

YANG

Sunday, September 11, 2016 5:03 PM

<https://github.com/CESNET/netopeer/issues/120> Very Very Very Important.

http://dld.netconfcentral.org/doc/ncorg_netconf_yang_tutorial.pdf

<http://www.yang-central.org/twiki/bin/view/Main/YangTutorials>

http://www.w3schools.com/xsl/xpath_syntax.asp -All about XPATH

Basic YANG module interfaces yang-version 1; namespace "some-unique-URI"; prefix import foo-module { prefix foo; include bar-submodule; organization "your organization/company name here"; contact "your name and email address here"; description "Module summary here"; reference "Module references here"; revision { description "Initial version."; 2008-03-05 <data model definitions here, in any order> Module Structure module-stmt module module-llan-e yang-version / ; namespace S'uri-string prefix "prefix-string" // zero or more imports import "module-natne" { prefix "prefix-string" revision-date "dale-string" // zæro or more includes include "submodule-nmne" rcwision-date "dare-string mga n iza tion •org-name ' contact 'c'onluca' description "module description Merenee ' •module reference // zero Or more revision identifiers "date-string" rev ISlon description 'revision description ' reference "revision rejérence bold = mandatory plain = optional black = keywords green = your text NETCONF/YANG Tutorial (c) 2014 YumaWorks, Inc // zero Or more 50



ncorg_net...

Reference file attached.

Screen clipping taken: 9/15/2016 6:23 PM

Screen clipping taken: 9/15/2016 6:59 PM

YANG Statements All statements have the same structure statement = keyword [argument] (";" / "(" "statement ")") any statement can terminate (";") or start a new section of sub-statements ("(" ")") YANG allows external statements to be defined - All standard statements begin with a keyword without a prefix container - All external statements begin with a keyword with a prefix acme:container The prefix is for the module containing the "extension" statement that defines the external statement All YANG tools are required to skip over an unsupported external statement

• YANG:

- 2 forward slashes (//) used to indicate rest of line is a comment

// Implementation of the Ethernet History group is optional.
// Consult the MODULE-COMPLIANCE macro for the authoritative
// conformance information for this MIB.

- C style multi-line comments also allowed

/* Implementation of the Ethernet History group is optional.
Consult the MODULE-COMPLIANCE macro for the authoritative
conformance information for this MIB.
*/

Screen clipping taken: 9/15/2016 6:23 PM

YANG String Encoding Strings can be specified in several ways: Double Quoted String: whitespace allowed, but it is adjusted (escape character sequences replaced, whitespace trimmed) » Single Quoted String: whitespace allowed, and is preserved (this form is safest for pattern strings) Unquoted String: no whitespace allowed (Language tokens

like and '}' are not allowed.) Strings can be specified in fragments if desired: » "foobarbaz" is the same as "foo" + "bar" + "baz" Mixing forms is allowed: "foo" + 'bar' + 'baz'

Screen clipping taken: 9/15/2016 6:23 PM

Identifiers in YANG Modules type int32; Built-in keywords and type names are not imported. No prefix is allowed when they are used. type fooType; OR type foo:fooType Local definitions such as type names and groupings are not imported. The current module prefix is allowed, but not required. From a different submodule include footypes; type fooType; OR type foo:fooType An include statement must be present. The current module prefix is allowed, but not required. import bartypes { prefix bar; } rom a different module type bar:barType; An import statement must be present, with a unique prefix. The declared module prefix must be present when identifiers are used. NETCONF/YANG Tutorial (c) 2014 YumaWorks, Inc.

Screen clipping taken: 9/15/2016 6:24 PM

YANG Module Revisions • Every YANG module has a revision most recent revision statement used (date string) revision 2014-07-01 { description "Initial release"; reference "RFC 6241"; - It there are no revision statements, then the module has the 'NULL' revision Only first version allowed! Must add a revision statement to update the module! • An explicit revision of an external module can be imported import foo { prefix foo; revision-date 2014-07-01;

A YANG module can be split up into multiple files, called "submodules" Similar syntax to module submodule example-foo-subl { belongs-to example-foo { prefix foo; } Each submodule contributes definitions to the module namespace The "include" statement is used within the module or a submodule to import definitions from another submodule Definitions in the module are not visible in any submodule

Screen clipping taken: 9/15/2016 7:00 PM

• YANG: "identity" statement Declares an identityref leaf or leaf-list value - Identity names are scoped by the module containing the identity

Refer the attached file for remaining information.

YANG Body Stater

- **extension**: external statement definition
 - **feature**: YANG feature definition
 - **identity**: YANG identity definition
 - **typedef**: reusable simple type definition
 - **grouping**: reusable set of data definitions
 - **augment**: add data definitions to another data definition (same or other module)
 - **rpc**: protocol operation definition
 - **notification**: notification event definition
-
- All body statements are exported except "augment"

```
extension extension-name {
  argument argument-name {
    yin-element true|false;
  }
}

feature feature-name {
  if-feature feature-name; // zero or more
  status current|deprecated|obsolete;
  description "description text";
  reference "reference text";
}

identity identity-name {
  base identity-base; // no base
  status current|deprecated|obsolete;
  description "description text";
  reference "reference text";
}

typedef type-name {
  type-stmt;
  units "units name";
  default "default value";
  status current|deprecated|obsolete;
  description "description text";
  reference "reference text";
}

grouping grouping-name {
  status current|deprecated|obsolete;
  description "description text";
  reference "reference text";
  typedef-stmt; // zero or more
  grouping-stmt; // zero or more
  data-def-stmt; // zero or more
}
```

NETCONF/YANG Tutorial

Screen clipping taken: 9/11/2016 5:03 PM

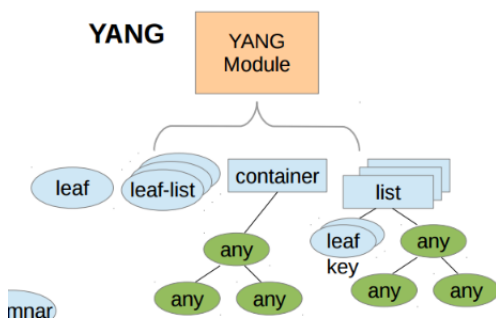
YANG XPath validation explicitly separates the configuration and non-configuration data nodes

• YANG

- mandatory "true":

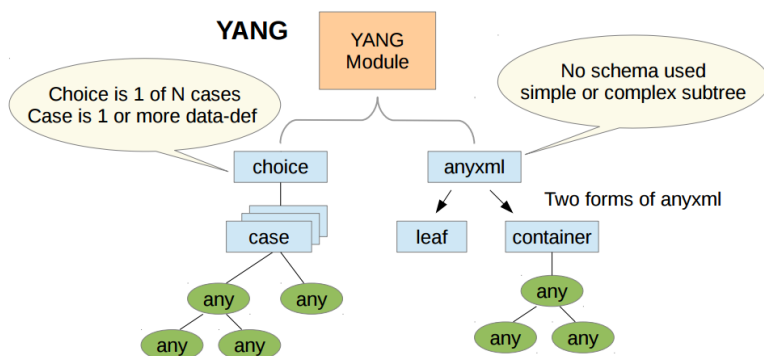
- **<rpc> input:**
parameter node **MUST** be present
- **<notification>:**
node **MUST** be present unless filtering and access control causes deletion of node
- **Configuration data:**
node **MUST** be present in a valid configuration datastore
- **Non-configuration data:**
node **MUST** be present in an **<rpc-reply>** for **<get>**, **<get-config>** unless filtering and access control causes omission of node

http://dld.netconfcentral.org/doc/ncorg_netconf_yang_tutorial.pdf

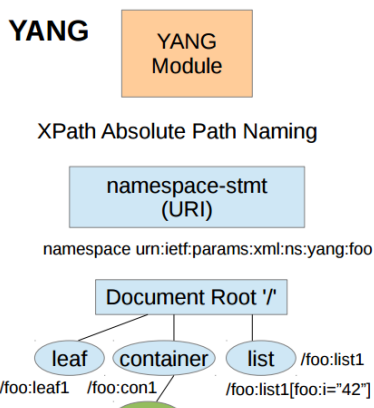


Screen clipping taken: 9/14/2016 10:55 AM

More Data Building Blocks



Screen clipping taken: 9/14/2016 12:47 PM

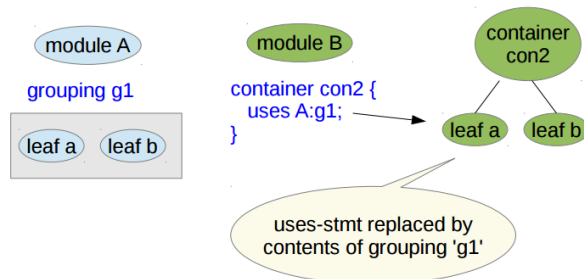


any key i leaf x
 /foo:con1/foo:leaf1
 /foo:list1[foo:i="42"]/foo:x

Data Naming

Data Reuse in YANG

- NETCONF allows data to be defined in reusable groupings
- **grouping-stmt** defines abstract group of objects
 - no namespace assigned to objects in the grouping
- **uses-stmt** defines a concrete expansion point for a grouping
 - given the namespace of the module containing the uses-stmt



Main RFCs

- NETCONF Protocol (RFC 6241) <http://www.ietf.org/rfc/rfc6241.txt>
- YANG Data Modeling Language (RFC 6020) <http://www.ietf.org/rfc/rfc6020.txt>