

Internet Engineering Task Force	A. Bierman	
Internet-Draft	InterWorking Labs	
Intended status: Informational	April 20, 2010	
Expires: October 22, 2010		

[TOC](#)

Guidelines for Authors and Reviewers of YANG Data Model Documents draft-ietf-netmod-yang-usage-04

Abstract

This memo provides guidelines for authors and reviewers of standards track specifications containing YANG data model modules. Applicable portions may be used as a basis for reviews of other YANG data model documents. Recommendations and procedures are defined, which are intended to increase interoperability and usability of NETCONF implementations which utilize YANG data model modules.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 22, 2010.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as

described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction
2.	Terminology
2.1.	Requirements Notation
2.2.	NETCONF Terms
2.3.	YANG Terms
2.4.	Terms
3.	General Documentation Guidelines
3.1.	Module Copyright
3.2.	Narrative Sections
3.3.	Definitions Section
3.4.	Security Considerations Section
3.5.	IANA Considerations Section
3.5.1.	Documents that Create a New Name Space
3.5.2.	Documents that Extend an Existing Name Space
3.6.	Reference Sections
3.7.	Copyright Notices
3.8.	Intellectual Property Section
4.	YANG Usage Guidelines
4.1.	Module Naming Conventions
4.2.	Identifiers
4.3.	Defaults
4.4.	Conditional Statements
4.5.	Lifecycle Management
4.6.	Header Contents
4.7.	Temporary Namespace Assignments
4.8.	Top Level Database Objects
4.9.	Data Types
4.10.	Reusable Type Definitions
4.11.	Object Definitions
4.12.	Operation Definitions
4.13.	Notification Definitions
5.	IANA Considerations
6.	Security Considerations
7.	Acknowledgments
8.	References
8.1.	Normative References
8.2.	Informative References
Appendix A.	Module Review Checklist
Appendix B.	YANG Module Template
Appendix C.	Change Log
C.1.	Changes from 03 to 04
C.2.	Changes from 02 to 03

- [C.3.](#) Changes from 01 to 02
 - [C.4.](#) Changes from 00 to 01
 - [S](#) Author's Address
-

1. Introduction

[TOC](#)

The standardization of network configuration interfaces for use with the [NETCONF \(Enns, R., "NETCONF Configuration Protocol," December 2006.\)](#) [RFC4741] protocol requires a modular set of data models, which can be reused and extended over time. This document defines a set of usage guidelines for standards track documents containing [YANG \(Bjorklund, M., "YANG - A data modeling language for NETCONF," April 2010.\)](#) [I-D.ietf-netmod-yang] data models. It is similar to the MIB usage guidelines specification [\[RFC4181\] \(Heard, C., "Guidelines for Authors and Reviewers of MIB Documents," September 2005.\)](#) in intent and structure. Many YANG constructs are defined as optional to use, such as the description clause. However, in order to maximize interoperability of NETCONF implementations utilizing YANG data models, it is desirable to define a set of usage guidelines which may require a higher level of compliance than the minimum level defined in the YANG specification. This document defines usage guidelines related to the NETCONF operations layer, and NETCONF content layer, as defined in [\[RFC4741\] \(Enns, R., "NETCONF Configuration Protocol," December 2006.\)](#).

2. Terminology

[TOC](#)

2.1. Requirements Notation

[TOC](#)

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 [\[RFC2119\] \(Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," March 1997.\)](#). RFC 2119 language is used here to express the views of the NETMOD working group regarding YANG module content. Yang modules complying with this document will treat the RFC 2119 terminology as if it were describing best current practices.

2.2. NETCONF Terms

[TOC](#)

The following terms are defined in [\[RFC4741\] \(Enns, R., "NETCONF Configuration Protocol," December 2006.\)](#) and are not redefined here:

*capabilities

*client

*operation

*server

2.3. YANG Terms

[TOC](#)

The following terms are defined in [\[I-D.ietf-netmod-yang\] \(Bjorklund, M., "YANG - A data modeling language for NETCONF," April 2010.\)](#) and are not redefined here:

*data node

*module

*namespace

*submodule

*version

Note that the term 'module' may be used as a generic term for a YANG module or submodule. When describing properties which are specific to submodules, the term 'submodule' is used instead.

2.4. Terms

[TOC](#)

The following terms are used throughout this document:

published: A stable release of a module or submodule, usually contained in an RFC.

unpublished: An unstable release of a module or submodule, usually contained in an Internet Draft.

3. General Documentation Guidelines

[TOC](#)

YANG data model modules under review are likely to be contained in Internet Drafts. All guidelines for Internet Draft authors MUST be followed. These guidelines are available online at:

<http://www.rfc-editor.org/rfc-editor/instructions2authors.txt>

The following sections MUST be present in an Internet Draft containing a module:

- *YANG data model boilerplate section

- *Narrative sections

- *Definitions section

- *Security Considerations section

- *IANA Considerations section

- *References section

3.1. Module Copyright

[TOC](#)

The module description statement MUST contain the latest approved IETF Trust Copyright statement, which is available on-line, in section 4 of the Trust Legal Provisions (TLP) document, at:

<http://trustee.ietf.org/license-info/>

Each YANG module or submodule contained within an Internet Draft or RFC MUST be identified as a 'Code Component'. The strings '<CODE BEGINS>' and '<CODE ENDS>' SHOULD be used to identify each Code Component.

3.2. Narrative Sections

[TOC](#)

The narrative part MUST include an overview section that describes the scope and field of application of the module(s) defined by the specification and that specifies the relationship (if any) of these modules to other standards, particularly to standards containing other module modules. The narrative part SHOULD include one or more sections

to briefly describe the structure of the modules defined in the specification.

If the module(s) defined by the specification import definitions from other modules (except for those defined in the [YANG \(Bjorklund, M., "YANG - A data modeling language for NETCONF," April 2010.\)](#)

[I-D.ietf-netmod-yang] or [YANG Types \(Schoenwaelder, J., "Common YANG Data Types," April 2010.\)](#) [I-D.ietf-netmod-yang-types] documents) or are always implemented in conjunction with other modules, then those facts MUST be noted in the overview section, as MUST any special interpretations of objects in other modules.

3.3. Definitions Section

[TOC](#)

This section contains the module(s) defined by the specification. These modules MUST be written in YANG [\[I-D.ietf-netmod-yang\] \(Bjorklund, M., "YANG - A data modeling language for NETCONF," April 2010.\)](#).

See [Section 4 \(YANG Usage Guidelines\)](#) for guidelines on YANG usage.

3.4. Security Considerations Section

[TOC](#)

Each specification that defines one or more modules MUST contain a section that discusses security considerations relevant to those modules. This section MUST be patterned after the latest approved template (available at <http://www.ops.ietf.org/yang-security.html>). In particular, writable module objects that could be especially disruptive if abused MUST be explicitly listed by name and the associated security risks MUST be spelled out; similarly, readable module objects that contain especially sensitive information or that raise significant privacy concerns MUST be explicitly listed by name and the reasons for the sensitivity/privacy concerns MUST be explained.

3.5. IANA Considerations Section

[TOC](#)

In order to comply with IESG policy as set forth in <http://www.ietf.org/ID-Checklist.html>, every Internet-Draft that is submitted to the IESG for publication MUST contain an IANA Considerations section. The requirements for this section vary depending what actions are required of the IANA.

3.5.1. Documents that Create a New Name Space

[TOC](#)

If an Internet-Draft defines a new name space that is to be administered by the IANA, then the document MUST include an IANA Considerations section, that specifies how the name space is to be administered.

Specifically, if any YANG module namespace statement value contained in the document is not already registered with IANA, then a new YANG Namespace registry entry must be requested from the IANA. The YANG specification includes the procedure for this purpose in its IANA Considerations section.

3.5.2. Documents that Extend an Existing Name Space

[TOC](#)

It is possible to extend an existing namespace using a YANG submodule which belongs to an existing module already administered by IANA. In this case, the document containing the main module MUST be updated to use the latest revision of the submodule.

3.6. Reference Sections

[TOC](#)

For every import or include statement which appears in a module contained in the specification, which identifies a module in a separate document, a corresponding normative reference to that document MUST appear in the Normative References section. The reference MUST correspond to the specific module version actually used within the specification.

For every reference statement which appears in a module contained in the specification, which identifies a separate document, a corresponding normative reference to that document SHOULD appear in the Normative References section. The reference SHOULD correspond to the specific document version actually used within the specification.

3.7. Copyright Notices

[TOC](#)

The proper copyright notices MUST be present in the module description statement. Refer to the IETF Trust Legal Provision for the exact legal text that needs to be included.

3.8. Intellectual Property Section

[TOC](#)

The proper IPR statements MUST be present in the document, according to the most current Internet Draft boilerplate. Refer to the IETF Trust Legal Provision for the exact legal text that needs to be included.

4. YANG Usage Guidelines

[TOC](#)

In general, modules in IETF standards-track specifications MUST comply with all syntactic and semantic requirements of YANG.

[\[I-D.ietf-netmod-yang\]](#) (Bjorklund, M., "YANG - A data modeling language for NETCONF," April 2010.). The guidelines in this section are intended to supplement the YANG specification, which is intended to define a minimum set of conformance requirements.

In order to promote interoperability and establish a set of practices based on previous experience, the following sections establish usage guidelines for specific YANG constructs.

Only guidelines which clarify or restrict the minimum conformance requirements are included here.

4.1. Module Naming Conventions

[TOC](#)

Modules contained in standards track documents SHOULD be named according to the guidelines in the IANA considerations section of [\[I-D.ietf-netmod-yang\]](#) (Bjorklund, M., "YANG - A data modeling language for NETCONF," April 2010.).

A distinctive word or acronym (e.g., protocol name or working group acronym) SHOULD be used in the module name. If new definitions are being defined to extend one or more existing modules, then the same word or acronym should be reused, instead of creating a new one.

All published module names MUST be unique.

Once a module name is published, it MUST not be reused, even if the RFC containing the module is reclassified to 'Historic' status.

4.2. Identifiers

[TOC](#)

Identifiers for all published modules, submodules, typedefs, groupings, data objects, operations, and notifications MUST be between 1 and 64 characters in length.

4.3. Defaults

[TOC](#)

In general, it is suggested that sub-statements containing default values SHOULD NOT be present. For example, 'status current;', 'config true;', 'mandatory false;', and 'max-elements unbounded;' are common defaults which would make the module difficult to read if used everywhere they are allowed.

Instead, it is suggested that common statements SHOULD only be used when being set to a value other than the default value.

4.4. Conditional Statements

[TOC](#)

A module may be conceptually partitioned in several ways, using the 'if-feature' and/or 'when' statements.

Data model designers need to carefully consider all modularity aspects, including the use of YANG conditional statements.

Objects SHOULD NOT directly reference NETCONF capabilities, in order to specify optional behavior. Instead, a 'feature' statement SHOULD be defined instead of a NETCONF capability, and the 'if-feature' statement SHOULD be used within the optional object definition.

If the condition associated with the desired semantics is not dependent on any particular instance value within the database, then an 'if-feature' statement SHOULD be used instead of a 'when' statement.

The 'attribute' and 'namespace' axis SHOULD NOT be used because the associated XML node types are not supported in YANG, and may not be supported consistently across NETCONF server implementations.

The 'position' and 'last' functions MAY be used with caution, within a single server implementation. These functions may be useful in some cases when processing user-ordered lists. A server is only required to maintain the XML order of a user-ordered list or leaf-list.

The 'preceding', and 'following' axes SHOULD NOT be used. These constructs rely on XML document order within a NETCONF server configuration database, which may not be supported consistently or produce reliable results across implementations. Predicate expressions based on static node properties (e.g., name, value, ancestors, descendants) SHOULD be used instead.

The 'preceding-sibling' and 'following-sibling' axes MAY be used, with caution. A server is not required to maintain a persistent or deterministic XML document order, which will affect use of these axes. Implicit 'position' function calls within predicates SHOULD NOT be used. (e.g., //chapter[42]).

Data nodes which use the 'int64' and 'uint64' built-in type MAY be used with caution, within relational expressions. There are boundary conditions in which the translation from the YANG 64-bit type to an XPath number can cause incorrect results. Specifically, an XPath double

precision floating point number cannot represent very large positive or negative 64-bit numbers because it only provides a total precision of 53 bits.

Data modelers need to be careful not to confuse the YANG value space and the XPath value space. The data types are not the same in both, and conversion between YANG and XPath data types SHOULD be considered carefully.

Explicit XPath data type conversions MAY be used (e.g., 'string', 'boolean', or 'number' functions), instead of implicit XPath data type conversions.

4.5. Lifecycle Management

[TOC](#)

The status statement SHOULD NOT be present if its value is 'current'. It MUST be present if its value is 'deprecated' or 'obsolete'.

The module or submodule name MUST NOT be changed, once the document containing the module or submodule is published.

The module namespace URI value MUST NOT be changed, once the document containing the module is published.

The revision-date sub-statement (within the imports statement) SHOULD be present if any groupings are used from the external module.

The revision-date sub-statement (within the include statement) SHOULD be present if any groupings are used from the external sub-module.

If submodules are used, then the document containing the main module MUST be updated so that the main module revision date is equal or more recent than the revision date of any submodule which is (directly or indirectly) included by the main module.

4.6. Header Contents

[TOC](#)

For published modules, the namespace MUST be a globally unique URI, as defined in [\[RFC3986\] \(Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier \(URI\): Generic Syntax," January 2005.\)](#).

This value is usually assigned by the IANA.

The organization statement MUST be present. If the module is contained in a document intended for standards-track status, then the organization SHOULD be the IETF working group chartered to write the document.

The contact statement MUST be present. If the module is contained in a document intended for standards-track status, then the working group WEB and mailing information MUST be present, and the document author contact information SHOULD be present. In addition, the Area Director and other contact information MAY be present.

The description statement MUST be present. The appropriate IETF Trust Copyright text MUST be present, as described in [Section 3.1 \(Module Copyright\)](#).

Modules are often extracted from their original documents and it is useful for developers and operators to know how to find the original source document in a consistent manner.

If the module relies on information contained in other documents, which are not the same documents implied by the import statements present in the module, then these documents MUST be identified in the reference statement.

A revision statement MUST be present for each published version of the module. The revision statement MUST have a reference substatement. It MUST identify the published document which contains the module. Each new revision MUST include a revision date which is higher than any other revision date in the module.

It is acceptable to reuse the same revision statement within unpublished versions (i.e., Internet Drafts), but the revision date MUST be updated to a higher value each time the Internet Draft is re-published.

4.7. Temporary Namespace Assignments

[TOC](#)

It is desirable to include only valid YANG modules in documents, whether they are published yet or not. This allows:

- *the module to compile correctly instead of generating disruptive fatal errors.
- *early implementors to use the modules without picking a random value for the XML namespace.
- *early interoperability testing since independent implementations will use the same XML namespace value.

Until a URI is assigned by the IANA, a temporary namespace URI MUST be provided for the namespace statement in a YANG module. A value SHOULD be selected which is not likely to collide with other YANG namespaces. A standard namespace statement value SHOULD have the following form:

```
<URN prefix string>:<module-name>
```

The following URN prefix string SHOULD be used for published and unpublished YANG modules

```
urn:ietf:params:xml:ns:yang:
```

The following example URNs would be valid temporary namespace statement values for standards-track modules:

```
urn:ietf:params:xml:ns:yang:ietf-netconf-partial-lock
```

```
urn:ietf:params:xml:ns:yang:ietf-netconf-state
```

```
urn:ietf:params:xml:ns:yang:ietf-netconf
```

Note that a different URN prefix string SHOULD be used for non-standards track modules. The string SHOULD be selected according to the guidelines in [\[I-D.ietf-netmod-yang\] \(Bjorklund, M., "YANG - A data modeling language for NETCONF," April 2010.\)](#).

The following examples of non-standards track modules are only suggestions. There are no guidelines for this type of URN in this document:

```
http://example.com/ns/example-interfaces
```

```
http://example.com/ns/example-system
```

4.8. Top Level Database Objects

[TOC](#)

There SHOULD only be one top-level data node defined in each YANG module. However, there MAY be more than one if needed.

The top-level data organization SHOULD be considered carefully, in advance. Data model designers need to consider how the functionality for a given protocol or protocol family will grow over time.

The names and data organization SHOULD reflect persistent information, such as the name of a protocol. The name of the working group SHOULD NOT be used because this may change over time.

A mandatory database object is defined as a node that a client must provide for the database to be valid. The server will not provide a value under any conditions.

Top-level database objects MUST NOT be mandatory. If a mandatory node appears at the top-level, it will immediately cause the database to be invalid. This can occur when the server boots or when a module is loaded dynamically at runtime.

4.9. Data Types

[TOC](#)

Selection of an appropriate data type (i.e., built-in type, existing derived type, or new derived type) is very subjective and therefore few requirements can be specified on that subject.

Data model designers SHOULD use the most appropriate built-in data type for the particular application.

If extensibility of enumerated values is required, then the identityref data type SHOULD be used instead of an enumeration or other built-in type.

For string data types, if a machine-readable pattern can be defined for the desired semantics, then one or more pattern statements SHOULD be present.

For string data types, if the length of the string is required to be bounded in all implementations, then a length statement SHOULD be present.

For string data types, object semantics SHOULD NOT rely on preservation of leading and trailing whitespace characters.

For numeric data types, if the values allowed by the intended semantics are different than those allowed by the unbounded intrinsic data type (e.g., int32), then a range statement SHOULD be present.

The signed numeric data types (i.e., 'int8', 'int16', 'int32', and 'int64') SHOULD NOT be used unless negative values are allowed for the desired semantics.

For enumeration or bits data types, the semantics for each enum or bit SHOULD be documented. A separate description statement (within each enum or bit statement) SHOULD be present.

4.10. Reusable Type Definitions

[TOC](#)

If an appropriate derived type exists in any standard module, such as [\[I-D.ietf-netmod-yang-types\] \(Schoenwaelder, J., "Common YANG Data Types," April 2010.\)](#), then it SHOULD be used instead of defining a new derived type.

If an appropriate units identifier can be associated with the desired semantics, then a units statement SHOULD be present.

If an appropriate default value can be associated with the desired semantics, then a default statement SHOULD be present.

If a significant number of derived types are defined, and it is anticipated that these data types will be reused by multiple modules, then these derived types SHOULD be contained in a separate module or submodule, to allow easier reuse without unnecessary coupling.

The description statement MUST be present.

If the type definition semantics are defined in an external document, then the reference statement SHOULD be present.

[TOC](#)

4.11. Object Definitions

The description statement MUST be present in the following body statements:

- *extension
- *feature
- *identity
- *typedef
- *grouping
- *augment
- *rpc
- *notification

The description statement MUST be present in the following data definition constructs:

- *container
- *leaf
- *leaf-list
- *list
- *choice
- *anyxml

If the object semantics are defined in an external document, then a reference statement SHOULD be present.

The 'anyxml' construct MAY be used with caution within configuration data. This may be useful to represent an HTML banner for example. However, this construct SHOULD NOT be used if other YANG data node types can be used instead to represent the desired syntax and semantics.

If there are referential integrity constraints associated with the desired semantics that can be represented with XPath, then one or more must statements SHOULD be present.

For list and leaf-list objects, if the number of possible instances is required to be bounded for all implementations, then the max-elements statements SHOULD be present.

If any must or when statements are used within the object definition, then the object description statement SHOULD describe the purpose of each one.

4.12. Operation Definitions

[TOC](#)

The description statement MUST be present in 'rpc' statements defining new operations.

If the operation semantics are defined in an external document, then a reference statement SHOULD be present.

If the operation impacts system behavior in some way, it SHOULD be mentioned in the description statement.

If the operation is potentially harmful to system behavior in some way, it MUST be mentioned in the Security Considerations section of the document.

4.13. Notification Definitions

[TOC](#)

The description statement MUST be present.

If the notification semantics are defined in an external document, then a reference statement SHOULD be present.

5. IANA Considerations

[TOC](#)

There are no actions requested of IANA at this time.

6. Security Considerations

[TOC](#)

This document defines documentation guidelines for NETCONF content defined with the YANG data modeling language. It does not introduce any new or increased security risks into the management system.

[TOC](#)

7. Acknowledgments

The structure and contents of this document are adapted from [Guidelines for MIB Documents \(Heard, C., "Guidelines for Authors and Reviewers of MIB Documents," September 2005.\)](#) [RFC4181], by C. M. Heard.

8. References

[TOC](#)

8.1. Normative References

[TOC](#)

[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997 (TXT , HTML , XML).
[RFC3986]	Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax," STD 66, RFC 3986, January 2005 (TXT , HTML , XML).
[RFC4741]	Enns, R., "NETCONF Configuration Protocol," RFC 4741, December 2006 (TXT).
[I-D.ietf-netmod-yang]	Bjorklund, M., "YANG - A data modeling language for NETCONF," draft-ietf-netmod-yang-12 (work in progress), April 2010 (TXT).
[I-D.ietf-netmod-yang-types]	Schoenwaelder, J., "Common YANG Data Types," draft-ietf-netmod-yang-types-09 (work in progress), April 2010 (TXT).

8.2. Informative References

[TOC](#)

[RFC4181]	Heard, C., "Guidelines for Authors and Reviewers of MIB Documents," BCP 111, RFC 4181, September 2005 (TXT).
-----------	--

Appendix A. Module Review Checklist

[TOC](#)

This section is adapted from RFC 4181.

The purpose of a YANG module review is to review the YANG module both for technical correctness and for adherence to IETF documentation requirements. The following checklist may be helpful when reviewing a draft document:

1. I-D Boilerplate -- verify that the draft contains the required Internet-Draft boilerplate (see <http://www.ietf.org/ietf/1id-guidelines.txt>), including the appropriate statement to permit publication as an RFC, and that I-D boilerplate does not contain references or section numbers.
2. Abstract -- verify that the abstract does not contain references, that it does not have a section number, and that its content follows the guidelines in <http://www.ietf.org/ietf/1id-guidelines.txt>.
3. IETF Trust Copyright -- verify that the draft contains the latest approved TLP boilerplate as described in [Section 3.1 \(Module Copyright\)](#).
4. Security Considerations Section -- verify that the draft uses the latest approved template from the OPS area web site (<http://www.ops.ietf.org/yang-security.html>) and that the guidelines therein have been followed.
5. IANA Considerations Section -- this section must always be present. For each module within the document, ensure that the IANA Considerations section contains entries for the following IANA registries:

XML Namespace Registry: Register the YANG module namespace.

YANG Module Registry: Register the YANG module name, prefix, namespace, and RFC number, according to the rules specified in [\[I-D.ietf-netmod-yang\] \(Bjorklund, M., "YANG - A data modeling language for NETCONF," April 2010.\)](#).

6. References -- verify that the references are properly divided between normative and informative references, that RFC 2119 is included as a normative reference if the terminology defined therein is used in the document, that all references required by the boilerplate are present, that all YANG modules containing imported items are cited as normative references, and that all citations point to the most current RFCs unless there is a valid reason to do otherwise (for example, it is OK to include an informative reference to a previous version of a specification to help explain a feature included for backward compatibility).
7. Copyright Notices -- verify that the draft contains an abbreviated IETF Trust copyright notice in the description statement of each YANG module or sub-module, and that it contains the full IETF Trust copyright notice at the end of the document. Make sure that the correct year is used in all

copyright dates. Use the approved text from the latest Trust Legal Provisions (TLP) document, which can be found at:

<http://trustee.ietf.org/license-info/>

8. Other Issues -- check for any issues mentioned in <http://www.ietf.org/ID-Checklist.html> that are not covered elsewhere.
9. Technical Content -- review the actual technical content for compliance with the guidelines in this document. The use of a YANG module compiler is recommended when checking for syntax errors; see [YANG tool URL TBD] for more information. Checking for correct syntax, however, is only part of the job. It is just as important to actually read the YANG module document from the point of view of a potential implementor. It is particularly important to check that description statements are sufficiently clear and unambiguous to allow interoperable implementations to be created.

Appendix B. YANG Module Template

[TOC](#)

```

<CODE BEGINS> file "ietf-template.yang"

module iETF-template {

    // replace this string with a unique namespace URN value
    namespace
        "urn:ietf:params:xml:ns:yang:ietf-template:DRAFT-02";

    // replace this string, and try to pick a unique prefix
    prefix "temp";

    // import statements here: e.g.,
    // import iETF-yang-types { prefix yang; }
    // import iETF-inet-types { prefix inet; }

    // identify the IETF working group if applicable
    organization
        "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

    // update this contact statement with your info
    contact
        "WG Web:  <http://tools.ietf.org/wg/your-wg-name/>
        WG List:  <mailto:your-wg-name@ietf.org>

        WG Chair: your-WG-chair
                  <mailto:your-WG-chair@example.com>

        Editor:   your-name
                  <mailto:your-email@example.com>";

    // replace the first sentence in this description statement.
    // replace the copyright notice with the most recent
    // version, if it has been updated since the publication
    // of this document
    description
        "This module defines a template for other YANG modules.

        Copyright (c) 2010 IETF Trust and the persons identified as
        the document authors.  All rights reserved.

        Redistribution and use in source and binary forms, with or
        without modification, is permitted pursuant to, and subject
        to the license terms contained in, the Simplified BSD License
        set forth in Section 4.c of the IETF Trust's Legal Provisions
        Relating to IETF Documents

```

```

(http://trustee.ietf.org/license-info).

This version of this YANG module is part of RFC XXXX; see
the RFC itself for full legal notices.";

// RFC Ed.: replace XXXX with actual RFC number and remove this note
reference "RFC XXXX";

// RFC Ed.: remove this note
// Note: extracted from draft-ietf-netmod-yang-usage-04.txt

// replace YYYY-MM-DD with a real date (year-month-day)
// here is an example revision date: 2009-08-12
revision YYYY-MM-DD {
    description
        "Initial version";
}

// extension statements

// feature statements

// identity statements

// typedef statements

// grouping statements

// data definition statements

// augment statements

// rpc statements

// notification statements

// DO NOT put deviation statements in a published module
}

<CODE ENDS>

```

Figure 1

C.1. Changes from 03 to 04

*Removed figure 1 to reduce duplication, just refer to 4741bis draft.

*Fixed bugs and typos found in WGLC reviews.

*Removed some guidelines and referring to YANG draft instead of duplicating YANG rules here.

*Changed security guidelines so they refer to the IETF Trust TLP instead of MIB-specific references.

*Change temporary namespace guidelines so the DRAFT-XX and RFC-nnnn suffix strings are not used.

*Changed some MIB boilerplate so it refers to YANG boilerplate instead.

*Introduced dangling URL reference to online YANG security guidelines

<http://www.ops.ietf.org/yang-security.html>

Text from Bert Wijnen will be completed soon and posted online, and then this URL will be finalized.

*Moved reference for identifying the source document inside the each revision statement.

*Removed guideline about valid XPath since YANG already requires valid XPath.

*Added guideline that strings should not rely on preservation of leading and trailing whitespace characters.

*Relaxed some XPath and anyxml guidelines from SHOULD NOT or MUST NOT to MAY use with caution.

*Updated the TLP text within the example module again.

*Reversed order of change log so most recent entries are first.

C.2. Changes from 02 to 03

[TOC](#)

- *Updated figure 1 to align with 4741bis draft.
- *Updated guidelines for import-by-revision and include-by-revision.
- *Added file name to code begins convention in ietf-template module.

C.3. Changes from 01 to 02

[TOC](#)

- *Updated figure 1 per mailing list comments.
- *Updated suggested organization to include the working group name.
- *Updated ietf-template.yang to use new organization statement value.
- *Updated Code Component requirements as per new TLP.
- *Updated ietf-template.yang to use new Code Component begin and end markers.
- *Updated references to the TLP in a couple sections.
- *Change manager/agent terminology to client/server.

C.4. Changes from 00 to 01

[TOC](#)

- *Added transport 'TLS' to figure 1.
- *Added note about RFC 2119 terminology.
- *Corrected URL for instructions to authors.
- *Updated namespace procedures section.
- *Updated guidelines on module contact, reference, and organization statements.

*Added note on use of preceding-sibling and following-sibling axes in XPath expressions.

*Added section on temporary namespace statement values.

*Added section on top level database objects.

*Added ietf-template.yang appendix.

Author's Address

[TOC](#)

	Andy Bierman
	InterWorking Labs
Email:	andyb@iwl.com