802.3 Management Module Development Process

Doc info

Date	Revision	Author	Notes
2/23/2024	0.0	Peter Jones	Initial draft for review
3/7/2024	0.1	Peter Jones	Updates for Denver March 2024

Contents

Doc info 1
Outline and Basic Principles 2
Principles 2
Tools 3
YANG 3
SNMP 3
Importing to Frame 4
Dealing with diffs 4
References 5
Commands 6
YANG 6
Validate 6
Format 6
Configuration 6
SNMP 6
Validate 6
Format 6
Configuration 6

Outline and Basic Principles

The development process is defined to simplify delivering high quality draft documents and machine-readable source files (e.g., SNMP MIBs, Yang Modules, YANG tree structures etc).

This document is intended to be:

- a living document and will be revised as and when needed.
- used by anyone authoring, reviewing, or editing these files

This may become an “official” 802.3 document at some stage when the “interested parties” are happy that it’s complete enough to be useful.

Principles

The process will follow these principles.

- All editing is done on the machine-readable files (i.e., .mib or .yang).
    - These files will be generically referred to as “source” or “source files” in the remainder of this document.
- All source files must pass the validation checks defined in this document and be warning free (with very limited exceptions) using specified tools/options before being included in a draft.
- All source files need to follow consistent formatting rules, e.g., for the MODULE level clauses. Other items to address include:
    - Indentation
    - White space
    - File format (e.g., unix vs dos)
    - Generic header formats.
- To enforce these formatting rules authors will use the formatting tools defined in this document before submitting files/incorporating files into the draft.
- The editor must import (see Importing to Frame,) not copy/paste, the machine-readable files to Frame before publication.
- Contributors (e.g., authors, commentors, editors) are strongly encouraged to follow this development process and submit source fragments or complete source files as proposed resolutions to comments.
- The published standard is the single source of truth, however 802.3 will ensure that source files accompany every draft and are publicly available from the 802.3 website.
- The “style guides” will be based on current best practice in the industry and will build on top of the IETF guidelines.
- A standard set of validation and formatting tools will be defined.
    - These tools are freely available open source and use the current best practice in the industry.
    - Authors must, and commentors should, ensure that any submissions pass validation and are correctly formatted.

Tools

This document defines common tools and settings used to validate and format source files. In general contributors should always use the latest “stable” release of any tools

YANG

pyang - *https://pypi.org/project/pyang/*

- An extensible YANG (RFC 6020/7950) validator. Provides a framework for plugins that can convert YANG modules to other formats.
- Validation
  - Required options:
    - --canonical: Validate the module(s) according the canonical YANG order
    - --ieee: Validate the module(s) according the “ieee” rules[1].
    - --verbose:
  - The following warnings/exceptions are expected/permitted:
    - warning: RFC 8407: 4.1: the module name should start with the string "ieee-"
- Formatting:
  - Required options:
    - --format yang: output YANG module
    - --yang-canonical: Generate all statements in the canonical order.
    - --yang-remove-unused-imports:
    - --yang-line-length=68:

SNMP

libsmi - https://www.ibr.cs.tu-bs.de/projects/libsmi/

- man page is at https://www.ibr.cs.tu-bs.de/projects/libsmi/libsmi.html
- Validation: smilint
  - Options/arguments
    - May require “preload options” depending on local file structures.
    - This can be done as part of the smilint command or in a “.smirc” file (https://www.ibr.cs.tu-bs.de/projects/libsmi/smi_config.html)
  - Required options:
    - --error-names: Show the error names in braces before error messages.
    - --severity: Show the error severity in brackets before error messages.
    - --level 6: report errors and warnings up to level 6.
    - --ignore namelength-32: ignore warnings about names longer that 32 characters.

[1] Where are these rules defined?

- Formatting: smidump
  - Required options:
    - --error-names: Show the error names in braces before error messages.
    - --severity: Show the error severity in brackets before error messages.
    - --level 6: report errors and warnings up to level 6.
    - --ignore namelength-32: ignore warnings about names longer that 32 characters.
    - --format smiv2: output the MIB in SMIv2 compliant format as defined in RFC 2578, RFC 2579, RFC 2580
    - --format tree: OID registration tree structure of a module

Importing to Frame

<Scott Mansfield knows how to do this>.
The editor uses the Frame "import" command[2] to import the source files into the FrameMaker book.

To make this a little simpler, the source files should be contained in "mib" and "yang" subfolders.

NOTE:
This should probably be done using the "Import unformatted text", "Treat Each Line As A Paragraph" which is described as "Use this option for a line-oriented text file such as a file containing computer code.", but this needs to be investigated by the editors, or those who have FrameMaker.

Dealing with diffs

To simplify the task of communicating the changes to the reviewers, the following procedure will be used.

For a given source file type:

1. Concatenate all the files together (e.g., "cat *.mib > ieee802_3_combined.mib) for the previous and current source file set.
2. Open Microsoft Work.
3. Use the Review->Compare function.
   a. Open the old and new concatenated files.
   b. Save the "compared document" which will have change bars or full revision marking as desired.
   c. Print the file to PDF and it will retain the revision marking. NOTE: save as PDF loses the revision marking.
4. Distribute the PDF with revision marks along with the source files as part of the draft package.

[2] https://help.adobe.com/en_US/framemaker/2019/using/using-framemaker-2019/user-guide/frm_text_tx-import-formatted-unformatted-text.html#topic_import-text

References

1. IETF Guidelines documents:
    a. SNMP
        i. Review tools - https://wiki.ietf.org/group/ops/mib-review-tools
        ii. RFC 4181 - Guidelines for Authors and Reviewers of MIB Documents (https://datatracker.ietf.org/doc/rfc4181/)
    b. YANG
        i. Review tools - https://wiki.ietf.org/group/ops/yang-review-tools
        ii. RFC 8407 - Guidelines for Authors and Reviewers of Documents Containing YANG Data Models (https://datatracker.ietf.org/doc/rfc8407/)
2. IEEE 802 procedures
    a. IEEE 802 Yangsters group (https://1.ieee802.org/yangsters/)

Commands

YANG

Validate

```
COMMAND:
      pyang --canonical --ieee –verbose <name>.yang
```

Format

```
COMMAND:
      pyang -o <name>_pretty.yang -f yang --yang-canonical --yang-remove-
      unused-imports --yang-line-length=68 <name>.yang
```

Configuration

```
YANG_MODPATH:
      <yang_path>/standard/iana: <yang_path>/standard/ietf:
      <yang_path>/standard/ieee/published/802:
      <yang_path>/standard/ieee/published/802.1
```

SNMP

Validate

```
COMMAND:
      smilint --error-names --severity --level 6 --ignore=namelength-
      32-object --ignore=group-membership <name>.mib
```

Format

```
COMMANDS:
      smidump --format tree --error-names --severity --level 6 <name>.mib
      smidump --format smiv2 --error-names --severity --level 6 <name>.mib
```

Configuration

```
      # $HOME/.smirc
      path: <snmp_path>/RFC
      path: <snmp_path>/Standard/ieee/802.1/standard

      # preload some basic modules
      load SNMPv2-SMI
      load SNMPv2-TC
      load SNMPv2-CONF
      load LLDP-V2-TC-MIB
      load LLDP-V2-MIB.mib
```