-NMS Information Model*.

This document includes YANG definitions that support the original Connectivity Fault Management (CFM) functionality as defined by IEEE in 802.1Q [19] and 802.1ap [22], and the enhanced functionality as defined by the SOAM-FM [11] and implemented via the SOAM FM MIB [12].

The primary purpose of this document is to provide a mechanism to enhance interoperability between equipment/software vendors and Service Providers and/or Operators. This document provides the Metro Ethernet Forum (MEF) SOAM FM functionality within the Metro Ethernet Networks (MEN) via the industry standard YANG modeling language.

4. Compliance Levels

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in RFC 2119 [1]. All key words must be in upper case, bold text.

5. Introduction

5.1 The Basic Need

One of the aspects of defining Metro Ethernet Networks (MEN) is the need to ensure the compatibility between equipment/software vendors and equipment operators in order to facilitate interoperability in local, metro, national, and international networks. One of the common ways to do this is through a common management interface using publically available or enterprise specific NETCONF YANG Modules.

NETCONF is an IETF network management protocol defining a simple mechanism through which a device can be managed, configuration data information can be retrieved, and new configuration data can be uploaded and manipulated. YANG is a data modeling language to define the structure, syntax and semantics of data that can be used for NETCONF operations, including configuration, state data, RPCs and notifications. This allows the device to expose a full, formal Application Programming Interface (API). Applications can use this API to send and receive full and partial configuration data sets. NETCONF uses an Extensible Markup Language (XML)-based data encoding for the configuration data as well as the protocol messages.

The NETCONF protocol standard is defined in a set of RFCs. The base protocol is defined in RFC 6241 - *Network Configuration Protocol (NETCONF)* and the mandatory transport mapping is defined in RFC 6242 - *Using the NETCONF Protocol over Secure Shell (SSH)*. A list of additional transport mappings and extensions to the protocol is maintained by the NETCONF Working Group in the IETF.

The YANG language standard is defined by the IETF in a set of RFCs, primary of which are RFC 6020 *YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)* [26] and RFC 6087 [27] *Guidelines for Authors and Reviewers of YANG Data Model Documents*.

5.2 The General Structure

A generalized system model is shown by Figure 1 that illustrates the relationship between the OSS/BSS, NMS, EMS, and Network Elements (NE). The primary focus of this specification defines the interaction between the EMS and the NE via NETCONF using the YANG modules defined in this specification.

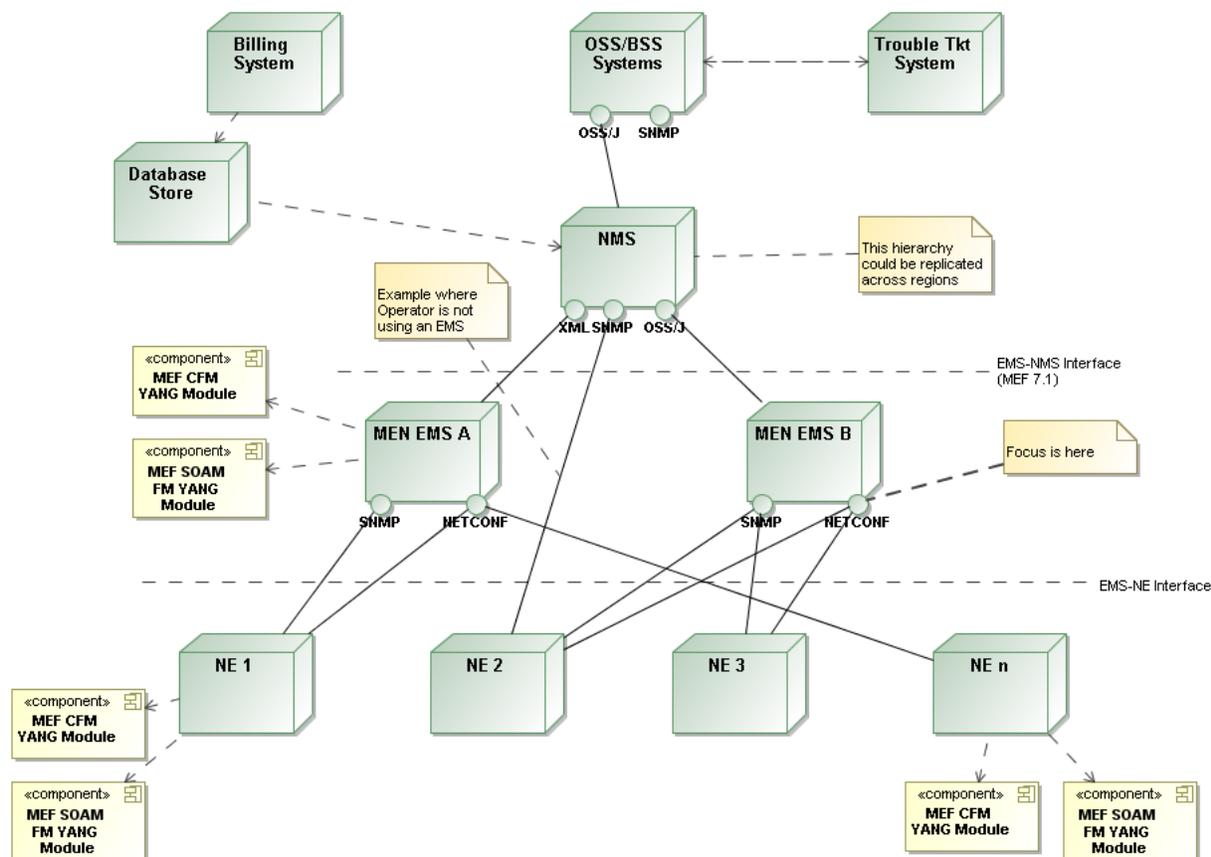


Figure 1 – Generalized OSS/BSS-NMS-EMS-NE Model

5.3 The Foundational Elements

MEF 17 [9] provides the Service OAM requirements and framework. It defines the OAM components and Service OAM requirements.

SOAM-FM [11] further defines the aspects of Service OAM requirements that deal with Fault Management (FM) and their extensions as needed to support MEF SOAM FM requirements.

SOAM-FM builds upon two existing documents: Connectivity Fault Management as defined in IEEE 802.1Q [19] and extended in ITU-T Y.1731 [18].

Service OAM Fault Management objects that provide the baseline for objects defined in this specification are found in MEF 7.1 [6].

MEF 7.1 draws heavily upon the models defined in ITU-T Q.840 [17].

The relationship between the various documents and the FM YANG modules presented in this specification is illustrated by Figure 2. The UML models found in MEF 7.1 and G.8052 provide a baseline for the SOAM YANG modules along with a number of the tables and objects in the IEEE CFM MIBs, the MEF-SOAM-FM and MEF-SOAM-TC MIBs as realized in the *mef-cfm*, *mef-soam-fm* and *mef-soam-pm* YANG modules.

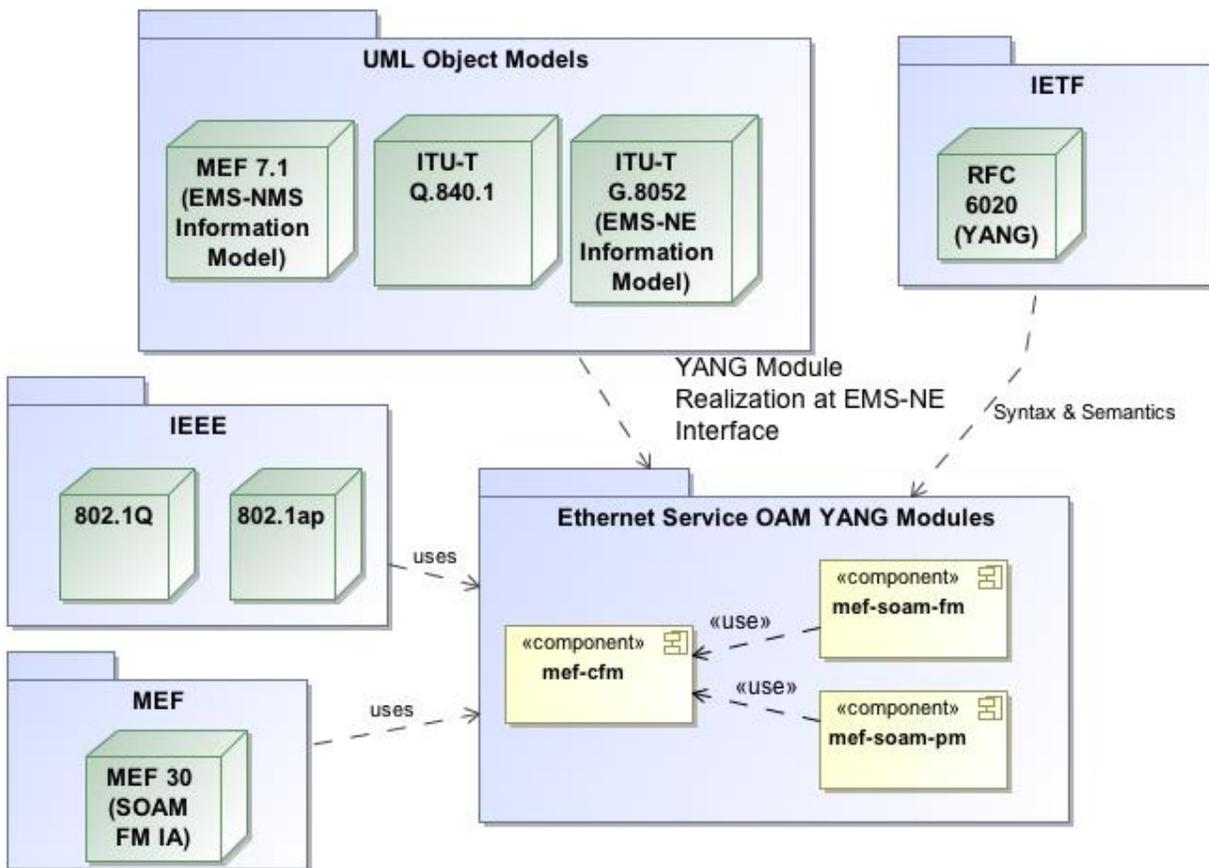


Figure 2 – Relationship between MEF and ITU-T specifications, the YANG language and MEF CFM and SOAM (PM and FM) modules

6. SOAM YANG CFM Overview

The YANG CFM Module implements the objects and functions found in IEEE 802.1Q [19], IEEE 802.1ap [22] and the corresponding objects found in IEEE8021-CFM-MIB, IEEE8021-CFM-V2-MIB, and IEEE8021-TC-MIB.

The purpose of the module is to provide the top level structure of Ethernet CFM including concepts like MD, MA/MEG and MEPs. These structures are then augmented with the SOAM FM data definitions in the SOAM YANG FM module.

The following are the major content deviations from the IEEE8021-CFM-MIB, IEEE8021-CFM-V2-MIB, and IEEE8021-TC-MIB:

- There is no equivalent of the CFM Stack Table (ieee8021CfmStackTable) as the retrieval of information about Maintenance Points configured on any given interface can be implemented using the basic query features of NETCONF

6.1 Type Definitions

Type definitions define derived types from base types using the *typedef* statement. A base type can be either a built-in type or a derived type, allowing a hierarchy of derived types.

The *typedef* statements is described in RFC6020 section 7.3.

The following type definitions are defined within the SOAM CFM YANG:

- **error-conditions-type** – A list of errors that may occur on creation or deletion of a MEP
- **fault-alarm-defect-bits-type** – A set of bits indicating active defects
- **fault-alarm-defect-type** - An enumerated value indicating the highest priority defect
- **id-permission-type** - An enumerated value indicating what, if anything, is to be included in a Sender ID TLV
- **interface-status-type** – A value obtained from the Interface Status TLV of a CCM
- **lbm-transaction-id-type** – The value to place in the Loopback Transaction Identifier field in the next LBM frame
- **ltm-transaction-id-type** - The value to place in the LTM Transaction Identifier field in the next LTM frame
- **md-level-type** – A value indicating the MD Level
- **mep-id-type** – A MEP identifier, unique over a given MA.
- **mhf-creation-type** – A set of enumerated values to control the creation of a MIP for a VID on a bridge port
- **port-status-type** – A value obtained from the Port Status TLV of a CCM
- **priority-type** – A 3 bit priority value to be used in the VLAN tag
- **remote-mep-state-type** – An enumerated value indicating the operational state of a Remote MEP state machine
- **vlan-id-type** – A 12-bit VLAN-ID used in the VLAN Tag header that uniquely identifies a VLAN
- **mac-address-and-uint-type** – A MAC address and a two-octet unsigned integer
- **component-id-type** – A component identifier used to distinguish between multiple virtual bridge instances

6.2 Groupings

Groupings are reusable collections of nodes using the *grouping* statement. Groupings of definitions are instantiated using the *uses* statement.

The *grouping* and *uses* statements are defined in RFC6020 sections 7.11, and 7.12 respectively.

The following groupings are defined for CFM YANG and are used throughout the CFM module.

- **target-address-group** – A group of data definitions identifying a target MEP or MAC address
- **loopback-parameters-group** – A group of data definitions associated with specific loopback sessions

- **linktrace-parameters-group** - A group of data definitions associated with specific linktrace sessions
- **md-level-group** – A group of data definitions related to the Default MD Level objects
- **port-id-tlv-group** – A group of data definitions related to 802.1AB [20] Port ID TLV
- **sender-id-tlv-group** – A group of data definitions related to IEEE 802.1Q [19] Sender ID TLV
- **maintenance-domain-reference** – A group of data definitions describing a reference to a specific maintenance domain
- **maintenance-association-reference** – A group of data definitions describing a reference to a specific maintenance association
- **maintenance-association-end-point-reference** – A group of data definitions describing a reference to a specific maintenance association end point

6.3 Data definitions

This section contains descriptions of the data definitions for SOAM CFM. Data definitions defines new data nodes in the data tree.

Figure 3 below describes the general structure of the YANG module. It closely corresponds to the definition of the CFM Entities in IEEE 802.1Q and is used as the basic structure to be extended by SOAM FM and PM.

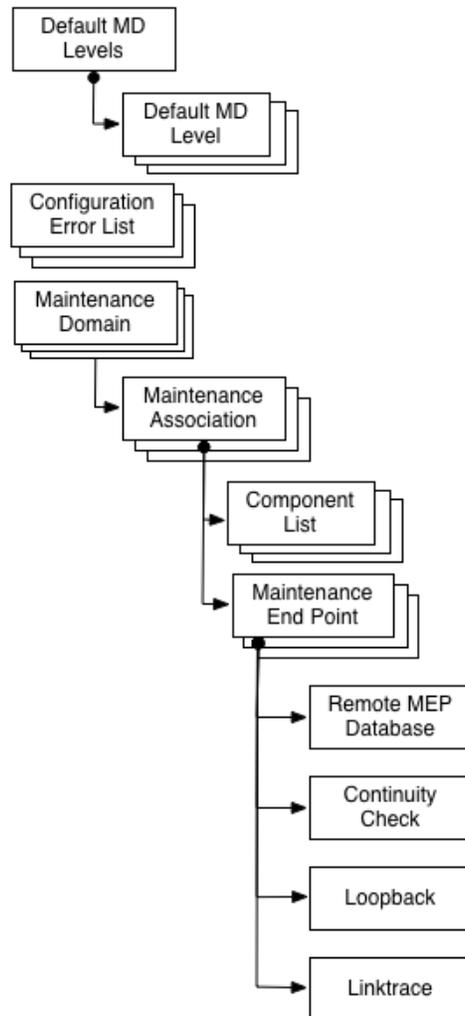


Figure 3 - CFM YANG Module Structure

The list entries below describes the informal name of all entries in bold text, e.g. **Default MD Level**, and the formal YANG definition statement in bold italics enclosed in parenthesis e.g. (*list default-md-level*). The list entries also include a description of the relative location of the data definition in the SOAM CFM module tree.

- **Default MD Levels** (*container default-md-levels*)

This container contains data definitions related to the global default MD level and the default MD levels associated with specific VIDs. The MD levels controls MIP Half Function (MHF) creation for VIDs that are not contained in the list of VIDs attached to any specific Maintenance Association managed object. It also controls the transmission of the Sender ID TLV by those MHFs. The parameters related to the global default MD level is located directly in this container while the default MD levels associated with VIDs are located in the *default-md-level* list.

This list is located on the top of the module.
- **Default MD Level** (*list default-md-level*)

This list contains the data definitions for the default MD level associated with specific

VIDs.

This list is contained in the *default-md-levels* container

- **Configuration Error List (*list configuration-error-list*)**

This list contains Bridge Ports, aggregated ports, and VIDs that are incorrectly configured.

This list is located on the top of the module.
- **Maintenance Domain (*list maintenance-domain*)**

This list contains all Maintenance Domains (MDs) on the bridge. An MD is required in order to create an MA with a MAID. From this Maintenance Domain managed object, all Maintenance Association managed objects associated with that Maintenance Domain managed object can be accessed, and thus controlled.

This list is located on the top of the module.
- **Maintenance Association (*list maintenance-association*)**

This list contains all Maintenance Association (MAs) on the bridge; one for each service instance for which an MP is defined on that Bridge. From this Maintenance Association managed object, all Maintenance association End Point managed objects associated with that Maintenance Association managed object can be accessed, and thus controlled.
- **Component List (*list component-list*)**

This list contains all bridge components on the bridge with associated configuration.

This list is a member of the *maintenance-association* list
- **Maintenance association End Point (*list maintenance-association-end-point*)**

This list contains all Maintenance association End Points (MEPs) on the bridge; one for each MEP defined within that Bridge. From this Maintenance association End Point managed object, all management objects related to that MEP can be controlled.

This list is a member of the *maintenance-association* list.
- The *maintenance-association-end-point* list may contain the following containers:
 - **Continuity Check (*container continuity-check*)**

This container contains data definitions related to Continuity Check (CC)
 - **Loopback (*container loopback*)**

This container contains data definitions related to Loopback (LB)
 - **Linktrace (*container link-trace*)**

This container contains data definitions related to Link Trace (LT)

This container contains a database container (*container linktrace-database*) that provides data definitions for historical linktrace results. The database container uses the *linktrace-parameters-group* grouping.
 - **Remote MEP Database (*container remote-mep-database*)**

This container contains data definitions related to the MEP database. The MEP database maintains received information about other MEPs in the Maintenance Association.

6.4 RPCs

Custom NETCONF RPCs are defined using the *rpc* statement. An RPC has a name and defines an optional set of input data using the *input* statement, and a set of return values using the *output* statement.

The *rpc* statement is defined in RFC6020 section 7.13.

The following RPCs are defined within the SOAM CFM YANG and are used by clients to perform specific tasks.

- **transmit-loopback** – This RPC creates a new loopback session associated with a specific MEP.
- **abort-loopback** – This RPC aborts a current loopback session associated with a specific MEP
- **transmit-linktrace** – This RPC creates a new linktrace session associated with a specific MEP. The return value is the transaction identifier for the newly created linktrace session.

6.5 Notifications

Custom NETCONF notifications are defined using the *notification* statement. A notification can contain arbitrary data defined in the YANG module.

The *notification* statement is defined in RFC6020 section 7.14.

The NETCONF notifications feature provides filtering capabilities allowing subscribers to filter out events based on event content. This feature makes the configuration to suppress specific notifications as defined in the *mefSoamAlarmEnable* MIB object in the MEF-SOAM-FM-MIB redundant.

The following notification is defined within the SOAM CFM YANG.

- **fault-alarm** – This notification signals an alarm condition associated with a specific MEP.

This notification carries the following data definitions:

- The ID of the affected MEP
- The current active defects

7. SOAM FM YANG Overview

The YANG FM Module implements the objects and functions found in SOAM FM IA [11] and various objects and tables from the MEF-SOAM-FM and MEF-SOAM-TC MIBs.

7.1 Type definitions

Type definitions define derived types from base types using the *typedef* statement. A base type can be either a built-in type or a derived type, allowing a hierarchy of derived types.

The *typedef* statement is described in RFC6020 section 7.3.

The following type definitions are defined within the SOAM FM YANG Module and are used locally in the SOAM FM YANG module.

- **interval-type-ais-lck-type** – This type defines the AIS/LCK interval (transmission period for a frame)
- **operational-state-type** – This type defines the operational state (current capability) of the MEP.
- **test-pattern-type** – This type indicates the type of test pattern to be sent in an OAM PDU data TLV

7.2 Groupings

Groupings are reusable collections of nodes using the *grouping* statement. Groupings of definitions are instantiated using the *uses* statement.

The *grouping* and *uses* statements are defined in RFC6020 sections 7.11, and 7.12 respectively.

The following groupings are defined for SOAM FM YANG and are used throughout the FM module.

- **locked-signal-parameters-group** – Data definitions related to the administrative locking state of the MEP
- **test-signal-parameters-group** – Data definitions related to generating and receiving test signals (ETH-Test) frames
- **test-signal-stats-in-group** – Data definitions for statistics gathered while receiving test signals (ETH-Test) frames
- **test-signal-stats-out-group** – Data definitions for statistics gathered while sending of test signals (ETH-Test) frames

7.3 Data definitions

This section contains descriptions of the data definitions for SOAM CFM. Data definitions defines new data nodes in the data tree.

The figure below describes the extensions to the CFM module provided by SOAM FM. The content of the FM module is depicted as boxes with dotted borders, extending existing certain CFM data definitions (e.g. MA Augments) and adding new data definitions (e.g. Alarm Indication Signal) to the tree.

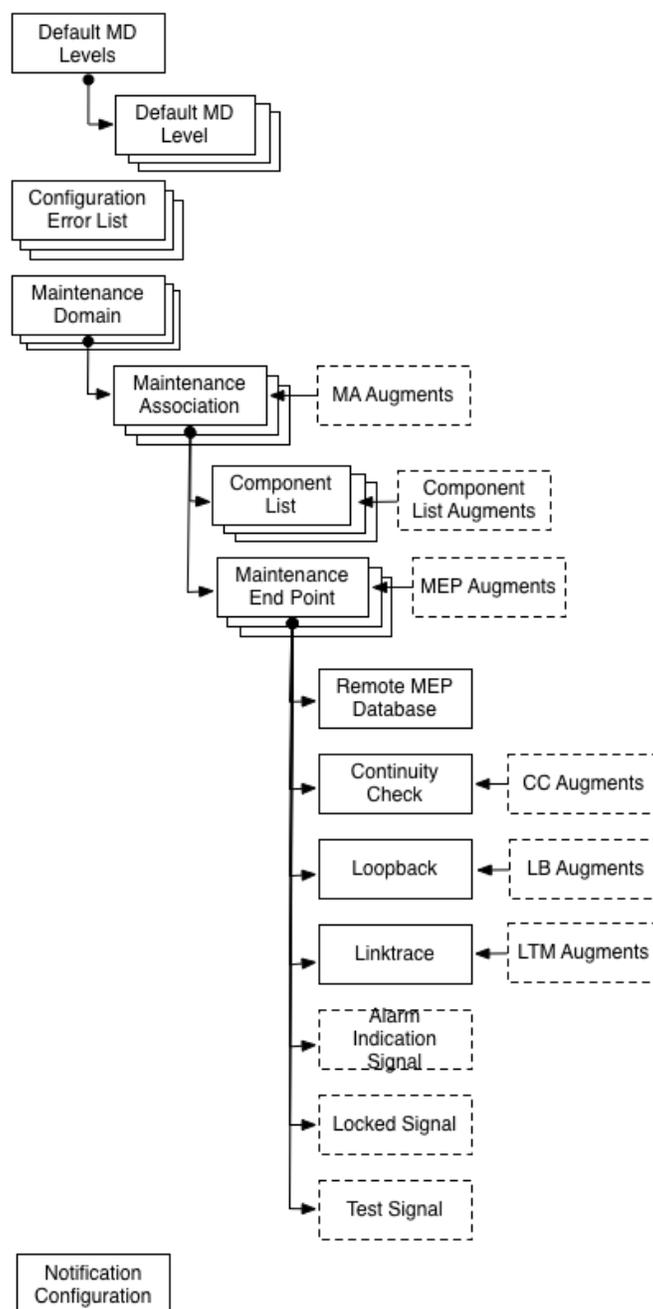


Figure 4 - Fault Management YANG Module Structure

The list entries below describes the informal name of all grouped definitions in bold text, e.g. **Augment of CFM Maintenance Association (MA)/MEG** and the formal YANG definition statement where applicable in bold italic enclosed in parenthesis e.g. *(list default-md-level)*. The list entries also include a description of the relative location of the data definition in the SOAM FM module tree.

- **Augment of CFM Maintenance Association (MA/MEG)** – A set of data definitions augmenting the CFM MA/MEG with parameters related to SOAM FM

- **Augment of CFM Component List** – A set of data definitions augmenting the CFM per-MA/MEG Component List with parameters related to SOAM FM
- **Augment of CFM Maintenance association End Point (MEP)** – A set of data definitions augmenting the CFM MEP with parameters related to SOAM FM
- **Augment of CFM Continuity Check (CC)** – A set of data definitions augmenting the CFM Continuity Check with parameters related to SOAM FM
- **Augment of CFM Link Trace (LTM)** – A set of data definitions augmenting the CFM Link Trace with parameters related to SOAM FM
- **Container Alarm Indication Signal (AIS)** – A set of data definitions augmenting the CFM MEP with parameters related to AIS.
- **Locked Signal (ETH-LCK)** – A set of data definitions extending the CFM MEP with parameters related to ETH-LCK. This containers uses the *locked-signal-parameters* grouping.
- **Test Signal (ETH-Test)** – A set of data definitions extending the CFM MEP with parameters related to ETH-Test. This containers uses the *test-signal-parameters* grouping.
- **Notification Configuration (*container notification-configuration*)** – A set of data definitions related FM event notifications.

7.4 RPCs

Custom NETCONF RPCs are defined using the *rpc* statement. An RPC has a name and defines an optional set of input data using the the *input* statement, and a set of optional return values using the *output* statement.

The *rpc* statement is defined in RFC6020 section 7.13.

The following RPCs are defined within the SOAM FM YANG Module and are used by clients to perform specific tasks.

- **create-test-signal** – This RPC creates a new test signal session associated with a specific MEP. It can be scheduled to execute immediately or at a later time.
- **abort-test-signal** – This RPC removes a scheduled test signal session that is scheduled to start at a later point in time or aborts an existing test signal session associated with a specific MEP

7.5 Notifications

Custom NETCONF notifications are defined using the *notification* statement. A notification can contain arbitrary data defined in the YANG module.

The *notification* statement is defined in RFC6020 section 7.14.

The NETCONF notifications feature provides filtering capabilities allowing subscribers to filter out events based on event content. This features makes the configuration to suppress specific

notifications as defined in e.g. the *mefSoamAlarmEnable* object in the MEF-SOAM-FM-MIB redundant.

The following notifications are defined within the SOAM FM YANG Module.

- **mep-defect-alarm** – This notification is generated when the value of *mep-defects* change. It indicates a persistent defect in the associated MEP.
This notification carries the following data definitions:
 - The ID of the affected MEP
 - The most recently sent defect
 - The value of the remote MEP state at the time of notification
- **configuration-error-assert** – This notification is generated when an entry is added to the *cfm:configuration-error-list* list in the CFM module. It indicates a configuration error during the setup of a SOAM FM entity.
- **configuration-error-clear** - This notification is generated when an entry is removed from the *cfm:configuration-error-list* list. It indicates that a configuration error has been removed during the setup of a SOAM FM entity.
This notification carries the following data definitions:
 - The removed entry in the *configuration-error* leafref.
- **mep-operational-state-changed** – This notification is sent when the value of a *operational-state* associated with the MEP changes.
This notification carries the following data definitions:
 - The ID of the affected MEP
 - The operational state at the time of notification
- **lock-alarm** – This notification is sent when either LCK *transmit-status* or *receive-status* changes.
This notification carries the following data definitions:
 - The ID of the affected MEP
 - The operational state at the time of notification
- **ais-alarm** – This notification is sent when the state of either AIS *transmit-status* or *receive-status* changes.
This notification carries the following data definitions:
 - A set of IDs identifying the affected MEP
 - The receive status at the time of notification
 - The transmit status at the time of notification

8. SOAM CFM YANG Requirements

The SOAM CFM YANG module defines the managed objects necessary to support SOAM CFM functionality. Its primary points of reference are ITU 802.1Q and ITU-T 802.1ap, and Y.1731 with corresponding SNMP MIBs (IEEE8021-CFM-MIB, IEEE8021-CFM-V2-MIB).

- [R1] All type definitions, groupings, data definitions, RPCs and notifications in the SOAM CFM YANG module **SHALL** be supported for devices that are compliant with SOAM FM YANG functionality.

9. SOAM FM YANG Requirements

The SOAM FM YANG module defines the managed objects necessary to support SOAM FM functionality. Its primary point of reference is the SOAM-FM Implementation Agreement [11].

- [R2] Type definitions, groupings, data definitions, RPCs and notifications in the SOAM FM YANG module **SHALL** be supported for devices that are compliant with SOAM FM YANG functionality except the following:
- [D1] Data definitions and RPCs related to the augmented continuity check (CC) function **SHOULD** be supported for devices that are compliant with SOAM FM YANG functionality
- [D2] Data definitions and RPCs related to alarm indication signal (AIS) **SHOULD** be supported for devices that are compliant with SOAM FM YANG functionality
- [D3] Data definitions and RPCs related to locking signal (ETH-LCK) **SHOULD** be supported for devices that are compliant with SOAM FM YANG functionality
- [D4] Data definitions and RPCs related to test signal (ETH-Test) **SHOULD** be supported for devices that are compliant with SOAM FM YANG functionality

10. SOAM CFM YANG MODULE

```

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";
  }

  //
  // 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
        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.DotlagCfmInterfaceStatus";
}

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.";
        }
        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";
  }
  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
        "[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
        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
    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
    "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
      "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";
      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";
    }
  }
}

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 'chassis 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 {
    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
    "[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;
  }
}
```

```
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;
  }
  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';
    }
  }
  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";
        }
        description
            "A reference to a specific Maintenance Domain.";
    }
}

grouping maintenance-association-reference {
    description
        "This grouping uniquely identifies a maintenance association. It consists
        of a maintenance-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 maintenance-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";

    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
    maintenanc-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;

    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.";
      }
      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
    (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";
}

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";
}

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
    "[802.1q] 12.14.6.1.3:e
    IEEE8021-CFM-MIB.DotlagCfmCcmInterval";
}

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
        Association.";

    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
    "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
      "[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
  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;
```

```
    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 {
    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";
    }
  }

  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";
}

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.DotlagCfmRelayActionFieldValue";
}

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 enforcement.";
    }
    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.";
    }
  }

  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.DotlagCfmIngressActionFieldValue";
}

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
        is returned in the LTR.";
    reference
        "[802.1q] 12.14.7.5.3:o, Table 21-32
        IEEE8021-CFM-MIB.DotlagCfmIngressActionFieldValue";
}

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";
            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 maintenance-domain-id {
    type string;

    description
      "The ID of a Maintenance Domain of the MEP on which a loopback session will
      be started";
  }

  leaf maintenance-association-id {
    type string;

    description
      "The ID of a Maintenance Association of the MEP on which a loopback session
      will be started";
  }

  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 maintenance-domain-id {
      type string;

      description
        "The ID of a Maintenance Domain of the MEP on which a loopback session
        should be administratively aborted";
    }

    leaf maintenance-association-id {
      type string;

      description
        "The ID of a Maintenance Association of the MEP on which a loopback session
        should be administratively aborted";
    }

    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 {
  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 maintenance-domain-id {
      type string;

      description
        "The ID of a Maintenance Domain of the MEP on which a linktrace session will
        be started";
    }

    leaf maintenance-association-id {
      type string;

      description
        "The ID of a Maintenance Association of the MEP on which a linktrace session
        will be started";
    }

    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 maintenance-domain-id {
  type string;

  description
    "The ID of a Maintenance Domain of the MEP on which a persistent defect
    has been detected";
}

leaf maintenance-association-id {
  type string;

  description
    "The ID of a Maintenance Association of the MEP on which a persistent
    defect has been detected";
}

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.";
}
}
}
}

```

11. SOAM FM YANG MODULE

```

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
//

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
                "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.";
    }
  }
  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" {
    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
            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";
    }
    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";
    }

    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.";

  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";

  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
    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";
}

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 {
  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
    "[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 {
  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
    "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;

    description
      "A list of responses from a Multicast Loopback Message";
    reference
      "[MEF31] MEF-SOAM-FM-MIB.mefSoamLbrMulticastTable";

    leaf transaction-id {
      type cfm:lbrm-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";
  }

  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";
    }
  }
}

```

```
}

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
    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";
}
}

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;

    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";
    }

    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.

        The content of the test-signal container in the associated MEP
        will be updated throughout the session upon successful creation.";

    input {
        leaf maintenance-domain-id {
            type string;

            description
                "The ID of a Maintenance Domain of the MEP on which a test session will
                be started";
        }

        leaf maintenance-association-id {
            type string;

            description
                "The ID of a Maintenance Association of the MEP on which a test session

```

```

        will be started";
    }

    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 maintenance-domain-id {
            type string;

            description
                "The ID of a Maintenance Domain of the MEP on which a test session will
                be aborted";
        }

        leaf maintenance-association-id {
            type string;

            description
                "The ID of a Maintenance Association of the MEP on which a test session
                will be aborted";
        }

        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 maintenance-domain-id {
  type string;

  description
    "The ID of a Maintenance Domain of the MEP on which the persistent
    defect is present";
}

leaf maintenance-association-id {
  type string;

  description
    "The ID of a Maintenance Association of the MEP on which the persistent
    defect is present";
}

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
    "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
      "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 maintenance-domain-id {
    type string;

    description
      "The ID of a Maintenance Domain of the MEP on which the operational state
      changed";
  }

  leaf maintenance-association-id {
    type string;

    description
      "The ID of a Maintenance Association of the MEP on which the operational
      state changed";
  }

  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 maintenance-domain-id {
        type string;

        description
            "The ID of a Maintenance Domain of the MEP on which the lock alarm is
            present";
    }

    leaf maintenance-association-id {
        type string;

        description
            "The ID of a Maintenance Association of the MEP on which the lock alarm
            is present";
    }

    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 {
        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 maintenance-domain-id {
  type string;

  description
    "The ID of a Maintenance Domain of the MEP on which the AIS alarm is
    present";
}

leaf maintenance-association-id {
  type string;

  description
    "The ID of a Maintenance Association of the MEP on which the AIS alarm
    is present";
}

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";
}
}
```

12. References

- [1] Bradner, S., *Key words for use in RFCs to Indicate Requirement Levels*, RFC 2119, March 1997. (Normative)
- [2] McCloghrie, K., et al., *Structure of Management Information Version 2 (SMIv2)*, RFC 2578, April 1999.
- [3] Harrington, D, et al, *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*, RFC 3411, December 2002.
- [4] Heard, C., *Guidelines for Authors and Reviewers of MIB Documents*, RFC 4181, September, 2005.
- [5] Metro Ethernet Forum, MEF 4, *Metro Ethernet Network Architecture Framework - Part 1: Generic Framework*, May 2004.
- [6] Metro Ethernet Forum, MEF 7.1, *Phase 2 EMS-NMS Information Model*, October 2009.
- [7] Metro Ethernet Forum, MEF 10.2.1, *Ethernet Services Attributes Phase 2*, January 2011.
- [8] Metro Ethernet Forum, MEF 15, *Requirements for Management of Metro Ethernet Phase 1 Network Elements*, November 2005.
- [9] Metro Ethernet Forum, MEF 17, *Service OAM Requirements & Framework – Phase 1*, April 2007.
- [10] Metro Ethernet Forum, MEF 26, *External Network Network Interface (ENNI) - Phase 1*, January 2010
- [11] Metro Ethernet Forum, MEF 30, *Service OAM Fault Management Implementation Agreement*, January 2011
- [12] Metro Ethernet Forum, MEF 31, *Service OAM Fault Management Definition of Managed Objects*, January 2011
- [13] International Telecommunication Union, Recommendation G.8011/Y.1307, *Ethernet over Transport – Ethernet services framework*, August 2004.
- [14] International Telecommunication Union, Recommendation G.8021/Y.1341, *Characteristics of Ethernet transport network equipment functional blocks*, December 2011.
- [15] International Telecommunication Union, Recommendation G.8051/Y.1345, *Management aspects of the Ethernet-over-Transport (EoT) capable network element*, October 2007.
- [16] International Telecommunication Union, Recommendation G.8051/Y.1345, *Management aspects of the Ethernet-over-Transport (EoT) capable network element*, October 2007.

- [17] International Telecommunication Union, Recommendation Q.840.1, *Requirements and Analysis for NMS-EMS Management Interface of Ethernet over Transport and Metro Ethernet Network*, March 2007
- [18] International Telecommunication Union, Recommendation Y.1731, *OAM functions and mechanisms for Ethernet based Networks*, July 2011.
- [19] IEEE Std 802.1Q-2011, *IEEE Standard for Local and metropolitan area networks-- Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks*, August 2011 (*This version incorporates 802.1ag*)
- [20] IEEE Std 802.3AB-2009, *IEEE Standard for Local and metropolitan area networks – Station and Media Access Control Connectivity Discovery*, 19 September 2009
- [21] IEEE Std 802.3-2008, *IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*, 26 December 2008.
- [22] IEEE Std 802.1ap-2008, *IEEE Standard for Local and metropolitan area networks - Virtual Bridged Local Area Networks Amendment 8: Management Information Base (MIB) Definitions for VLAN Bridges*, March 2009
- [23] International Organization for Standardization, *International Standard 8824 Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1)*, December, 1987.
- [24] Lam, K., et al., *Network Management Requirements for MPLS-based Transport Networks*, RFC 5951, September 2010.
- [25] Enns, R., *NETCONF Configuration Protocol*, RFC 6241, June 2011
- [26] Björklund, M., et al., *YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)*, RFC 6020, October 2010
- [27] Bierman, A., *Guidelines for Authors and Reviewers of YANG Data Model Documents*, RFC 6087, January 2011