

 WoojungHuh / 802dot1-yangster-faq 



 Code  Issues  Pull requests  Actions  Projects  Security  Insights 

 wip-draft-0.2 ▾

802dot1-yangster-faq / YANGster-FAQ.md 



WoojungHuh Update YANGster-FAQ.md 

843c638 · 1 minute ago



99 lines (98 loc) · 7.16 KB

YANGsters FAQ (draft 0.2)

Introduction to YANG

- YANG overview
 - YANG is a data modeling language used to model configuration data, state data, Remote Procedure Calls, and notifications for network management protocols. YANG has been used or proposed to be used for other protocols (e.g., RESTCONF and the Constrained Application Protocol (CoAP) Management Interface (CoMI). Further, encodings other than XML have been proposed (e.g., JSON). [RFC7950](#)
 - YANG version 1.1
 - YANG version 1.1 is a maintenance release of the YANG language, addressing ambiguities and defects in the original specification [RFC6020](#) of version 1.
 - Additional changes are introduced in version 1.1 and also some changes are not backward compatible with version 1.
- YANG data model
 - Not explicitly defined. Data model is defined in the terminology section of IETF RFC 7950 as: "data model: A data model describes how data is represented and accessed."
- YANG module
 - RFC7950 Section 3: "module: A YANG module defines hierarchies of schema nodes. With its definitions and the definitions it imports or includes from elsewhere, a module is self-contained and "compilable".

- RFC7950 Section 4.1: "YANG structures data models into modules and submodules."
- RFC7950 Section 4.2.1: "YANG data models are defined in modules. A module contains a collection of related definitions."
- RFC7950 Section 5.2: "YANG modules and submodules are typically stored in files, one "module" or "submodule" statement per file."

802.1 YANG Guidelines

- YANG Terms
- YANG Usage Guidelines
 - 802.1Qcp-2018 - Bridges and Bridged Networks-Amendment 30: YANG Data Model
 - Interface Management (IETF RFC 8343) Model
 - Generic IEEE 802.1Q-2018 bridge model
- YANG Module
- YANG Module Naming Conventions
- ...

YANGster YANG Guidelines

- IEEE 802.1 Editor's Guidelines
 - [Editor's Resources](#) (password protected – 802.1 private area password)
- Module Naming Conventions
 - [YANG Module Naming Guidelines](#)
- IEEE URN
 - [URN\(Uniform Resource Names\)](#)
- YANG Prefix
- Abbreviations or Acronyms
- YANG Copyright text in the YANG module
 - Insert the following at the of the description statement associated with the module.

Copyright (C) IEEE (< <year> >).

All rights reserved. This version of this YANG module is part of IEEE Std 802.1< <std number> >; see the standard itself for full legal notices.

- YANG Copyright text in the standard
 - Add as footnote to module sub-clause

Copyright release for YANG: Users of this standard may freely reproduce the YANG modules contained in this standard so that they can be used for their intended purpose.

- YANG Revision Numbering
 - Current practice is to import without specific revision number for imported modules and to continue the discussion when the IETF's work on YANG packages and Module Versioning completes. [IETF Updated YANG Module Revision Handling](#)
- YANGsters Do's and Don'ts
 - Consider interaction with other models
 - YANG models are rarely used alone but are meant to be mixed and matched, so consider potentially unwanted interaction! In particular, just loading another YANG model should not invalidate previously valid instance data.
 - Care must be taken to ensure augmentations are done in such a way that the capabilities being added by the augmentation can be ignored. Leverage features, presence containers or rearrange the must statements to avoid the issue.
 - See [yangsters-smansfield-axdz-augmentation-issue-0524-v01](#) for an example.
 - Overeager must statements
 - "must" statements in currently unused branches are often evaluated as false, implicitly forcing the existence of certain leafs far down in unused branches. Consider to use `not(count(/gate-control-entry) > ../supported-list-max)` instead of `count(/gate-control-entry) <= ../supported-list-max`

Deviation Guidelines

- Difference between other YANG standards and YANGsters
 - WIP
- Not supported in YANGsters
 - WIP

YANGster YANG tools and links

- YANG Catalog GitHub Repository
 - <https://github.com/YangModels/yang>
- YANG Pretty Printer
 - This is a tool that is currently under development, the preferred tool is currently pyang (see Tool – pyang) below.

- <https://www.ieee802.org/1/files/public/docs2020/new-specht-yang-pretty-printing-0620-v05.pdf>
- <https://www.ieee802.org/1/files/public/docs2020/new-specht-yang-pretty-printer-live-1120-v01.pdf>

- pyang

- An open source tool that is essential to the YANG development lifecycle. The tool can be [downloaded](#) and installed on Windows/Linux.
- Options used for checking

```
-verbose: more warning and error information provided
-canoncial: Validate usual YANG order as defined by RFC 7950
-ieee: Validates with ieee options, there is an issue with the
checking for module name in the tool. pyang complains that all
ieee modules should start with ieee and not ieee802. This should
be fixed at some point.
-max-identifier-length: set a value for how long the longest yang
identifier can be
-path: search path for modules (multiple -path parameters are
allowed)
-f tree -tree-print-groupings: using those two options will
produce a tree and also print out the groupings in tree format. If
you don't want the groupings leave that option off.
```

- Current preferred pretty printing settings

```
pyang -o <pretty-file> -f yang --yang-canonical --yang-remove-
unused-imports --yang-remove-comments --yang-line-length=68
<original-file>
```



wip-draft-0.2 ▾

802dot1-yangster-faq / YANGster-FAQ.md

↑ Top

Preview

Code

Blame

Raw



- Examples

- <https://1.ieee802.org/wp-content/uploads/2020/11/psfs-xml.txt>
- <https://1.ieee802.org/wp-content/uploads/2020/11/psfs-out.txt>
- <https://1.ieee802.org/yangsters/yang-instance-examples/>

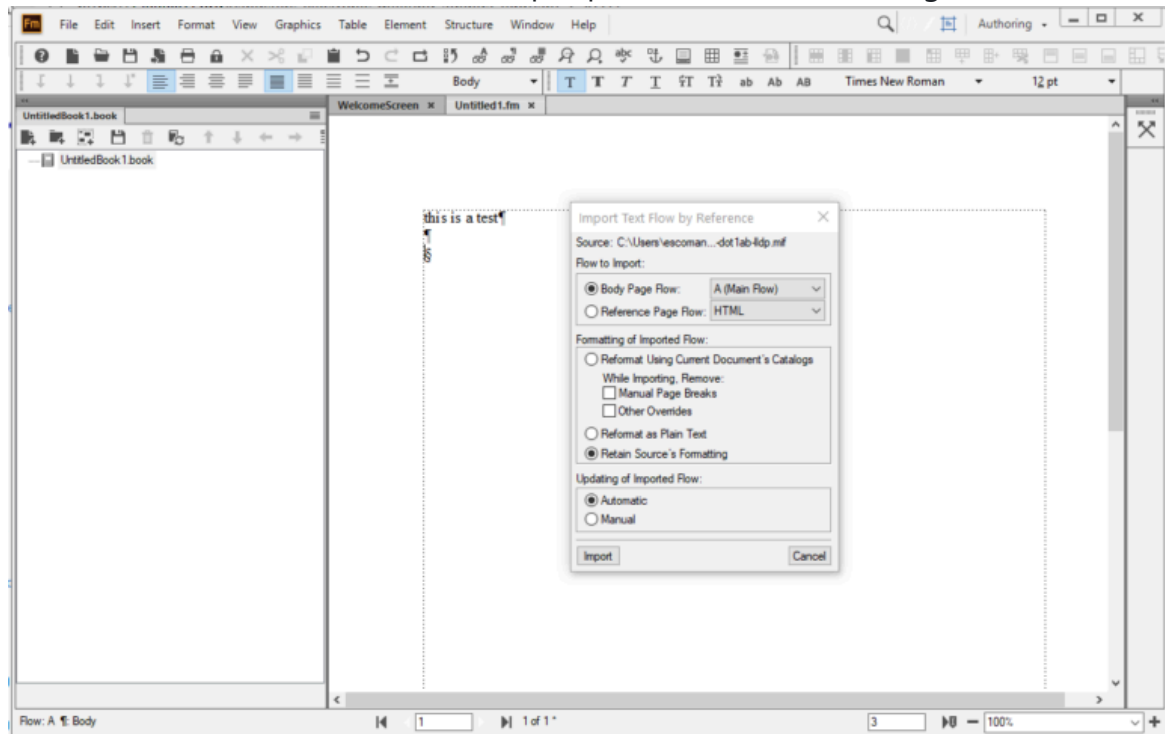
- Online tools

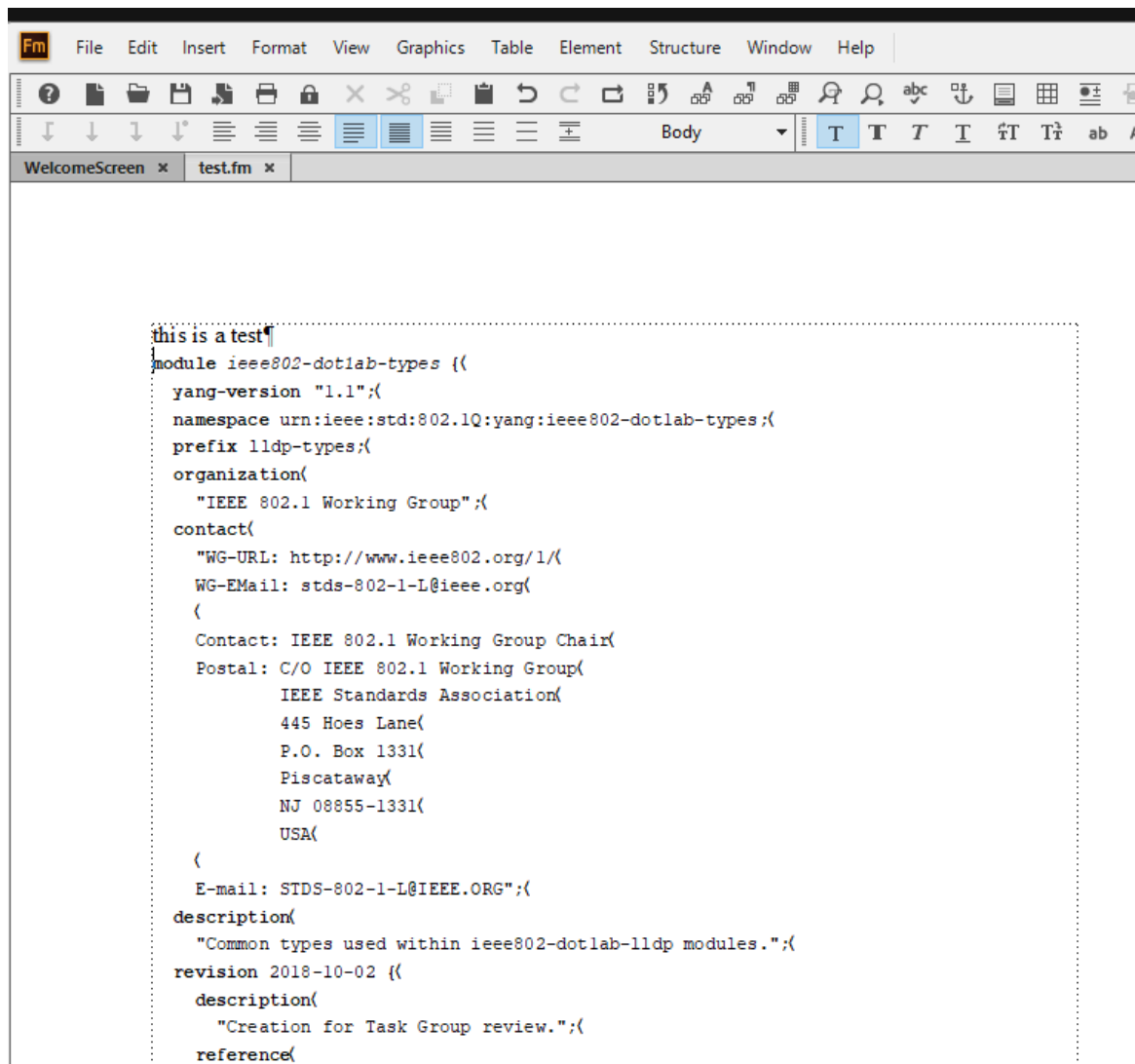
- WIP

- Framemaker and YANG files

- YANG file handling in framemaker
 - a. Click in the frame where you want file to be inserted
 - b. File->Import->File

- c. Import Text Flow by Reference dialog pops up
- d. Change "Formatting of Imported Flow:" to "Retain Source's Formatting"
- e. Click Import
- o The module is inserted and will be kept up to date if the file changes





The screenshot shows a text editor window with a menu bar (File, Edit, Insert, Format, View, Graphics, Table, Element, Structure, Window, Help) and a toolbar. The editor has two tabs: 'WelcomeScreen' and 'test.fm'. The main text area contains a YANG module definition for 'ieee802-dot1ab-types'. The text is as follows:

```
this is a test
module ieee802-dot1ab-types {
  yang-version "1.1";
  namespace urn:ieee:std:802.1Q:yang:ieee802-dot1ab-types;
  prefix lldp-types;
  organization(
    "IEEE 802.1 Working Group";
  contact(
    "WG-URL: http://www.ieee802.org/1/"
    WG-EMail: stds-802-1-L@ieee.org(
      (
        Contact: IEEE 802.1 Working Group Chair(
        Postal: C/O IEEE 802.1 Working Group(
          IEEE Standards Association(
            445 Hoes Lane(
              P.O. Box 1331(
                Piscataway(
                  NJ 08855-1331(
                    USA(
              (
                E-mail: STDS-802-1-L@IEEE.ORG";(
      description(
        "Common types used within ieee802-dot1ab-lldp modules.";
    revision 2018-10-02 {(
      description(
        "Creation for Task Group review.";
    reference(
```

YANG Examples

- Sample YANG file(s) and tool usages