Excentis DOCSIS config file editor
User Manual

support.configfileeditor@excentis.com
Excentis n.v.

Contents

List of Figures

Part I
Introduction

Chapter 1

Introduction to this Manual

Overview

Introduction

Welcome to the user manual of the Excentis DOCSIS config file editor . This manual will describe and explain how to use the editor to create and edit DOCSIS config files as described in the DOCSIS specifications.

This manual is organised in such a way, that everybody will directly find the information he/she needs.

This part of the manual will provide you with information regarding the manual, its structure and the conventions which are used throughout this manual.

Content

We will cover the following topics:

Organisation of this manual

Introduction	In this section, we will provide a short overview of each part and chapter of the manual.
Configurations And Settings	The first chapter on the Excentis DOCSIS config file editor explains the default settings and initialisation files used by the program.
User Interface Overview	This chapter guides you through the GUI. It will explain you how the userinterface is working and explain the different icons used in the GUI. After reading this part you will understand what you see when you start the editor.
Using the Editor	Chapter three describes the typical uses of the Excentis DOCSIS config file editor . The typical actions that can be done using the editor are described in detail. It is intended to be read when you have a specific question about how to do a certain action.
Appendix	The appendix contains an index that can be used for finding information about specific topics. It ends this manual with a bibliography. Furthernmore a reference is made of all supported TLVs and their textual representation in this appendix
Bibliography	A bibliography is added which references the original DOCSIS specifications which describe the TLV's and the format and use of the config files .

Continuing. . .

Organisation of this manual, *Continued*

Tip: We suggest you certainly read the User Interface part as it will help you understand how to work with the program. The chapter "Using the editor" is the ideal partner when you really want to edit and create DOCSIS config files, it will guide you through the typical actions needed to do so.

Structure and Conventions

Introduction

During this manual, we will use a fixed structure. We use Structured Writing, and we will explain the basic blocks of our document structure here.

When you are reading this, you are reading an example of an introduction.

Approach

We will help you in small steps, with clear examples and good illustrations. The smallest unit of information is called a block, and will contain a couple of lines and sometimes an illustration.

Tip: This is a tip to help you. A tip will contain useful information for dealing with a possible problem.

Warning: This is a warning. Reading the Excentis DOCSIS config file editor manual can be dangerous.

Note: This is a note.

Conclusion: This manual will show you how to use Excentis DOCSIS config file editor in an easy way.

Chapter 2

Introduction to the Excentis Config File Editor

Overview

Introduction

In this chapter the Excentis DOCSIS config file editor is introduced and its main features are listed.

Content

We will cover the following topics:

DOCSIS config files Overview

Introduction

This section of the manual provides some background information on the the DOCSIS config file .

The config file is a file defined by the cable operator that defines the service offered, limits imposed, features enabled on a cable modem. It is downloaded by the cable modem during the registration process and its content is communicated to the CMTS.

Content

We will cover the following topics:

Format

Introduction

The DOCSIS config file file is a binary file which is TLV encoded.

TLV Encoding

Type-Length-Value encoding is a binary encoding technique. It allows for the translation of structured (typed) information into a sequence of bytes. Before TLV encoding can be done all types need to be defined and a number is assigned per type. Per type the length of the information itself is also defined.

For DOCSIS config file the type and length are byte encoded. This restricts the number of types to 256. At first sight this is not much, however TLVs can be nested using sub-TLVs.

Sub-TLV

If the value of a TLV contains another TLV encoding then this is called a sub-TLV. Like this TLVs could be nested forever.

Suggested Reading

The CableLabs (2011) Annex C "Common TLV Encodings" describes the format of the config file together with all DOCSIS defined TLVs.

Vendor Specific TLV Encodings

Within the vendor specific TLV encoding (type 43) vendors define their own TLVs which configure the vendor specific features.

Different config file types

Types	DOCSIS defines 2 types of config files . One for DOCSIS 1.0 and one DOCSIS 1.1 style. There is no specific 2.0 nor 3.0 type of config file - only version specific TLV's were added for these new versions of DOCSIS.
1.0 style config file	This style of config file is only containing one CoS definition. No serviceflow (QoS) definition is contained in this type of config file .
1.1 style config file	This style of config file contains service flow QoS definitions and no Cos definition.

Editor Features

Binary Files	The editor can read and write binary formatted DOCSIS config files.
Textual Files	A textual representation of the config file can be imported or exported.
Shared Secret and Extended Shared Secret	The editor allows for the configuration both the shared secret and the extended shared secret
Cheat Mode	The editor has a "cheat mode" operation which will allow creating non-specification-conformant or 'illegal' config files for debugging and testing purposes.
TLV-tree	Next to the editable textual representation, the editor provides a tree like view on the config file to visualise the nesting of TLVs and sub-TLVs.

Part II
Excentis DOCSIS Config File Editor

Chapter 1

User Interface Overview

Overview

Introduction

This chapter focusses on the user interface of the Excentis DOCSIS config file editor . The different components are highlited and explained.

Content

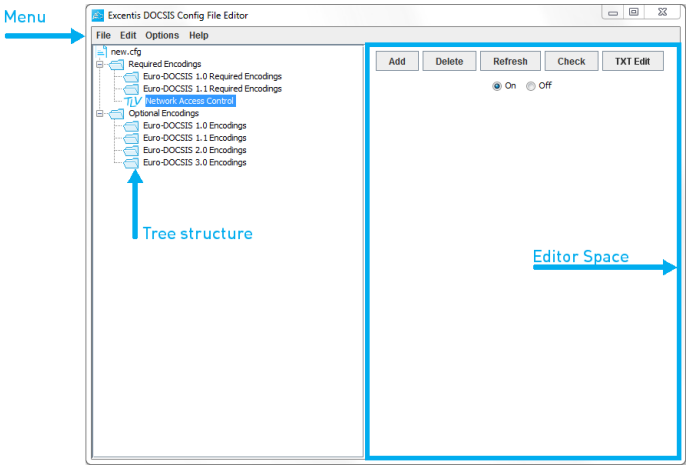
We will cover the following topics:

User Interface Overview

Main Interface

The illustration below shows the overview of the user interface of the editor. On the illustration the main components are annotated.

Illustration:



Main Interface

Menu

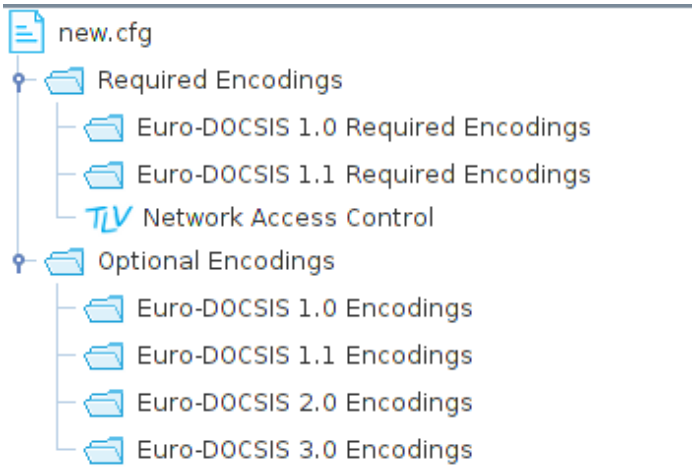
File	This part of the menu contains the typical file actions.
Edit	This part of the menu provides access to the configuration of the DOCSIS config file shared-secret. Using this menu you can change and view the shared-secret to be used.
Options	This part allows you to change the behaviour of the editor. The current option provided is the go to "cheat mode" which will excluce certain sanity checks from being executed when creating the binary configfile.
Help	The typical Help menu section provides access to the About box, Info box and License viewer.

Tree

Tree structure

The tree structure immediately shows the total overview of any config file . The example tree structure in the illustration below not only shows the relation TLV/sub-TLV but also visualises the difference between required options, optional options and the version of DOCSIS in which they were introduced.

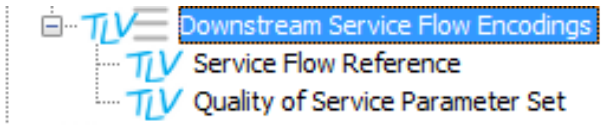
Illustration:



Tree visualisation

Tip: Unfolding sub-TLVs Click on the tree-wrench to fold open the sub-TLV.

Illustration:



Unfold sub-TLV

Continuing. . .

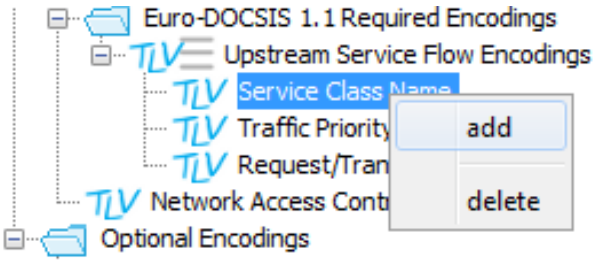
Tree, *Continued*

Tip: DOCSIS cable modems will ignore unknown TLVs found in a configuration file. Question remains, which TLV's will be understood by which version of DOCSIS cable modems? Therefore the TLV's are organised in the tree per DOCSIS version. The benefit of this is that you immediately see what TLV's will be understood/interpreted by which version of modem.

Note: Sorting per DOCSIS version only applies to toplevel TLVs. The DOCSIS extensionfield (TLV 43) sub-TLVs added per specific DOCSIS version are not sorted per DOCSIS version as the TLV43 itself was added in version 1.0

Tip: Clicking the right-mouse-button will result in a popup menu which can be used for edition the TLV.

Illustration:



Unfold sub-TLV

Editor Space

Introduction

The right hand side of the Excentis DOCSIS config file editor user interface shows the so called "Editor Space". The content of this area will change depending on the selected item or executed action.

Following items will be displayed:

- Textual config file representation
- TLV Specific Editors
- Action Buttons

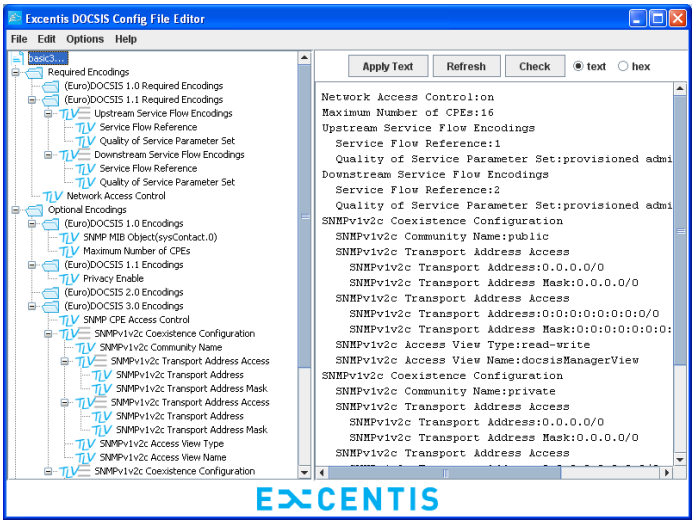
Continuing. . .

Editor Space, *Continued*

Textual Representation

When you click on the "TXT edit" button or on the name of the configfile (next to the root of the TLV tree structure) the textual version of the configfile is displayed in the Editor Space. In the textual representation, you can also edit the text and apply the changes by pressing the Apply Text button. You can toggle between readable text and a hexadecimal representation of the configfile.

Illustration:



Editor Space

Continuing. . .

Editor Space, *Continued*

Action Buttons

The top of the Editor Space always displays the typical Action Buttons Add, Delete, Apply Text, Refresh and Check

- "Add" adds a specific TLV (not available in Textual Representation mode)
- "Delete" deletes the currently selected TLV (not available in Textual Representation mode)
- "Apply Text" applies the textual representation to the binary configfile (only in Textual Representation mode)
- "Apply Hex" applies the hexadecimal representation to the binary configfile (only in Textual Representation mode)
- "Refresh" refreshes/reloads the treeview and textual representation and discards any not-applied changes in textual representation
- "Check" triggers the config file validation
- "verbose" this checkbox enables/disables the full TLV type number as comment. Note: comments are only present in textual version, not in binary version!
- "TXT Edit" moves the editor to textual representation mode (when not in Texual Representation mode)

The result and detailed use of these buttons is explaned in the next chapter "Using the editor"

TLV Specific Editors

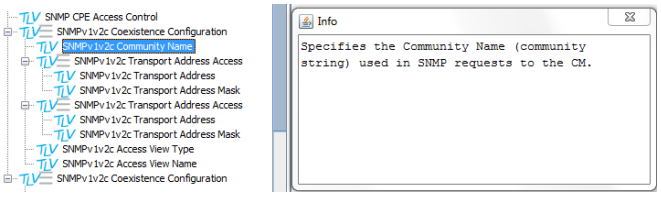
When a TLV is selected in the TLV-tree then the view of the Editor Space changes to a specfic editor showing the details of the selected TLV. This Specific TLV Editor is a convenient way to edit a TLV value.

Information Box

Context Help

The Excentis DOCSIS config file editor has a built in help system which shows context sensitive help with additional specification information. If you click an item in the tree structure the content of the Information box will change and show relevant information about the item clicked.

Illustration:



Information Box

Note: Hiding/showing the Information Box By default the Information Box is shown at startup of the program. It can be hidden by closing the window.

Re-opening the Information Box can be done via the Help menu.

Showing the Information Box

```
Menu -> Help -> Info(checkbox)
```

Note: Context Sensitive Help When hovering over the Tree View, or hovering over items in a selection dropbox, a ToolTip appears containing the TLV information from the Information box.

Chapter 2

Command Line Tool

Overview

Introduction

This chapter focusses on the command line functionality of the Excentis DOCSIS config file editor . The different methods are highlited and explained.

Content

We will cover the following topics:

Java command line

Java command line

The Excentis DOCSIS config file editor can also be used as a command-line tool, using the standard java JRE. The following functions are available:

- Showing the textual representation of a binary configfile (showfile)
- Writing the textual representation of a binary configfile to a textfile (bin2text)
- Converting a textual configfile to a binary configfile using optional sharedsecret and extended shared secret (text2bin)

Tip: You can use the GUI cheat mode (using generics etc.) by adding "-cheat" option to the command line

Syntax

```
java -classpath <path-to-jar>/coupe.jar
  com.excentis.configfile.CommandLine
  text2bin/bin2text/showfile [-cheat] inputfilename [outputfilename]
  [sharedsecret] [extendedsharedsecret]
```

Examples

```
e.g. 1: java -classpath jar/coupe.jar
  com.excentis.configfile.CommandLine
  text2bin basic.txt basic.cfg Euro ExtendedEuro
e.g. 2: java -classpath jar/coupe.jar
  com.excentis.configfile.CommandLine showfile basic.cfg
```

Chapter 3

Configurations and Settings

Overview

Introduction

This part of the manual describes the default startup configurations and settings

Content

We will cover the following topics:

Configuring the shared secret

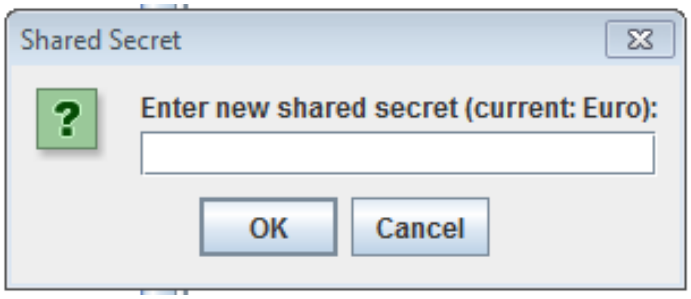
Introduction

The shared secret , is used during the provisioning process of the CM to authenticate the config file on the CMTS which itself is also configured to know the shared secret . The shared secret itself will not be part of the config file as it is only used during the creation of the config file , more precise the shared secret is used for CMTS-MIC calculation which is part of the config file .

Changing the shared-secret

```
Menu -> Edit -> Shared Secret
```

Illustration:



Enter the shared secret

Continuing. . .

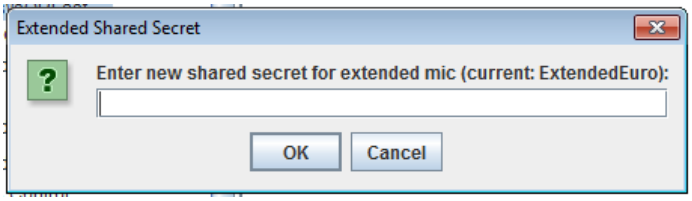
Configuring the shared secret , *Continued*

Changing the Extended Shared Secret (DOCSIS 3.0)

The extended shared secret was added in DOCSIS 3.0 to allow for an extended checking and authentication mechanism.

```
Menu -> Edit -> Extended Shared Secret
```

Illustration:



Enter the extended shared secret

Configuration of the SNMP OID translation database - MIBS.INI

Introduction

The Excentis DOCSIS config file editor keeps track of a text based database which is used to translate between the SNMP OID and the textual representation. Not only a number-to-text translation can be done using this database also the intended syntax of the OID can be retrieved.

Format of the MIBS.INI

The format of the MIBS.INI is straightforward. It is a flat textual file containing per line a definition of three elements.

- "Textual Name" The readable name of the OID
- "OID" The dotted-decimal representation of the OID, its position in the MIB tree
- "Type" The type of the encoding to use

Example - Showing all currently accepted types

```
ifNumber        1.3.6.1.2.1.2.1              Integer
sysObjectID     1.3.6.1.2.1.1.2              Object Identifier
ifDescr         1.3.6.1.2.1.2.2.1.2          Octet String
ifSpeed         1.3.6.1.2.1.2.2.1.5          Gauge
ifLastChange    1.3.6.1.2.1.2.2.1.9          Timeticks
atNetAddress    1.3.6.1.2.1.3.1.1.3          IP Address
ifInOctets      1.3.6.1.2.1.2.2.1.10         Counter
```

Location of MIBS.INI

In Windows and Linux, the MIBS.INI file is stored in the same directory as the application itself.

In MAC, MIBS.INI is embedded in the program package. Right-click teh executable, choose "Show Package Contents", and browse to Contents/Resources/Java. There you find the MIBS.INI file that is used.

Continuing. . .

Configuration of the SNMP OID translation database - MIBS.INI, *Continued*

Warning: After changing the MIBS.INI file a restart of Excentis DOCSIS config file editor is needed for changes to take effect

Warning: Sorting Order The order of the appearence in the MIBS.INI file will be the order in which the MIBs will be presented in the dropdown selection box when inserting a TLV 11.

Configuration of the default values - DEFAULTS.INI

Introduction

When the Excentis DOCSIS config file editor starts a default settings configuration file is read which can contain several preferential settings.

Format of the DEFAULTS.INI

The following keys are defined in the DEFAULTS.INI file.

- "shared_secret" The default shared secret to use
- "extended_shared_secret" The default extended shared secret to use
- "workingdir" The default path for config files during save/load operations. Note that the path of the last saved/opened file within a session is remembered.
- "cheatMode" If set to true or on or enabled, the cheatMode is enabled upon starting the editor.
- "startupConfig" Full path to a configfile which will be used as a template when starting the editor or selecting File–New
- "verboseMode" If set to true or on or enabled, the verbose mode is enabled upon starting the editor.

Example

```
shared_secret=Euro
extended_shared_secret=ExtendedEuro
workingdir=c:/tftp
cheatMode=true
startupConfig=c:/tftp/basic30.cfg
verboseMode=true
```

Continuing. . .

Configuration of the default values - DEFAULTS.INI, *Continued*

Location of the DEFAULTS.INI

In Windows and Linux, the DEFAULTS.INI file is stored in the same directory as the application itself.

In MAC, DEFAULTS.INI is embedded in the program package. Right-click teh executable, choose S̈how Package Contents, and browse to Contents/Resources/Java. There you find the DEFAULTS.INI file that is used.

Changing DEFAULTS.INI

After changing the DEFAULTS.INI file a restart of Excentis DOCSIS config file editor is needed for changes to take effect

Chapter 4

Using the editor

Overview

Introduction

This part of the manual describes the typical use cases of the configuration file editor.

Content

We will cover the following topics:

How to Open and Save config files

Introduction

The Excentis DOCSIS config file editor supports both binary encoded and textual files. The binary encoded files are encoded according to the DOCSIS specifications whereas the textual representation is a specific encoding only used by Excentis DOCSIS config file editor . The appendix provides a description of all supported TLVs and their textual representation.

Opening binary files

```
Menu -> File -> Open
```

Opening text formatted configfiles

```
Menu -> File -> Open TXT
```

Saving binary files

```
Menu -> File -> Save
  or
Menu -> File -> Save As
```

Saving to text formatted files

```
Menu -> File -> Save TXT
```

How to Edit the TLVs Overview

Introduction

This section of the manual describes process of editing the TLVs inside the config file .

Content

We will cover the following topics:

Adding TLV

Using the Action Button Add

```
Editor Space -> Action Buttons -> Add
```

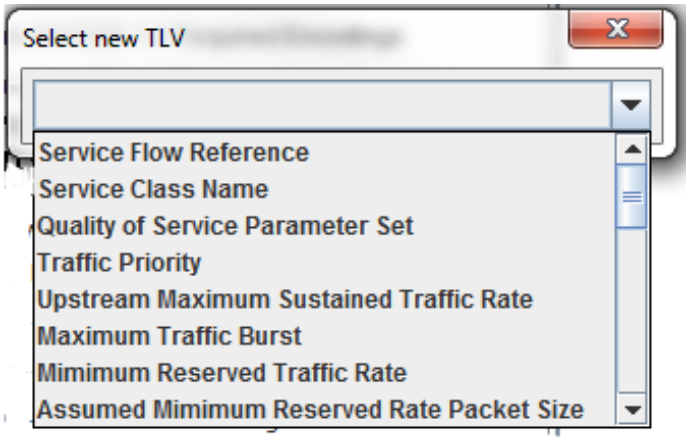
Using the Popup on TLV-tree

Using the right-mouse-button on the selected TLV in the TLV-Tree will result in a popup that allows for the selection of the Add action.

Selecting form dropdown

Once the "Add" action is initiated a selection box is popped-up. The drop-down selection includes all TLVs that can be added at the selected location.

Illustration:



Unfold sub-TLV

Deleting TLV

Select TLV first

Before a TLV can be deleted it needs to be selected in the TLV-Tree.

Using the Action Button Delete

`Editor Space -> Action Buttons -> Delete`

Using Popup on TLV-tree

Using the right-mouse-button on the selected TLV in the TLV-Tree will result in a popup that allows for the selection of the Delete action.

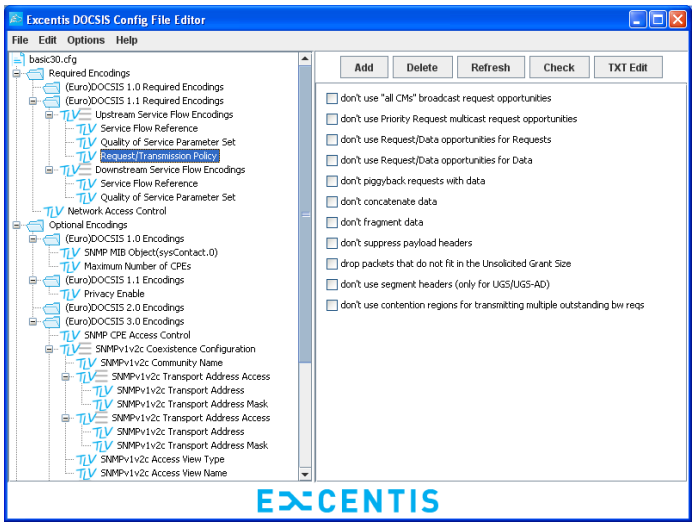
Editing a TLV - Specific TLV Editors

Introduction

TLVs require a specific formatting of the value part. To avoid erroneous formatting of this value part the Excentis DOCSIS config file editor provides a specific TLV editor per TLV. When a TLV is selected in the TLV-tree then the Editor Space will show a specific editor for the selected TLV.

In the illustration below an example is given of such a TLV specific editor.

Illustration:



Specific TLV editor for Request/Transmission Policy

Note: Before an edited TLV value is stored or updated the "Apply" button needs to be pressed. This does not save the config file , it only retains the value in memory.

```
Specific TLV Editor -> Apply
```

TLV 11 - SNMP Values

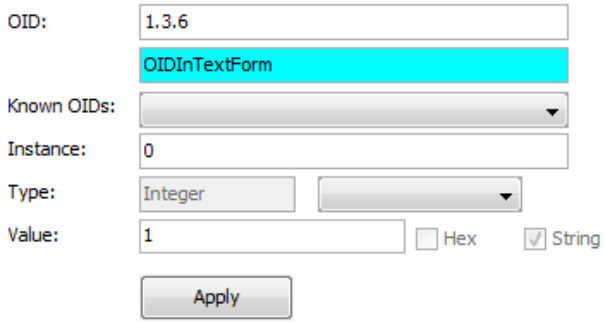
Introduction

SNMP values can be added in a config files . These SNMP values will be applied to the cable modem before it is online. Typical elements added via SNMP are IP filters or vendor specific or so called private MIB settings. Any SNMP OID can be added to the config file .

SNMP Specific Editor

The illustration below shows the TLV 11 editor which is used to edit existing OIDs or to add new OIDs to the config file .

Illustration:



TLV 11 Specific Editor

OID

The desired OID can be selected from the dropdown box or can be edited in the entry field. There is a second entry box which will show the textual representation of the OID. Any text can be given to an OID as it is only used as a lookup. What really counts is the dotted decimal OID notation in the OID entry because this value is used to encode the OID in its binary varbind form.

Continuing. . .

TLV 11 - SNMP Values, *Continued*

Tip: OID Database The Excentis DOCSIS config file editor keeps track of a lookup database for OID <-> name translations. If you add an OID which is not yet stored in that database a popup will ask you to add the new OID to the database.

The storage (MIBS.INI) and definition of the database is explained in the configuration chapter.

TLV 43 - Vendor Specific Settings

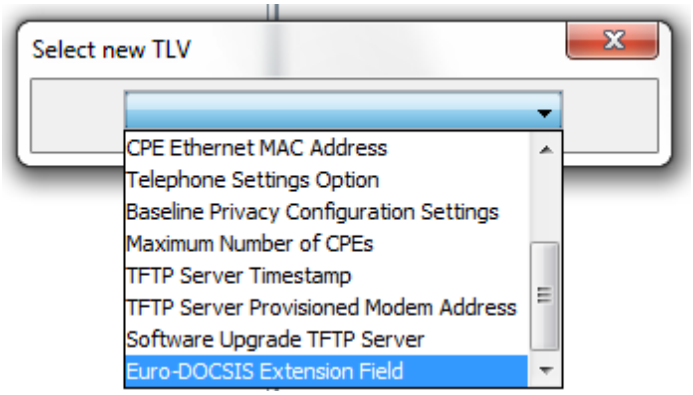
Introduction

Many TLVs have been defined in the DOCSIS specification, however many more have been defined by the cable modem vendors to specifically configure their equipment. This is where the TLV 43 is used, all vendor specific TLVs reside under the TLV 43 as sub-TLVs. TLV 43 can also be added as a sub-TLV for Service Flow, Classifier and PHS settings.

Add the TLV

You can add this TLV to the Optional Encodings, under Euro-DOCSIS 1.0 Encodings. It uses the same TLV number as the DOCSIS Extention Fields (43). For Service Flow, Classifier and PHS encodings, Select the TLV from the respective dropdown boxes with sub-TLVs.

Illustration:



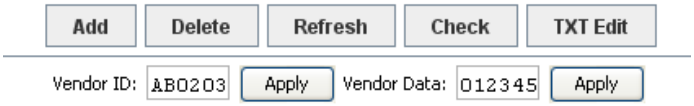
TLV 43

Continuing. . .

TLV 43 - Vendor Specific Settings, *Continued*

How Vendor Specific TLVs use the same TLV number as the DOCSIS Extension Fields (43). The distinction between both is based on the Vendor ID that is encoded as a sub-TLV. A Vendor ID of 0xFFFFFF indicates a DOCSIS Extension Field, any other value indicates a vendor specific setting. To create a Vendor Specific Setting, choose 'Euro-DOCSIS Extension Field' from the selection list, and change the Vendor ID into the Vendor's ID. The TLV will change into a 'Euro-DOCSIS vendor specific Extension Field' where you can enter the vendor specific data (as HEX data, also TLV encoded, e.g. 010103, for type 1 with length 1 and value 3).

Illustration:



TLV 43 configuration

TLV 43.6 - Extended CMTS MIC Configuration Settings

Introduction

The Extended CMTS MIC Configuration is one of the DOCSIS Extension Fields (subtype 6) and allows calculating a CMTS MIC over a chosed set of TLVs.

Usage

There are two modes for using the Extended CMTS MIC. In the first mode, the standard CMTS MIC (TLV 7) is overwritten, in the second mode, another TLV, the Explicit Extended CMTS MIC Digest is added. In both modes, the MICs are calculated only upon saving the configfile. To use the second mode, simply add an Explicit Extended CMTS MIC Digest TLV to the configfile. To use the first mode, delete the Explicit Extended CMTS MIC Digest TLV.

TLV 32 and TLV 33 - Code Verification Certificates (CVC)

Introduction

When a secure software upgrade is done for cable modems then the Code Verification Certificates are used. Cable modems will only successfully perform a secure software if the Manufacturer CVC (TLV 32) or Co-signer CVC (TLV 33) is present in the config file.

X.509

Digital certificates are commonly encoded according to the X.509 standard. The Excentis DOCSIS config file editor simply reads a binary encoded x.509 file and splits it into multiple TLVs.

Select

Select the desired X.509 file as the CVC.

```
Specific TLV Editor -> Select
```

Illustration:

Selecting the CVC

Other encodings

Introduction

The Excentis DOCSIS config file editor also allows encoding TLVs in the config file that are not destined for the CM itself.

eRouter

eRouter encodings are fully supported by the Excentis DOCSIS config file editor . The TLVs can be added by selecting the eRouter Configuration Encodings (202) from the dropdown in the 'Other Encodings' part of the selection tree.

Generic

In cheat mode, encodings that are currently not supported by the Excentis DOCSIS config file editor can be added as generic TLVs by selecting a Generic TLV from the dropdown in the 'Other Encodings' part of the selection tree. A default TLV type 0 is added. The type and value (in hex) can be edited afterwards.

Tip: For some TLVs, Generic Sub TLVs are available (cheat mode only), they can be selected from the dropdown list. These TLVs are:

- Upstream Service Flow
- Downstream Service Flow
- Upstream Packet Classification
- Downstream Packet Classification
- Euro-DOCSIS Extension Field
- eRouter Configuration
- eSTB Configuration

How to Check/validate the current config file

Introduction

What is a correct config file ? The Excentis DOCSIS config file editor has several checks built in to verify the correctness of a config file.

What is validated

During the validation process it is checked if:

- Network Access Control TLV is present
- Either Class Of Service (CoS) TLV or US/DS SF TLVs are present and not both at the same time
- For 1.0 style config file (CoS): Check if Class ID and Privacy Enable subTLVs are present
- For 1.1 style config file (QoS): Check if both US and DS service flows are present
- For US/DS SF: Are SF refreence, QoS ParameterSet defined
- For US SF: if scheduling type is not best effort: define request transmission policy
- Classifiers: verify Classifier and SF ref are present; SF ref refers to existing SF; either IP, IPv6, Eth or IEEE classifier subtype is present
- Payload Header Suppression: Classifier and SF ref are present and refer to existing SF/Classifiers; PHS size matches size of PHS field and PHS mask

Continuing. . .

How to Check/validate the current config file , *Continued*

How and When

If a file is saved then the validation checks are executed before the file is saved. If the file is not conformant, a popup describing the validation fail is shown, if the popup is ignored then the file can be saved anyways.

In explicit check can be executed by using the Action Button "Check"

```
Editor Space -> Action Buttons -> Check
```

Tip: The editor has an option to cheat on the validation checks.

In cheat mode no validation checks are done.

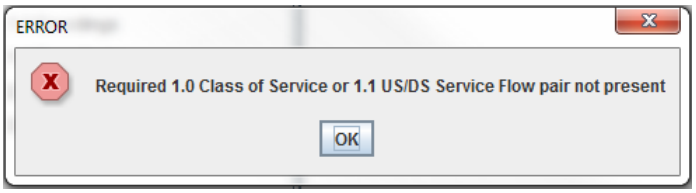
In cheat mode 'bad' config files can be generated.

```
Menu -> Options -> Cheat(checkbox)
```

Feedback

A popup will be generated explaining what chack failed during the execution of the checks.

Illustration:



Validation feedback

Continuing. . .

How to Check/validate the current config file , *Continued*

Warning: CMTS-MIC, CM-MIC check During the validation checks no check is done on the CMTS-MIC nor CM-MIC. At no point in time these MICs are verified for correctness by the Excentis DOCSIS config file editor , the editor only generates MICs at file save time. If the files are opened the MICs are stripped.

How to Control the TLV sequence order

Introduction

As defined in the DOCSIS specification only some TLVs require a strict/specific order.

- "Service Flow TLVs" The first to service flow TLV defines primary service flow
- "Long length (›256) TLVs" Long TLVs are split over multiple subsequent TLVs where the value part is concatenated. It is required that these TLVs are sequentially encoded.

For service flows the primary service flow will be the first one shown in the tree.

For long TLVs (e.g. CVC's) the TLV specific editors take care of generating the correct order.

Warning: Using the GUI the order of the TLVs is defined by the creation-order of the TLVs. If you really want to control the order of the TLV sequence the Textual Representation must be used. This requires you to edit a text file, open the text file and save it in binary format.

How to Edit config files as text files

Introduction

Sometimes it is convenient to use text files for editing the config files . For this purpose the Excentis DOCSIS config file editor has the capability to edit the textual representation and to load and save text versions of the config file .

Format of Textual Representation

The appendix contains a list of all TLVs supported by Excentis DOCSIS config file editor .

Tip:

The easiest way to edit the text is to simply edit and apply from within textual representation window.

You can also export the file as text so that you can edit is using your preferred text editor. You can use any plaintext editor to edit the exported text files.

Note: Indentation When config files are created in a textual form no care must be taken over the indentation. The TLV versus sub-TLV relation will be recognised from the name of the TLV.

Note: Comment All lines starting with # or // will be ignored while parsing the texual representation.

shared secret

The shared secret is not part of the textual representation neither are the CM nor CMTS-MIC. These TLVs are calculated at creation time of the binary version of the file.

Continuing. . .

How to Edit config files as text files, *Continued*

Tip: If you want to know how the value of a TLV looks like in text mode the best approach is to add the TLV using the GUI and switch to textual representation by clicking the filename.

Textual TLV List

```
Network Access Control:on
Maximum Number of CPEs:16
Upstream Service Flow Encodings
  Service Flow Reference:1
  Quality of Service Parameter Set:provisioned \
          admitted active
Downstream Service Flow Encodings
  Service Flow Reference:2
  Quality of Service Parameter Set:provisioned \
          admitted active
SNMPv1v2c Coexistence Configuration
  SNMPv1v2c Community Name:public
  SNMPv1v2c Transport Address Access
    SNMPv1v2c Transport Address:0.0.0.0/0
    SNMPv1v2c Transport Address Mask:0.0.0.0/0
  SNMPv1v2c Transport Address Access
    SNMPv1v2c Transport Address:0:0:0:0:0:0:0:0/0
    SNMPv1v2c Transport Address Mask:0:0:0:0:0:0:0:0/0
  SNMPv1v2c Access View Type:read-write
  SNMPv1v2c Access View Name:docsisManagerView
SNMPv1v2c Coexistence Configuration
  SNMPv1v2c Community Name:private
  SNMPv1v2c Transport Address Access
    SNMPv1v2c Transport Address:0.0.0.0/0
    SNMPv1v2c Transport Address Mask:0.0.0.0/0
  SNMPv1v2c Transport Address Access
    SNMPv1v2c Transport Address:0:0:0:0:0:0:0:0/0
    SNMPv1v2c Transport Address Mask:0:0:0:0:0:0:0:0/0
  SNMPv1v2c Access View Type:read-write
  SNMPv1v2c Access View Name:docsisManagerView
 SNMP CPE Access Control:on
 Privacy Enable:off
 SNMP MIB Object(sysContact.0):1.3.6.1.2.1.1.4.0, \
          Octet String, basic30.cfg
```

Part III
Appendix I - Supported TLV overview and textual name

What follows is the complete list of supported TLVs. The TLVs are ordered by number.
All TLVs are explained in detail in the CableLabs (2011), Annex C "Common TLV Encodings"
Textual TLV List

```
Downstream Frequency Configuration: 1

Upstream Channel ID Configuration: 2

Network Access Control: 3

Euro-DOCSIS 1.0 Class of Service Configuration: 4
    Class ID: 1
    Maximum Downstream Rate: 2
    Maximum Upstream Rate: 3
    Upstream Channel Priority: 4
    Guaranteed Minimum US Channel Data Rate: 5
    Maximum Upstream Channel Transmit Burst: 6
    Class-of-Service Privacy Enable: 7

Software Upgrade Filename: 9

SNMP Write-Access Control: 10

SNMP MIB Object: 11

CPE Ethernet MAC address: 14

Telephone Settings Option: 15

Baseline Privacy Configuration Settings: 17
    Authorize Wait Timeout: 1
    Reauthorize Wait Timeout: 2
    Authorization Grace Time: 3
    Operational Wait Timeout: 4
    Rekey Wait Timeout: 5
    TEK Grace Time: 6
    Authorize Reject Wait Timeout: 7
    SA Map Wait Timeout: 8
    SA Map Max Retries: 9

Maximum Number of CPEs: 18

TFTP Server Timestamp: 19

TFTP Server Provisioned Modem Address: 20

Software Upgrade TFTP server: 21
```

```
Upstream Packet Classification Encoding: 22
    Classifier Reference: 1
    Service Flow Reference: 3
    Rule Priority: 5
    Classifier Activation State: 6
    IP Packet Classification Encodings: 9
        IP Type of Service Range and Mask: 1
        IP Protocol: 2
        IP Source Address: 3
        IP Source Mask: 4
        IP Destination Address: 5
        IP Destination Mask: 6
        TCP/UDP Source Port Start: 7
        TCP/UDP Source Port End: 8
        TCP/UDP Destination Port Start: 9
        TCP/UDP Destination Port End: 10
    Ethernet LLC Packet Classification Encodings: 10
        Destination MAC Address: 1
        Source MAC Address: 2
        Ethertype/DSAP/MacType: 3
    IEEE 802.1P/Q Packet Classification Encodings: 11
        IEEE 802.1P User_Priority: 1
        IEEE 802.1Q VLAN_ID: 2
    IPv6 Packet Classification Encodings: 12
        IPv6 Traffic Class Range and Mask: 1
        IPv6 Flow Label: 2
        IPv6 Next Header Type: 3
        IPv6 Source Address: 4
        IPv6 Source Prefix Length: 5
        IPv6 Destination Address: 6
        IPv6 Destination Prefix Length: 7
    CM Interface Mask (CMIM) Encoding: 13
    IEEE 802.1ad S-Tag and C-Tag Frame Classification Encodings: 14
        S-TPID: 1
        S-VID: 2
        S-PCP: 3
        S-DEI: 4
        C-TPID: 5
        C-VID: 6
        C-PCP: 7
        C-CFI: 8
        S-TCI: 9
        C-TCI: 10
    IEEE 802.1ah Packet Classification Encodings: 15
        I-TPID: 1
        I-SID: 2
        I-TCI: 3
        I-PCP: 4
        I-DEI: 5
        I-UCA: 6
```

```
        B-TPID: 7
        B-TCI: 8
        B-PCP: 9
        B-DEI: 10
        B-VID: 11
        B-DA: 12
        B-SA: 13
    ICMPv4/ICMPv6 Packet Classification Encodings: 16
        ICMPv4/ICMPv6 Type Start: 1
        ICMPv4/ICMPv6 Type End: 2
    MPLS Classification Encodings: 17
        MPLS TC bits: 1
        MPLS Label: 2
    Vendor Specific Classifier Parameters: 43
    Euro-DOCSIS vendor specific Classifier Extension Field: 43
    Euro-DOCSIS Classifier Extension Field: 43 (see Euro-DOCSIS Extension Field for subencodings

Downstream Packet Classification Encoding: 23
    Classifier Reference: 1
    Service Flow Reference: 3
    Rule Priority: 5
    Classifier Activation State: 6
    IP Packet Classification Encodings: 9
        IP Type of Service Range and Mask: 1
        IP Protocol: 2
        IP Source Address: 3
        IP Source Mask: 4
        IP Destination Address: 5
        IP Destination Mask: 6
        TCP/UDP Source Port Start: 7
        TCP/UDP Source Port End: 8
        TCP/UDP Destination Port Start: 9
        TCP/UDP Destination Port End: 10
    Ethernet LLC Packet Classification Encodings: 10
        Destination MAC Address: 1
        Source MAC Address: 2
        EtherType/DSAP/MacType: 3
    IEEE 802.1P/Q Packet Classification Encodings: 11
        IEEE 802.1P User Priority: 1
        IEEE 802.1Q VLAN_ID: 2
    IPv6 Packet Classification Encodings: 12
        IPv6 Traffic Range And Mask: 1
        IPv6 Flow Label: 2
        IPv6 Next Header Type: 3
        IPv6 Source Address: 4
        IPv6 Source Prefix Length: 5
        IPv6 Destination Address: 6
        IPv6 Destination Prefix Length: 7
    CM Interface Mask (CMIM) Encoding: 13
    IEEE 802.1ad S-Tag and C-Tag Frame Classification Encodings: 14
```

```
            S-TPID: 1
            S-VID: 2
            S-PCP: 3
            S-DEI: 4
            C-TPID: 5
            C-VID: 6
            C-PCP: 7
            C-CFI: 8
            S-TCI: 9
            C-TCI: 10
        IEEE 802.1ah Packet Classification Encodings: 15
            I-TPID: 1
            I-SID: 2
            I-TCI: 3
            I-PCP: 4
            I-DEI: 5
            I-UCA: 6
            B-TPID: 7
            B-TCI: 8
            B-PCP: 9
            B-DEI: 10
            B-VID: 11
            B-DA: 12
            B-SA: 13
        ICMPv4/ICMPv6 Packet Classification Encodings: 16
            ICMPv4/ICMPv6 Type Start: 1
            ICMPv4/ICMPv6 Type End: 2
        MPLS Classification Encodings: 17
            MPLS TC bits: 1
            MPLS Label: 2
        Vendor Specific Classifier Parameters: 43
        Euro-DOCSIS Classifier vendor specific Extension Field: 43
        Euro-DOCSIS Classifier Extension Field: 43 (see Euro-DOCSIS Extension Field for s

    Upstream Service Flow Encodings: 24
        Service Flow Reference: 1
        Service Class Name: 4
        Quality of Service Parameter Set: 6
        Traffic Priority: 7
        Upstream Maximum Sustained Traffic Rate: 8
        Maximum Traffic Burst: 9
        Minimum Reserved Traffic Rate: 10
        Assumed Minimum Reserved Rate Packet Size: 11
        Timeout for Active QoS Parameters: 12
        Timeout for Admitted QoS Parameters: 13
        Maximum Concatenated Burst: 14
        Service Flow Scheduling Type: 15
        Request/Transmission Policy: 16
        Nominal Polling Interval: 17
        Tolerated Poll Jitter: 18
```

```
    Unsolicited Grant Size: 19
    Nominal Grant Interval: 20
    Tolerated Grant Jitter: 21
    Grants per Interval: 22
    IP Type of Service Overwrite: 23
    Unsolicited Grant Time Reference: 24
    Multiplier to Contention Request Backoff Window: 25
    Multiplier to Number of Bytes Requested: 26
    Upstream Peak Traffic Rate: 27
    Service Flow Required Attribute Mask: 31
    Service Flow Forbidden Attribute Mask: 32
    Service Flow Attribute Aggregation Rule Mask: 33
    Application Identifier: 34
    Buffer Control: 35
        Minimum Buffer: 1
        Target Buffer: 2
        Maximum Buffer: 3
    Aggregate Service Flow Reference: 36
    MESP Reference: 37
    Service Flow to IATC Profile Name Reference: 39
    AQM Encodings: 40
        SF AQM Disable: 1
        SF AQM Latency Target: 2
    Data Rate Unit Setting: 41
    Vendor Specific QoS Parameters: 43
    Euro-DOCSIS QoS vendor specific Extension Field: 43
    Euro-DOCSIS QoS Extension Field: 43 (see Euro-DOCSIS Extension Field for subencodings)

Downstream Service Flow Encodings: 25
    Service Flow Reference: 1
    Service Class Name: 4
    Quality of Service Parameter Set: 6
    Traffic Priority: 7
    Downstream Maximum Sustained Traffic Rate: 8
    Maximum Traffic Burst: 9
    Minimum Reserved Traffic Rate: 10
    Assumed Minimum Reserved Rate Packet Size: 11
    Timeout for Active QoS Parameters: 12
    Timeout for Admitted QoS Parameters: 13
    Maximum Downstream Latency: 14
    Downstream Resequencing: 17
    IP Type of Service Overwrite: 23
    Downstream Peak Traffic Rate: 27
    Service Flow Required Attribute Mask: 31
    Service Flow Forbidden Attribute Mask: 32
    Service Flow Attribute Aggregation Rule Mask: 33
    Application Identifier: 34
    Buffer Control: 35
        Minimum Buffer: 1
        Target Buffer: 2
```

```
        Maximum Buffer: 3
    Aggregate Service Flow Reference: 36
    MESP Reference: 37
    Service Flow to IATC Profile Name Reference: 39
    AQM Encodings: 40
        SF AQM Disable: 1
        SF AQM Latency Target: 2
    Data Rate Unit Setting: 41
    Vendor Specific QOS Parameters: 43
    Euro-DOCSIS QoS vendor specific Extension Field: 43
    Euro-DOCSIS QoS Extension Field: 43 (see Euro-DOCSIS Extension Field for subencod

Payload Header Suppression: 26
    Classifier Reference: 1
    Service Flow Reference: 3
    PHS Field: 7
    PHS Mask: 9
    PHS Size: 10
    PHS Verification: 11
    Vendor Specific PHS Parameters: 43
    Euro-DOCSIS PHS vendor specific Extension Field: 43
    Euro-DOCSIS PHS Extension Field: 43 (see Euro-DOCSIS Extension Field for subencod

Maximum Number of Classifiers: 28

Privacy Enable: 29

Manufacturer Code Verification Certificate: 32

Co-signer Code Verification Certificate: 33

Snmp V3 Kickstart Value: 34
    Snmp V3 Kickstart Security Name: 1
    Snmp V3 Kickstart Manager Public Number: 2

Subscriber Management Control: 35

Subscriber Management CPE IP Table: 36

Subscriber Management Filter Groups: 37

DOCSIS V3 Notification Receiver: 38
    IP Address of trap receiver: 1
    UDP Port number of trap receiver: 2
    Type of trap: 3
    Timeout for sending inform: 4
    Number of retries: 5
    Filtering Parameters: 6
    Security Name for SNMP v3 Notification: 7
    IPv6 Address of trap receiver: 8
```

```
Enable 2.0 mode: 39

Enable Test mode: 40

Downstream Channel List: 41
    Single Downstream Channel: 1
        Single Downstream Channel Timeout: 1
        Single Downstream Channel Frequency: 2
    Downstream Frequency Range: 2
        Downstream Frequency Range Timeout: 1
        Downstream Frequency Range Start: 2
        Downstream Frequency Range End: 3
        Downstream Frequency Range Step Size: 4
    Default Scanning: 3

Multicast MAC Address: 42

Euro-DOCSIS vendor specific Extension Field: 43

Euro-DOCSIS Extension Field: 43
    CM Load Balancing Policy ID: 1
    CM Load Balancing Priority: 2
    CM Load Balancing Group ID: 3
    CM Ranging Class ID Extension: 4
    L2VPN Encoding: 5
        VPN Identifier: 1
        NSI Encapsulation Subtype: 2
            Service Multiplexing Value Other: 1
            Service Multiplexing Value IEEE 802.1Q: 2
            Service Multiplexing Value IEEE 802.1ad: 3
            Service Multiplexing Value MPLS PW: 4
                MPLS Pseudowire ID: 1
                MPLS Peer IP address: 2
                MPLS Pseudowire Type: 3
                MPLS Backup Pseudowire ID: 4
                MPLS Backup Peer IP address: 5
            Service Multiplexing Value L2TPv3 Peer: 5
            Service Multiplexing Value IEEE 802.1ah Encapsulation: 6
                IEEE 802.1ah I-TCI: 1
                IEEE 802.1ah B-DA: 2
                IEEE 802.1ah B-TCI: 3
                IEEE 802.1ah I-TPID: 4
                IEEE 802.1ah I-PCP: 5
                IEEE 802.1ah I-DEI: 6
                IEEE 802.1ah I-UCA: 7
                IEEE 802.1ah I-SID: 8
                IEEE 802.1ah B-TPID: 9
                IEEE 802.1ah B-PCP: 10
                IEEE 802.1ah B-DEI: 11
```

```
            IEEE 802.1ah B-VID: 12
        Service Multiplexing Value IEEE 802.1ad S-TPID: 8
    eSAFE DHCP Snooping: 3
    CM Interface Mask (CMIM) Subtype: 4
    Attachment Group ID: 5
    Source Attachment Individual ID: 6
    Target Attachment Individual ID: 7
    Ingress User Priority: 8
    User Priority Range: 9
    BGP Attribute sub TLV: 21
        BGP VPNID: 1
        Route Distinguisher: 2
        Route Target (Import): 3
        Route Target (Import): 4
        CE-ID/VE-ID: 5
    Pseudowire Signaling: 23
    SOAM Subtype: 24
        MEP Configuration: 1
            MD Level: 1
            MD Name: 2
            MA Name: 3
            MEP Id: 4
        Remote MEP Configuration: 2
            Remote MD Level: 1
            Remote MD Name: 2
    Remote MA Name: 3
            Remote MEP Id: 4
        Fault Management Configuration: 3
            Continuity Check Messages:0
            Loopback Function:disabled
            Linktrace Function:disabled
        Performance Management Configuration: 4
            Frame Delay Measurement: 1
                Frame Delay Measurement Enable: 1
                Frame Delay Measurement One-way-Two-way: 2
                Frame Delay Measurement Transmission Periodicity: 3
            Frame Loss Measurement: 2
                Frame Loss Measurement Enable: 1
                Frame Loss Measurement Transmission Periodicity: 2
    L2VPN DSID: 26
    Vendor Specific L2VPN Subtype: 43
Extended CMTS MIC Configuration Setting: 6
    Extended CMTS MIC HMAC type: 1
    Extended CMTS MIC Bitmap: 2
    Explicit Extended CMTS MIC Digest: 3
SAV Authorization Encoding: 7
    SAV Group Name: 1
    SAV Static Prefix Rule: 2
Cable Modem Attribute Masks: 9
    CM Downstream Required Attribute Mask: 1
```

```
        CM Downstream Forbidden Attribute Mask: 2
        CM Upstream Required Attribute Mask: 3
        CM Upstream Forbidden Attribute Mask: 4
    IP Multicast Join Authorization Encoding:10
        IP Multicast Profile Name:1
        IP Multicast Join Authorization Static Session Rule:2
            RulePriority:1
            Authorization Action:2
            Source Prefix Address:3
            Source Prefix Length:4
            Group Prefix Address:5
            Group Prefix Length:6
        Maximum Multicast Sessions:3
    Service Type Identifier:11

Downstream Unencrypted Traffic Filtering: 45
    DUT Control: 1
    DUT CMIM: 2

SNMPv1v2c Coexistence Configuration: 53
    SNMPv1v2c Community Name: 1
    SNMPv1v2c Transport Address Access: 2
        SNMPv1v2c Transport Address: 1
        SNMPv1v2c Transport Address Mask: 2
    SNMPv1v2c Access View Type: 3
    SNMPv1v2c Access View Name: 4

SNMPv3 Access View Configuration: 54
    SNMPv3 Access View Name: 1
    SNMPv3 Access View Subtree: 2
    SNMPv3 Access View Mask: 3
    SNMPv3 Access View Type: 4

SNMP CPE Access Control: 55

Channel Assignment Configuration Settings: 56
    Transmit Channel Assignment Configuration Setting: 1
    Receive Channel Assignment Configuration Setting: 2

Software Upgrade IPv6 TFTP Server: 58

TFTP Provisioned Modem IPv6 Address: 59

Upstream Drop Packet Classification Encoding: 60
    Classifier Reference: 1
    Service Flow Reference: 3
    Rule Priority: 5
    Classifier Activation State: 6
    IP Packet Classification Encodings: 9
        IP Type of Service Range and Mask: 1
```

```
    IP Protocol: 2
    IP Source Address: 3
    IP Source Mask: 4
    IP Destination Address: 5
    IP Destination Mask: 6
    TCP/UDP Source Port Start: 7
    TCP/UDP Source Port End: 8
    TCP/UDP Destination Port Start: 9
    TCP/UDP Destination Port End: 10
Ethernet LLC Packet Classification Encodings: 10
    Destination MAC Address: 1
    Source MAC Address: 2
    Ethertype/DSAP/MacType: 3
IEEE 802.1P/Q Packet Classification Encodings: 11
    IEEE 802.1P User Priority: 1
    IEEE 802.1Q VLAN_ID: 2
IPv6 Packet Classification Encodings: 12
    IPv6 Traffic Range And Mask: 1
    IPv6 Flow Label: 2
    IPv6 Next Header Type: 3
    IPv6 Source Address: 4
    IPv6 Source Prefix Length: 5
    IPv6 Destination Address: 6
    IPv6 Destination Prefix Length: 7
CM Interface Mask (CMIM) Encoding: 13
IEEE 802.1ad S-Tag and C-Tag Frame Classification Encodings: 14
    S-TPID: 1
    S-VID: 2
    S-PCP: 3
    S-DEI: 4
    C-TPID: 5
    C-VID: 6
    C-PCP: 7
    C-CFI: 8
    S-TCI: 9
    C-TCI: 10
IEEE 802.1ah Packet Classification Encodings: 15
    I-TPID: 1
    I-SID: 2
    I-TCI: 3
    I-PCP: 4
    I-DEI: 5
    I-UCA: 6
    B-TPID: 7
    B-TCI: 8
    B-PCP: 9
    B-DEI: 10
    B-VID: 11
    B-DA: 12
    B-SA: 13
```

```
    ICMPv4/ICMPv6 Packet Classification Encodings: 16
        ICMPv4/ICMPv6 Type Start: 1
        ICMPv4/ICMPv6 Type End: 2
    MPLS Classification Encodings: 17
        MPLS TC bits: 1
        MPLS Label: 2
    Vendor Specific Classifier Parameters: 43
    Euro-DOCSIS Classifier vendor specific Extension Field: 43
    Euro-DOCSIS Classifier Extension Field: 43 (see Euro-DOCSIS Extension Field for subencodings

Subscriber Mgmt CPE IPv6 Prefix List: 61

Upstream Drop Classifier Group ID: 62

Subscriber Mgmt Control Max CPE IPv6 Addresses: 63

CMTS Static Multicast Session Encoding: 64
    Static Multicast Group Encoding: 1
    Static Multicast Source Encoding: 2
    Static Multicast CMIM Encoding: 3

L2VPN MAC Aging Encoding: 65
    L2VPN MAC Aging Mode: 1

Management Event Control Encoding: 66

Subscriber Mgmt CPE IPv6 List: 67

Default Upstream Target Buffer Configuration: 68

MAC Address Learning Control Encoding: 69
    MAC Address Learning Control: 1
    MAC Address Learning Holdoff Timer: 2

Upstream Aggregate Service Flow (ASF): 70
    Service Flow Reference: 1
    ASF QoS Profile Name: 4
    MESP Reference: 37
    Service Flow Matching Criteria: 38
        Service Flow to ASF Matching by Application Id: 1
        Service Flow to ASF Matching by Service Class Name: 2
        Service Flow to ASF Matching by Traffic Priority Range: 3

Downstream Aggregate Service Flow (ASF): 71
    Service Flow Reference: 1
    ASF QoS Profile Name: 4
    MESP Reference: 37
    Service Flow Matching Criteria: 38
        Service Flow to ASF Matching by Application Id: 1
        Service Flow to ASF Matching by Service Class Name: 2
```

```
        Service Flow to ASF Matching by Traffic Priority Range: 3

Metro Ethernet Service Profile (MESP) Encoding: 72
    MESP Reference: 1
    MESP Bandwidth Profile (MESP-BP): 2
        MESP-BP Committed Information Rate: 1
        MESP-BP Committed Burst Size: 2
        MESP-BP Excess Information Rate: 3
        MESP-BP Excess Burst Size: 4
        MESP-BP Coupling Flag: 5
        MESP-BP Color Mode: 6
            MESP-BP-CM Color Identification Field: 1
            MESP-BP-CM Color Identification Field Value: 2
        MESP-BP Color Marking: 7
            MESP-BP-CR Color Marking Field: 1
            MESP-BP-CR Color Marking Field Value: 2
    MESP Name: 3

Network Timing Profile: 73
  Network Timing Profile Reference: 1
  Network Timing Profile Name: 2

Energy Management Parameter Encoding: 74
    Energy Management Feature Control: 1
    1x1 Mode Encodings: 2
        Downstream Activity Detection Parameters: 1
            Downstream Entry Bitrate Threshold: 1
            Downstream Entry Time Threshold: 2
            Downstream Exit Bitrate Threshold: 3
            Downstream Exit Time Threshold: 4
        Upstream Activity Detection Parameters: 2
            Upstream Entry Bitrate Threshold: 1
            Upstream Entry Time Threshold: 2
            Upstream Exit Bitrate Threshold: 3
            Upstream Exit Time Threshold: 4
    Energy Management Cycle Period: 3

CM Upstream AQM Disable: 76

Unidirectional (UNI) Control Encodings: 79
    Context CMIM: 1
    UNI Admin Status: 2
    UNI Auto-Negotiation Status: 3
    UNI Operating Speed: 4
    UNI Duplex: 5
    EEE Status: 6
    Maximum Frame Size: 7
    Power Over Ethernet (PoE) Status: 8
    Media Type: 9
```

```
Manufacturer CVC Chain: 81

Co-signer CVC Chain: 82

eRouter Configuration Encodings: 202
    eRouter Initialization Mode Encoding: 1
    eRouter TR-069 Management Server: 2
        EnableCWMP: 1
        URL: 2
        Username: 3
        Password: 4
        ConnectionRequestUsername: 5
        ConnectionRequestPassword: 6
        ACSOverride: 7
    eRouter Initialization Mode Override: 3
    Router Advertisement (RA) Transmission Interval: 10
    eRouter SNMP MIB Object: 11
    eRouter Topology Mode Encoding: 42
    eRouter Vendor Specific Information: 43
    eRouter SNMPv1v2c Coexistence Configuration: 53 (see SNMPv1v2c Coexistence Configuration for
    eRouter SNMPv3 Access View Configuration: 54 (see SNMPv3 Access View COnfiguration for subenc

eSTB Configuration: 217
    eSTB IP Mode Control: 1
    eSTB SNMP MIB Object: 11
    eSTB SNMPv3 Notification Receiver: 38 (see DOCSIS V3 Notification Receiver for subencodings)
    eSTB Home Network Prefix Validation: 39
        Instance Number: 1
        Prefix Usage: 2
        IP Address Version: 3
        IPv4 Prefix Length: 4
        IPv4 Subnet Address: 5
        IPv6 Prefix Length: 6
        IPv6 Network Address: 7
    eSTB SEB Server Enable TLS Cipher Suites: 40
    eSTB Vendor Specific Information: 43
    eSTB SNMPv1v2c Coexistence Configuration: 53 (see SNMPv1v2c Coexistence Configuration for su
    eSTB SNMPv3 Access View Configuration: 54 (see SNMPv3 Access View COnfiguration for subencod
```

Part IV
Appendix II

Index

Glossary

D

DOCSIS (Data Over Cable Service Interface Specification) Data Over Cable Service Interface Specification is an international telecommunications standard that permits the addition of high-speed data transfer to an existing cable TV (CATV) system. It is employed by many cable television operators to provide Internet access (see cable Internet) over their existing hybrid fiber-coaxial (HFC) infrastructure.

O

OID (Object Identifier) An object identifier or OID is an identifier used to name an object (compare URN). Structurally, an OID consists of a node in a hierarchically-assigned namespace, formally defined using the ITU-T's ASN.1 standard. Successive numbers of the nodes, starting at the root of the tree, identify each node in the tree.

S

SNMP (Simple Network Management Protocol) Simple Network Management Protocol (SNMP) is an Internet-standard protocol for managing devices on IP networks. Devices that typically support SNMP include routers, switches, servers, workstations, printers, modem racks, and more.â[1] It is used mostly in network management systems to monitor network-attached devices for conditions that warrant administrative attention. SNMP is a component of the Internet Protocol Suite as defined by the Internet Engineering Task Force (IETF). It consists of a set of standards for network management, including an application layer protocol, a database schema, and a set of data objects. SNMP exposes management data in the form of variables on the managed systems, which describe the system configuration. These variables can then be queried (and sometimes set) by managing applications.

T

TLV (Type-Length-Value) Within data communication protocols, optional information may be encoded as a type-length-value or TLV ele-

ment inside of the protocol. The type and length fields are fixed in size (typically 1-4 bytes), and the value field is of variable size. These fields are used as follows:

- Type A binary code, often simply alphanumeric, which indicates the kind of field that this part of the message represents.
- Length The size of the value field (typically in bytes).
- Value Variable-sized series of bytes which contains data for this part of the message.

Some of the advantages of using a TLV representation are:

- TLV sequences are easily searched using generalized parsing functions.
- New message elements which are received at an older node can be safely skipped and the rest of the message can be parsed. This is similar to the way that unknown XML tags can be safely skipped.
- TLV elements can be placed in any order inside the message body.
- TLV elements are typically used in a binary format which makes parsing faster and the data smaller.
- It's fairly easy to generate XML from TLV to make human inspection of the data possible.

.

Bibliography

CableLabs: MAC and Upper Layer Protocols Interface Specification. February 2011 ⟨URL: http://www.cablelabs.com/specifications/CM-SP-MULPIv3.0-I15-110210.pdf⟩