

Internet Engineering Task Force (IETF)  
Request for Comments: 6991  
Obsoletes: 6021  
Category: Standards Track  
ISSN: 2070-1721

J. Schoenwaelder, Ed.  
Jacobs University  
July 2013

## Common YANG Data Types

### Типы данных YANG общего назначения

#### Тезисы

Этот документ вводит набор типов данных общего назначения для использования в языке моделирования данных YANG. Документ отменяет RFC 6021.

#### Статус документа

Документ относится к категории Internet Standards Track.

Документ является результатом работы IETF<sup>1</sup> и представляет согласованный взгляд сообщества IETF. Документ прошел открытое обсуждение и был одобрен для публикации IESG<sup>2</sup>. Не все одобренные IESG документы претендуют на статус Internet Standard (см. раздел 2 в RFC 5741).

Информацию о текущем статусе документа, ошибках и способах обратной связи можно найти по ссылке <http://www.rfc-editor.org/info/rfc6991>.

#### Авторские права

Авторские права (Copyright (c) 2013) принадлежат IETF Trust и лицам, указанным в качестве авторов документа. Все права защищены.

Этот документ является субъектом прав и ограничений, перечисленных в BCP 78 и IETF Trust Legal Provisions и относящихся к документам IETF (<http://trustee.ietf.org/license-info>), на момент публикации данного документа. Прочтите упомянутые документы внимательно, поскольку в них описаны права и ограничения, относящиеся к данному документу. Фрагменты программного кода, включенные в этот документ, распространяются в соответствии с упрощенной лицензией BSD, как указано в параграфе 4.e документа Trust Legal Provisions, без каких-либо гарантий (как указано в Simplified BSD License).

Документ может содержать материалы из IETF Document или IETF Contribution, опубликованных или публично доступных до 10 ноября 2008 года. Лица, контролирурующие авторские права на некоторые из таких документов, могли не предоставить IETF Trust права разрешать внесение изменений в такие документы за рамками процессов IETF Standards. Без получения соответствующего разрешения от лиц, контролирующих авторские права этот документ не может быть изменен вне рамок процесса IETF Standards, не могут также создаваться производные документы за рамками процесса IETF Standards за исключением форматирования документа для публикации или перевода с английского языка на другие языки.

## Оглавление

1. Введение.....	1
2. Обзор.....	2
3. Базовые производные типы YANG.....	2
4. Связанные с Internet производные типы.....	8
5. Взаимодействие с IANA.....	14
6. Вопросы безопасности.....	14
7. Участники работы.....	15
8. Благодарности.....	15
9. Литература.....	15
9.1. Нормативные документы.....	15
9.2. Дополнительная литература.....	15
Приложение А. Отличия от RFC 6021.....	16

## 1. Введение

YANG [RFC6020] является языком моделирования данных, служащих для создания моделей конфигурации и состояния, которыми манипулирует протокол NETCONF<sup>3</sup> [RFC6241]. YANG поддерживает небольшой набор встроенных типов данных и обеспечивает механизмы создания производных типов на основе встроенных.

В этом документе представлен набор производных типов данных общего назначения, выведенных из встроенных в YANG типов данных. Производные типы рассчитаны на применение в моделировании всех типов данных управления. Определения типов организованы в несколько модулей YANG. Модуль `ietf-yang-types` содержит типы данных общего назначения, а в модуле `ietf-inet-types` содержатся определения, относящиеся к стеку протоколов Internet.

Этот документ добавляет определения новых типов данных и отменяет действие [RFC6021]. Подробности приведены в операторах `revision` модулей YANG (разделы 3 и 4), а различия между документами описаны в Приложении А.

<sup>1</sup>Internet Engineering Task Force.

<sup>2</sup>Internet Engineering Steering Group.

<sup>3</sup>Network Configuration Protocol — протокол настройки конфигурации сети.

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не нужно** (SHALL NOT), **следует** (SHOULD), **не следует** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с BCP 14 [RFC2119].

## 2. Обзор

В этом разделе представлен краткий обзор типов данных, определенных в последующих разделах, и их эквивалентов в SMIv2<sup>1</sup> [RFC2578][RFC2579]. Тип данных YANG считается эквивалентным типу данных SMIv2, если их семантика и набор значений эквивалентны.

В таблице 1 перечислены типы данных, определенные в модуле ietf-yang-types, и соответствующие типы SMIv2 (- указывает отсутствие соответствующего типа в SMIv2).

Таблица 1. Модуль ietf-yang-types.

Тип YANG	Эквивалентный тип SMIv2 (модуль)
counter32	Counter32 (SNMPv2-SMI)
zero-based-counter32	ZeroBasedCounter32 (RMON2-MIB)
counter64	Counter64 (SNMPv2-SMI)
zero-based-counter64	ZeroBasedCounter64 (HCNUM-TC)
gauge32	Gauge32 (SNMPv2-SMI)
gauge64	CounterBasedGauge64 (HCNUM-TC)
object-identifier	-
object-identifier-128	OBJECT IDENTIFIER
yang-identifier	-
date-and-time	-
timeticks	TimeTicks (SNMPv2-SMI)
timestamp	TimeStamp (SNMPv2-TC)
phys-address	PhysAddress (SNMPv2-TC)
mac-address	MacAddress (SNMPv2-TC)
xpath1.0	-
hex-string	-
uuid	-
dotted-quad	-

В таблице 2 перечислены типы данных, определенные в модуле ietf-inet-types, и соответствующие типы SMIv2 (если они имеются).

Таблица 2. Модуль ietf-inet-types.

Тип YANG	Эквивалентный тип SMIv2 (модуль)
ip-version	InetVersion (INET-ADDRESS-MIB)
dscp	Dscp (DIFFSERV-DSCP-TC)
ipv6-flow-label	IPv6FlowLabel (IPV6-FLOW-LABEL-MIB)
port-number	InetPortNumber (INET-ADDRESS-MIB)
as-number	InetAutonomousSystemNumber (INET-ADDRESS-MIB)
ip-address	-
ipv4-address	-
ipv6-address	-
ip-address-no-zone	-
ipv4-address-no-zone	-
ipv6-address-no-zone	-
ip-prefix	-
ipv4-prefix	-
ipv6-prefix	-
domain-name	-
host	-
uri	Uri (URI-TC-MIB)

## 3. Базовые производные типы YANG

Модуль YANG ietf-yang-types ссылается на [IEEE802], [ISO9834-1], [RFC2578], [RFC2579], [RFC2856], [RFC3339], [RFC4122], [RFC4502], [RFC6020], [XPATH], [XSD-TYPES].

```
<CODE BEGINS> file "ietf-yang-types@2013-07-15.yang"

module ietf-yang-types {

  namespace "urn:ietf:params:xml:ns:yang:ietf-yang-types";
  prefix "yang";

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

  contact
    "WG Web: <http://tools.ietf.org/wg/netmod/>
    WG List: <mailto:netmod@ietf.org>

    WG Chair: David Kessens <mailto:david.kessens@nsn.com>
```

<sup>1</sup>Structure of Management Information Version 2 — структура данных управления, версия 2.

WG Chair: Juergen Schoenwaelder  
<mailto:j.schoenwaelder@jacobs-university.de>

Editor: Juergen Schoenwaelder  
<mailto:j.schoenwaelder@jacobs-university.de>;

#### description

"Этот модуль содержит набор производных типов данных YANG общего назначения.

Авторские права (Copyright (c) 2013) принадлежат IETF Trust и лицам, указанным как авторы кода. Все права защищены.

Распространение и использование в исходной и двоичной форме с изменениями или без них разрешается в соответствии с условиями, указанными в упрощенной лицензии BSD, изложенной в разделе 4.c Правового положения IETF Trust применительно к документам IETF (<http://trustee.ietf.org/license-info>).

Эта версия модуля YANG является частью RFC 6991, где правовые аспекты выражены более полно.";

#### revision 2013-07-15 {

##### description

"В этом выпуске добавлены новые типы данных:

- yang-identifier
- hex-string
- uuid
- dotted-quad";

##### reference

"RFC 6991: Common YANG Data Types";

}

#### revision 2010-09-24 {

##### description

"Первый выпуск.";

##### reference

"RFC 6021: Common YANG Data Types";

}

/\*\* Набор типов для счетчиков и датчиков \*\*/

typedef counter32 {

type uint32;

description

"Тип counter32 представляет неотрицательные целые числа, которые монотонно растут до максимального значения  $2^{32}-1$  (десятичное число 4294967295), затем счет повторяется с нуля.

Для счетчиков не определены 'начальные' значения, поэтому одиночное значение счетчика (обычно) не имеет смысла. Разрывы в монотонном росте значений счетчиков обычно связаны с реинициализацией системы управления или другими событиями, указанными в описании узла схемы, применяющего этот тип. Если не связанные с инициализацией события возможны, при создании счетчиков типа counter32 не в процессе инициализации, следует определить соответствующий узел схемы подходящего типа для указания последнего разрыва счетчика.

Тип counter32 не следует применять для конфигурационных узлов схемы. НЕ СЛЕДУЕТ применять оператор default вместе с типом counter32.

Набор значений и семантика этого типа эквивалентны типу Counter32 в SMIV2.";

reference

"RFC 2578: Structure of Management Information Version 2 (SMIV2)";

}

typedef zero-based-counter32 {

type yang:counter32;

default "0";

description

"Тип zero-based-counter32 представляет counter32 с 'начальным' значением 0.

Узел схемы этого типа будет устанавливаться в 0 при создании и далее будет монотонно расти до максимального значения  $2^{32}-1$  (десятичное число 4294967295), затем обращаться в 0 с последующим монотонным ростом.

При условии, что приложение обнаруживает новый узел схемы этого типа в течение минимального времени от перехода через 0, 'начальное' значение может считаться приращением. Для станции управления важно знать это минимальное время и реальное время между опросами, чтобы отбрасывать данные при слишком большой задержке или неопределенном минимальном времени.

Набор значений и семантика этого типа эквивалентны текстовому соглашению ZeroBasedCounter32 в SMIV2.";

reference

"RFC 4502: Remote Network Monitoring Management Information Base Version 2";

}

typedef counter64 {

type uint64;

description

"Тип counter64 представляет неотрицательные целые числа, которые монотонно растут до максимального значения  $2^{64}-1$  (десятичное число 18446744073709551615), затем счет повторяется с нуля.

Для счетчиков не определены 'начальные' значения, поэтому одиночное значение счетчика (обычно) не имеет смысла. Разрывы в монотонном росте значений счетчиков обычно связаны с реинициализацией системы управления или другими событиями, указанными в описании узла схемы, применяющего этот тип. Если не связанные с инициализацией события возможны, при создании счетчиков типа counter64 не в процессе инициализации, следует определить соответствующий узел схемы подходящего типа для указания последнего разрыва счетчика.

Тип counter64 не следует применять для конфигурационных узлов схемы. НЕ СЛЕДУЕТ применять оператор default вместе с типом counter64.

Набор значений и семантика этого типа эквивалентны типу Counter64 в SMIV2.";

reference

"RFC 2578: Structure of Management Information Version 2 (SMIV2)";

}

typedef zero-based-counter64 {

type yang:counter64;

default "0";

description

"Тип zero-based-counter64 представляет counter64 с 'начальным' значением 0.

Узел схемы этого типа будет устанавливаться в 0 при создании и далее будет монотонно расти до максимального значения  $2^{64}-1$  (десятичное число 18446744073709551615), затем обращаться в 0 с последующим монотонным ростом.

При условии, что приложение обнаруживает новый узел схемы этого типа в течение минимального времени от перехода через 0, 'начальное' значение может считаться приращением. Для станции управления важно знать это минимальное время и реальное время между опросами, чтобы отбрасывать данные при слишком большой задержке или неопределенном минимальном времени.

Набор значений и семантика этого типа эквивалентны текстовому соглашению ZeroBasedCounter64 в SMIV2.";

reference

"RFC 2856: Textual Conventions for Additional High Capacity Data Types";

}

```

typedef gauge32 {
    type uint32;
    description
        "Тип gauge32 представляет неотрицательное целое число,
        которое может расти и уменьшаться, не переходя максимального
        и минимального значения. Максимальное значение не может быть
        больше 2^32-1 (десятичное число 4294967295), а минимальное -
        меньше 0. Значение gauge32 достигает максимума, когда
        моделируемые данные не меньше максимума, а минимума - когда
        моделируемые данные не больше минимального значения. Если
        моделируемые затем данные снижаются (растут) ниже максимума
        (выше минимума) значение gauge32 также снижается (растет).

        Набор значений и семантика этого типа эквивалентны типу
        Gauge32 в SMIV2.";
    reference
        "RFC 2578: Structure of Management Information Version 2
        (SMIV2)";
}

typedef gauge64 {
    type uint64;
    description
        "Тип gauge64 представляет неотрицательное целое число,
        которое может расти и уменьшаться, не переходя максимального
        и минимального значения. Максимальное значение не может быть
        больше 2^64-1 (десятичное число 18446744073709551615), а
        минимальное - меньше 0. Значение gauge32 достигает максимума,
        когда моделируемые данные не меньше максимума, а минимума -
        когда моделируемые данные не больше минимального значения. Если
        моделируемые затем данные снижаются (растут) ниже максимума
        (выше минимума) значение gauge64 также снижается (растет).

        Набор значений и семантика этого типа эквивалентны текстовому
        соглашению CounterBasedGauge64 SMIV2, определенному в RFC 2856";
    reference
        "RFC 2856: Textual Conventions for Additional High Capacity
        Data Types";
}

/** Набор связанных с идентификаторами типов */
typedef object-identifier {
    type string {
        pattern '((([0-1](\.[1-3]?[0-9]))|(2\.(0|([1-9]\d*))))'
            + '(\.(0|([1-9]\d*)))*)';
    }
    description
        "Тип object-identifier представляет административно
        назначаемые имена в дереве registration-hierarchical-name.

        Значения этого типа представляются последовательностью
        числовых неотрицательных значений субидентификаторов, каждое
        ДОЛЖНО быть не более 2^32-1 (десятичное число 4294967295).
        Субидентификаторы разделяются одним символом точки без
        промежуточных пробельных символов.

        Стандарт ASN.1 ограничивает пространство значений первого
        субидентификатора цифрами 0, 1 и 2. Кроме того, диапазон
        второго субидентификатора ограничен значениями от 0 до 39,
        если первый субидентификатор равен 0 или 1. В дополнение
        стандарт ASN.1 требует наличия в идентификаторе объекта не
        менее 2 субидентификаторов. Шаблон соответствует этим
        ограничениям.

        Хотя число субидентификаторов не ограничено, разработчикам
        модулей следует понимать, что могут быть реализации, которые
        придерживаются принятого в SMIV2 ограничения в 128 значений.

        Этот тип является надмножеством типа SMIV2 OBJECT IDENTIFIER,
        поскольку не ограничен пределом в 128 субидентификаторов.
        Поэтому данный тип НЕ СЛЕДУЕТ применять для представления
        SMIV2 OBJECT IDENTIFIER, взамен СЛЕДУЕТ использовать тип
        object-identifier-128.";
    reference
        "ISO9834-1: Information technology -- Open Systems
        Interconnection -- Procedures for the operation of OSI

```

```
Registration Authorities: General procedures and top
arcs of the ASN.1 Object Identifier tree";
```

```
}

typedef object-identifier-128 {
  type object-identifier {
    pattern '\d*(\.\d*){1,127}';
  }
  description
    "Этот тип представляет идентификаторы объектов, содержащие
    не более 128 субидентификаторов.

    Набор значений и семантика этого типа эквивалентны типу
    OBJECT IDENTIFIER в SMIV2.";
  reference
    "RFC 2578: Structure of Management Information Version 2
    (SMIV2)";
}
```

```
typedef yang-identifier {
  type string {
    length "1..max";
    pattern '[a-zA-Z_][a-zA-Z0-9\-\_\.]*';
    pattern '\.\.\.[^xX].*\.[^mM].*\.\.\.[^lL].*';
  }
  description
    "Строка идентификатора YANG, определенная правилом
    'identifier' в разделе 12 RFC 6020. Идентификатор должен
    начинаться с буквы или символа подчеркивания, за которыми
    может следовать произвольное число букв, цифр, символов
    подчеркивания, дефисов и точек.

    Идентификаторам YANG НЕДОПУСТИМО начинаться с символов
    'xml' в любой комбинации строчных и прописных букв.";
  reference
    "RFC 6020: YANG - A Data Modeling Language for the Network
    Configuration Protocol (NETCONF)";
}
```

/\*\* Набор типов, связанных с датами и временем \*/

```
typedef date-and-time {
  type string {
    pattern '\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}(\.\d+)?'
      + '(Z|[\+-]\d{2}:\d{2})';
  }
  description
    "Тип date-and-time type является профилем стандарта ISO 8601
    для представления дат и времени с использованием григорианского
    календаря. Профиль определен date-time в параграфе 5.6 RFC 3339.

    Тип date-and-time совместим с типом dateTime схемы XML с учетом
    приведенных ниже исключений.

    (a) date-and-time не разрешает отрицательные значения лет.

    (b) В типа date-and-time time-offset -00:00 указывает
    неизвестный часовой пояс (RFC 3339), тогда как -00:00,
    +00:00 и Z представляют такой же часовой пояс в dateTime.

    (c) Канонический формат (см. ниже) значений data-and-time
    отличается от канонического формата типа dateTime XML,
    которые требует указания времени в UTC с использованием
    смещения 'Z'.


Этот тип не эквивалентен текстовому соглашению DateAndTime
в SMIV2, поскольку RFC 3339 использует другой разделитель
между full-date и full-time, а также обеспечивает большую
точность time-secfrac.

Канонический формат для значений date-and-time с известным
часовым поясом использует сдвиг часового пояса, который
рассчитывается с использованием настроенного в устройстве
смещения от UTC. Смена смещения в устройстве меняет значение
date-and-time должным образом. Такие изменения могут быть
периодическими в результате перехода на летнее время (DST1).
Канонический формат для значений date-and-time с


```

<sup>1</sup>Daylight saving time.

```

        неизвестным часовым поясом (обычно это называют местным
        временем) использует смещение -00:00.";
reference
    "RFC 3339: Date and Time on the Internet: Timestamps
    RFC 2579: Textual Conventions for SMIV2
    XSD-TYPES: XML Schema Part 2: Datatypes Second Edition";
}

typedef timeticks {
    type uint32;
    description
        "Тип timeticks представляет неотрицательные целые числа,
        которые указывают по модулю 2^32 (десятичное число
        4294967296) время в сотых долях секунды между двумя эпохами.
        При определении узла схемы, использующего этот тип, описание
        узла указывает обе опорных эпохи.

        Набор значений и семантика этого типа эквивалентны типу
        TimeTicks в SMIV2.";
reference
    "RFC 2578: Structure of Management Information Version 2
    (SMIV2)";
}

typedef timestamp {
    type yang:timeticks;
    description
        "Тип timestamp представляет значение связанного с ним узла
        схемы timeticks, где происходит конкретное вхождение,
        которое должно быть определено в описании каждого узла схемы,
        определенного с использованием этого типа. Когда конкретное
        вхождение происходит до того, как связанный атрибут timeticks
        в последний раз был равен нулю, значение timestamp = 0.
        Отметим, что это требует сброса в 0 всех значений timestamp,
        когда значение связанного атрибута timeticks превышает 497
        дней и переходит через 0.

        Связанный узел схемы timeticks должен быть задан в описании
        любого узла схемы, использующего этот тип.

        Набор значений и семантика этого типа эквивалентны
        текстовому соглашению TimeStamp в SMIV2.";
reference
    "RFC 2579: Textual Conventions for SMIV2";
}

/** Набор базовых типов для адресов */
typedef phys-address {
    type string {
        pattern '([0-9a-fA-F]{2}(:[0-9a-fA-F]{2})*)?';
    }
    description
        "Представляет адрес среды или физического уровня в форме
        последовательности октетов, каждый из которых указывается
        двумя шестнадцатеричными цифрами. В каноническом
        представлении используются строчные буквы (нижний регистр).

        Набор значений и семантика этого типа эквивалентны
        текстовому соглашению PhysAddress в SMIV2.";
reference
    "RFC 2579: Textual Conventions for SMIV2";
}

typedef mac-address {
    type string {
        pattern '[0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){5}';
    }
    description
        "Тип mac-address представляет адрес IEEE 802 MAC.
        В каноническом представлении используются строчные буквы.

        Набор значений и семантика этого типа эквивалентны
        текстовому соглашению MacAddress в SMIV2.";
reference
    "IEEE 802: IEEE Standard for Local and Metropolitan Area
    Networks: Overview and Architecture

```

```

RFC 2579: Textual Conventions for SMIv2";
}

/** Набор относящихся к XML типов */
typedef xpath1.0 {
  type string;
  description
    "Этот тип представляет выражение XPATH 1.0.

    Когда определяется узел схемы, использующий этот тип,
    описание узла ДОЛЖНО указывать контекст XPath, в котором
    вычисляется выражение XPath.";
  reference
    "XPATH: XML Path Language (XPath) Version 1.0";
}

/** Набор строковых типов */
typedef hex-string {
  type string {
    pattern '([0-9a-fA-F]{2}(:[0-9a-fA-F]{2})*)?';
  }
  description
    "Шестнадцатеричная строка с октетами, представленными двумя
    шестнадцатеричными цифрами, разделенными двоеточием. В
    каноническом варианте применяются строчные буквы.";
}

typedef uuid {
  type string {
    pattern '[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-'
      + '[0-9a-fA-F]{4}-[0-9a-fA-F]{12}';
  }
  description
    "Глобально уникальный идентификатор (UUID) в строковом
    представлении, определенном в RFC 4122. В каноническом вариант
    применяются строчные буквы.

    Ниже приведен пример строкового представления UUID
    f81d4fae-7dec-11d0-a765-00a0c91e6bf6";
  reference
    "RFC 4122: A Universally Unique Identifier (UUID) URN
    Namespace";
}

typedef dotted-quad {
  type string {
    pattern
      '((([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.){3}'
      + '([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])');
  }
  description
    "32-битовое целое число без знака, представленное 4 десятичными
    числами, разделенными точками (full stop).";
}
}

<CODE ENDS>

```

#### 4. Связанные с Internet производные типы

Модуль YANG `ietf-inet-types` ссылается на [RFC0768], [RFC0791], [RFC0793], [RFC0952], [RFC1034], [RFC1123], [RFC1930], [RFC2460], [RFC2474], [RFC2780], [RFC2782], [RFC3289], [RFC3305], [RFC3595], [RFC3986], [RFC4001], [RFC4007], [RFC4271], [RFC4291], [RFC4340], [RFC4960], [RFC5017], [RFC5890], [RFC5952], [RFC6793].

```

<CODE BEGINS> file "ietf-inet-types@2013-07-15.yang"

module ietf-inet-types {

  namespace "urn:ietf:params:xml:ns:yang:ietf-inet-types";
  prefix "inet";

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

  contact
    "WG Web: <http://tools.ietf.org/wg/netmod/>
    WG List: <mailto:netmod@ietf.org>

```



WG Chair: David Kessens <<mailto:david.kessens@nsn.com>>

WG Chair: Juergen Schoenwaelder  
<<mailto:j.schoenwaelder@jacobs-university.de>>

Editor: Juergen Schoenwaelder  
<<mailto:j.schoenwaelder@jacobs-university.de>>;

#### description

"Этот модуль содержит набор производных типов данных YANG для адресов Internet и связанных элементов.

Авторские права (Copyright (c) 2013) принадлежат IETF Trust и лицам, указанным как авторы кода. Все права защищены.

Распространение и использование в исходной и двоичной форме с изменениями или без них разрешается в соответствии с условиями, указанными в упрощенной лицензии BSD, изложенной в разделе 4.с Правового положения IETF Trust применительно к документам IETF (<http://trustee.ietf.org/license-info>).

Эта версия модуля YANG является частью RFC 6991, где правовые аспекты выражены более полно.";

revision 2013-07-15 {

#### description

"В этом выпуске добавлены новые типы данных:

- ip-address-no-zone
- ipv4-address-no-zone
- ipv6-address-no-zone";

#### reference

"RFC 6991: Common YANG Data Types";

}

revision 2010-09-24 {

#### description

"Первый выпуск.";

#### reference

"RFC 6021: Common YANG Data Types";

}

/\*\* Набор типов, связанных с протокольными полями \*/

typedef ip-version {

type enumeration {

enum unknown {

value "0";

description

"Неизвестная или не указанная версия протокола IP.";

}

enum ipv4 {

value "1";

description

"Протокол IPv4 в соответствии с RFC 791.";

}

enum ipv6 {

value "2";

description

"Протокол IPv6 в соответствии с RFC 2460.";

}

}

#### description

"Это значение представляет версию протокола IP.

По набору значений и семантике этот тип эквивалентен текстовому соглашению InetVersion в SMIPv2.";

#### reference

"RFC 791: Internet Protocol

RFC 2460: Internet Protocol, Version 6 (IPv6) Specification

RFC 4001: Textual Conventions for Internet Network Addresses";

}

typedef dscp {

type uint8 {

range "0..63";

}

#### description

"Тип dscp представляет коды дифференцированного обслуживания, которые могут применяться для маркировки пакетов в потоке.

Набор значений и семантика этого типа эквивалентны текстовому соглашению Dscp в SMiv2.";

reference

"RFC 3289: Management Information Base for the Differentiated Services Architecture  
RFC 2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers  
RFC 2780: IANA Allocation Guidelines For Values In the Internet Protocol and Related Headers";

}

typedef ipv6-flow-label {

type uint32 {  
range "0..1048575";

}

description

"Тип ipv6-flow-label предоставляет идентификатор или метку потока в заголовке пакета IPv6, которая может служить для различения потоков трафика.

Набор значений и семантика этого типа эквивалентны текстовому соглашению IPv6FlowLabel в SMiv2.";

reference

"RFC 3595: Textual Conventions for IPv6 Flow Label  
RFC 2460: Internet Protocol, Version 6 (IPv6) Specification";

}

typedef port-number {

type uint16 {  
range "0..65535";

}

description

"Тип port-number представляет 16-битовый номер порт протоколов транспортного уровня Internet, таких как UDP, TCP, DCCP, SCTP. Номера портов распределяются IANA. Текущий список выделенных портов доступен на сайте <<http://www.iana.org/>>.

Отметим, что номер 0 зарезервирован IANA. В ситуациях, где нулевое значение не имеет смысла, оно может быть исключено путем создания субтипа для port-number без 0.

Набор значений и семантика для этого типа эквивалентны текстовому соглашению InetPortNumber в SMiv2.";

reference

"RFC 768: User Datagram Protocol  
RFC 793: Transmission Control Protocol  
RFC 4960: Stream Control Transmission Protocol  
RFC 4340: Datagram Congestion Control Protocol (DCCP)  
RFC 4001: Textual Conventions for Internet Network Addresses";

}

/\*\* Набор типов, связанных с автономными системами \*/

typedef as-number {

type uint32;

description

"Тип as-number представляет номера автономных систем (AS<sup>1</sup>).

AS образует набор маршрутизаторов с общим техническим администрированием, использующих протокол внутренней маршрутизации и общую метрику внутри AS, а также протокол внешней маршрутизации для пересылки пакетов в другие AS. IANA поддерживает пространство номеров AS, передав большую часть блоков AS для распределения региональным регистраторам.

Номера AS исходно были 16-битовыми, но расширения BGP увеличили размер пространства номеров AS до 32 битов. Поэтому данный тип использует базовый тип uint32 без ограничения диапазона для поддержки полного пространства.

Набор значений и семантика этого типа эквивалентны текстовому соглашению InetAutonomousSystemNumber в SMiv2.";

reference

"RFC 1930: Guidelines for creation, selection, and registration of an Autonomous System (AS)

<sup>1</sup>Autonomous System.

```

RFC 4271: A Border Gateway Protocol 4 (BGP-4)
RFC 4001: Textual Conventions for Internet Network Addresses
RFC 6793: BGP Support for Four-Octet Autonomous System (AS)
          Number Space";
}

/**/ Набор типов, связанных с адресами IP и именами хостов ***/
typedef ip-address {
    type union {
        type inet:ipv4-address;
        type inet:ipv6-address;
    }
    description
    "Тип ip-address представляет адрес IP без учета версии протокола.
    Формат текстового представления предполагает версию IP.
    Этот тип поддерживает ограниченные (scoped) адреса, разрешая
    указывать идентификаторы зон в формате адресов.";
    reference
    "RFC 4007: IPv6 Scoped Address Architecture";
}

typedef ipv4-address {
    type string {
        pattern
        '(([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.){3}'
        + '([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])'
        + '(%[\p{N}\p{L}]+)?';
    }
    description
    "Тип ipv4-address представляет адрес IPv4 в десятичной
    нотации с разделением тосками. Адрес IPv4 может включать
    индекс зоны, отделенный символом %."

    "Индекс зоны служит для того, чтобы различать идентичные
    значения адресов. Для адресов link-local индексом зоны
    обычно служит индекс или имя интерфейса. Если индекса зоны
    нет, будет использоваться принятая по умолчанию зона
    устройства."

    "Каноническим форматом для индекса зоны является числовой
    формат.";
}

typedef ipv6-address {
    type string {
        pattern '((:|[0-9a-fA-F]{0,4}):)([0-9a-fA-F]{0,4}:){0,5}'
        + '(((0-9a-fA-F){0,4}:)?(:|[0-9a-fA-F]{0,4}))|'
        + '(((25[0-5]|2[0-4][0-9]|01?[0-9]?[0-9])\.){3}'
        + '(25[0-5]|2[0-4][0-9]|01?[0-9]?[0-9]))|'
        + '(%[\p{N}\p{L}]+)?';
        pattern '([[:]:+]{6}([[:]:+]|\.|\.|\.))|'
        + '([[:]:+]*[[:]:+]?::([[:]:+]*[[:]:+]?)'
        + '(%.+)?';
    }
    description
    "Тип ipv6-address представляет адрес IPv6 в полной, смешанной
    сокращенной или сокращенной смешанной нотации. Адрес IPv6
    может включать индекс зоны, отделенный символом %."
    "Индекс зоны служит для того, чтобы различать идентичные
    значения адресов. Для адресов link-local индексом зоны
    обычно служит индекс или имя интерфейса. Если индекса зоны
    нет, будет применяться принятая по умолчанию зона устройства."

    "Канонический формат адреса IPv6 использует текстовое
    представление, определенное в разделе 4 RFC 5952.
    Для индекса зоны каноническим является числовой
    формат, описанный в параграфе 11.2 RFC 4007.";
    reference
    "RFC 4291: IP Version 6 Addressing Architecture
    RFC 4007: IPv6 Scoped Address Architecture
    RFC 5952: A Recommendation for IPv6 Address Text
    Representation";
}

typedef ip-address-no-zone {
    type union {

```

```

type inet:ipv4-address-no-zone;
type inet:ipv6-address-no-zone;
}
description
"Тип ip-address представляет адрес IP без учета версии протокола.
Формат текстового представления предполагает версию IP.
Этот тип не поддерживает ограниченные (scoped) адреса, поскольку
не разрешает указывать идентификаторы зон в формате адресов.";
reference
"RFC 4007: IPv6 Scoped Address Architecture";
}

typedef ipv4-address-no-zone {
type inet:ipv4-address {
pattern '[0-9\.]*';
}
description
"Адрес IPv4 без индекса зоны. Этот тип, производный от
ipv4-address, может применяться в ситуациях, где индекс зоны
известен из контекста и поэтому не требуется.";
}

typedef ipv6-address-no-zone {
type inet:ipv6-address {
pattern '[0-9a-fA-F\.]*';
}
description
"Адрес IPv6 без индекса зоны. Этот тип, производный от
ipv6-address, может применяться в ситуациях, где индекс зоны
известен из контекста и поэтому не требуется.";
reference
"RFC 4291: IP Version 6 Addressing Architecture
RFC 4007: IPv6 Scoped Address Architecture
RFC 5952: A Recommendation for IPv6 Address Text
Representation";
}

typedef ip-prefix {
type union {
type inet:ipv4-prefix;
type inet:ipv6-prefix;
}
description
"Тип ip-prefix представляет префикс IP без учета версии IP.
Формат текстового представления предполагает версию IP.";
}

typedef ipv4-prefix {
type string {
pattern
'((([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.){3}'
+ '([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])'
+ '/((([0-9])|([1-2][0-9])|(3[0-2])))';
}
description
"Тип ipv4-prefix представляет адресный префикс IPv4. Размер
префикса указывается числом после знака / и должен
быть не больше 32.
Размер префикса n соответствует маске адреса IP, где
n старших битов подряд имеют значение 1, а остальные 0.

Канонический формат префикса IPv4 имеет значение 0 во всех
битах адреса IPv4, не являющихся частью префикса IPv4.";
}

typedef ipv6-prefix {
type string {
pattern '((:[0-9a-fA-F]{0,4})|(:[0-9a-fA-F]{0,4}){0,5})'
+ '(((:[0-9a-fA-F]{0,4})|(:[0-9a-fA-F]{0,4}))|)'
+ '(((25[0-5]|2[0-4][0-9]|01?[0-9]?[0-9])\.){3}'
+ '(25[0-5]|2[0-4][0-9]|01?[0-9]?[0-9]))'
+ '/((([0-9])|([0-9]{2})|(1[0-1][0-9])|(12[0-8])))';
pattern '(([:+]{6}([[:+]:+[:+])|(.*\..*))|)'
+ '((([:+]:+)*[:+]?::(([:+]:+)*[:+])?)'
+ '(/.+)'
}
}

```

```

description
"Тип ipv6-prefix представляет адресный префикс IPv6. Размер
префикса указывается числом после знака / и должен
быть не больше 128.

Размер префикса n соответствует маске адреса IP, где
n старших битов подряд имеют значение 1, а остальные 0.

В адресе IPv6 все биты, не относящиеся к префиксу, следует
устанавливать в 0.

Канонический формат префикса IPv6 имеет значение 0 во всех
битах адреса IPv6, не являющихся частью префикса IPv6.
Кроме того, адреса IPv6 представляется в соответствии
с разделом 4 RFC 5952.";
reference
"RFC 5952: A Recommendation for IPv6 Address Text
Representation";
}

/** Набор типов для доменных имен и URI */
typedef domain-name {
type string {
pattern
'((([a-zA-Z0-9_]([a-zA-Z0-9\_-]){0,61})?[a-zA-Z0-9]\.)*'
+ '([a-zA-Z0-9_]([a-zA-Z0-9\_-]){0,61})?[a-zA-Z0-9]\.?)'
+ '\\\.';
length "1..253";
}
description
"Тип domain-name представляет доменные имена DNS.
По-возможности СЛЕДУЕТ применять полные имена (FQDN1).

Доменные имена Internet не заданы жестко. Параграф 3.5 в
RFC 1034 задает рекомендуемый синтаксис (изменен в
параграфе 2.1 RFC 1123). Показанный выше шаблон рассчитан
на современную практику применения доменных имен и
возможные расширения. Он предназначен для записи разных
типов доменных имен, включая имена в записях A или AAAA
(имена хостов) и других записях, атких как SRV. Отметим,
что имена хостов Internet имеют более строгий синтаксис
(см. RFC 952), чем рекомендации DNS в RFC 1034 и 1123, и
системам, которые хотят хранить имена хостов в узлах схем
с применением типа domain-name, рекомендуется следовать
более строгому синтаксису для обеспечения совместимости.

Размер имен DNS в протоколе DNS ограничен 255 символами.
Поскольку в кодировании применяются метки с префиксом,
задающим размер в байтах, и завершающим NULL-байтом,
максимальный размер текстовой части метки составляет 253.

Раздел описания узлов схемы, использующих тип domain-name,
ДОЛЖЕН указывать, когда и как эти имена преобразуются в адреса
IP. Отметим, что для преобразования значений domain-name может
потребоваться запрос множества записей DNS (например, A для IPv4
и AAAA для IPv6). Порядок преобразования и приоритет записей DNS
могут указываться явно или определяться конфигурацией
распознавателя (resolver).

Значения доменных имен используют кодировку US-ASCII со строчными
буквами US-ASCII для канонического формата. Имена на других языках
ДОЛЖНЫ быть A-метками в соответствии с RFC 5890.";
reference
"RFC 952: DoD Internet Host Table Specification
RFC 1034: Domain Names - Concepts and Facilities
RFC 1123: Requirements for Internet Hosts -- Application
and Support
RFC 2782: A DNS RR for specifying the location of services
(DNS SRV)
RFC 5890: Internationalized Domain Names in Applications
(IDNA): Definitions and Document Framework";
}

typedef host {
type union {

```

<sup>1</sup>Fully qualified domain name.

```

    type inet:ip-address;
    type inet:domain-name;
}
description
  "Тип host представляет адрес IP или доменное имя DNS.";
}

typedef uri {
  type string;
  description
    "Тип uri представляет идентификаторы URI1, определенные в STD 66.

    Объекты, использующие тип uri, ДОЛЖНЫ указываться в кодировке
    US-ASCII и ДОЛЖНЫ нормализоваться в соответствии с параграфами
    6.2.1, 6.2.2.1 и 6.2.2.2 в RFC 3986. Все необязательные
    %-представления удаляются, для независимых от регистра символов
    устанавливается нижний регистр, за исключением шестнадцатеричных
    цифр, которые нормализуются в верхний регистр как указано в
    параграфе 6.2.2.1.

    Целью этой нормализации является помощь в создании уникальных
    URI. Отметим, что нормализации не достаточно для обеспечения
    уникальности. Два URI, которые после нормализации различаются
    текстом, могут оставаться эквивалентными.

    Объекты, использующие тип uri, могут ограничивать разрешенные
    схемы. Например, схемы 'data:' и 'urn:' могут не подойти.

    URI нулевого размера не являются пригодными. Они могут служить
    для указания 'отсутствия URI' при необходимости.

    Набор значений и семантика этого типа эквивалентны
    текстовому соглашению Uri SMIV2, определенному в RFC 5017.";
  reference
    "RFC 3986: Uniform Resource Identifier (URI): Generic Syntax
    RFC 3305: Report from the Joint W3C/IETF URI Planning Interest
    Group: Uniform Resource Identifiers (URIs), URLs,
    and Uniform Resource Names (URNs): Clarifications
    and Recommendations
    RFC 5017: MIB Textual Conventions for Uniform Resource
    Identifiers (URIs)";
}
}

<CODE ENDS>

```

## 5. Взаимодействие с IANA

Этот документ регистрирует два URI в реестре IETF XML [RFC3688]. В соответствии с форматом RFC 3688 были сделаны приведенные ниже регистрации.

```

URI: urn:ietf:params:xml:ns:yang:ietf-yang-types
Registrant Contact: The NETMOD WG of the IETF.
XML: N/A, the requested URI is an XML namespace.

```

```

URI: urn:ietf:params:xml:ns:yang:ietf-inet-types
Registrant Contact: The NETMOD WG of the IETF.
XML: N/A, the requested URI is an XML namespace.

```

Документ регистрирует два модуля YANG в реестре YANG Module Names [RFC6020].

```

name:          ietf-yang-types
namespace:     urn:ietf:params:xml:ns:yang:ietf-yang-types
prefix:        yang
reference:     RFC 6991

```

```

name:          ietf-inet-types
namespace:     urn:ietf:params:xml:ns:yang:ietf-inet-types
prefix:        inet
reference:     RFC 6991

```

## 6. Вопросы безопасности

Этот документ определяет типы данных общего назначения для языка моделирования YANG. Определения сами по себе не влияют на безопасность, но использование этих определений в конкретных модулях YANG может иметь такое влияние. Вопросы безопасности, рассмотренные в спецификации YANG [RFC6020], применимы и к этому документу.

<sup>1</sup>Uniform Resource Identifier — унифицированный идентификатор ресурса.

## 7. Участники работы

Ниже перечислены внесшие существенный вклад участники разработки начальной версии этого документа.

- Andy Bierman (Brocade)
- Martin Bjorklund (Tail-f Systems)
- Balazs Lengyel (Ericsson)
- David Partain (Ericsson)
- Phil Shafer (Juniper Networks)

## 8. Благодарности

Редактор благодарит за полезные замечания к разным версиям этого документа Andy Bierman, Martin Bjorklund, Benoit Claise, Joel M. Halpern, Ladislav Lhotka, Lars-Johan Liman и Dan Romascanu.

Работу Juergen Schoenwaelder частично финансировал проект Flamingo в рамках Network of Excellence (ICT-318488), поддерживаемый Европейской комиссией по программе Seventh Framework.

## 9. Литература

### 9.1. Нормативные документы

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, [RFC 2119](#), March 1997.
- [RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, [RFC 3688](#), January 2004.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, [RFC 3986](#), January 2005.
- [RFC4007] Deering, S., Haberman, B., Jinmei, T., Nordmark, E., and B. Zill, "IPv6 Scoped Address Architecture", RFC 4007, March 2005.
- [RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, July 2005.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", [RFC 4291](#), February 2006.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](#), October 2010.
- [XPATH] Clark, J. and S. DeRose, "XML Path Language (XPath) Version 1.0", World Wide Web Consortium Recommendation REC-xpath-19991116, November 1999, <<http://www.w3.org/TR/1999/REC-xpath-19991116>>.

### 9.2. Дополнительная литература

- [IEEE802] IEEE, "IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture", IEEE Std. 802-2001, 2001.
- [ISO9834-1] ISO/IEC, "Information technology -- Open Systems Interconnection -- Procedures for the operation of OSI Registration Authorities: General procedures and top arcs of the ASN.1 Object Identifier tree", ISO/IEC 9834-1:2008, 2008.
- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, [RFC 768](#), August 1980.
- [RFC0791] Postel, J., "Internet Protocol", STD 5, [RFC 791](#), September 1981.
- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, [RFC 793](#), September 1981.
- [RFC0952] Harrenstien, K., Stahl, M., and E. Feinler, "DoD Internet host table specification", [RFC 952](#), October 1985.
- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, [RFC 1034](#), November 1987.
- [RFC1123] Braden, R., "Requirements for Internet Hosts - Application and Support", STD 3, [RFC 1123](#), October 1989.
- [RFC1930] Hawkinson, J. and T. Bates, "Guidelines for creation, selection, and registration of an Autonomous System (AS)", BCP 6, [RFC 1930](#), March 1996.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", [RFC 2460](#), December 1998.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", [RFC 2474](#), December 1998.
- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIPv2)", STD 58, RFC 2578, April 1999.
- [RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIPv2", STD 58, RFC 2579, April 1999.
- [RFC2780] Bradner, S. and V. Paxson, "IANA Allocation Guidelines For Values In the Internet Protocol and Related Headers", BCP 37, RFC 2780, March 2000.
- [RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

- [RFC2856] Bierman, A., McCloghrie, K., and R. Presuhn, "Textual Conventions for Additional High Capacity Data Types", RFC 2856, June 2000.
- [RFC3289] Baker, F., Chan, K., and A. Smith, "Management Information Base for the Differentiated Services Architecture", RFC 3289, May 2002.
- [RFC3305] Mealling, M. and R. Denenberg, "Report from the Joint W3C/IETF URI Planning Interest Group: Uniform Resource Identifiers (URIs), URLs, and Uniform Resource Names (URNs): Clarifications and Recommendations", RFC 3305, August 2002.
- [RFC3595] Wijnen, B., "Textual Conventions for IPv6 Flow Label", RFC 3595, September 2003.
- [RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, February 2005.
- [RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", [RFC 4271](#), January 2006.
- [RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", [RFC 4340](#), March 2006.
- [RFC4502] Waldbusser, S., "Remote Network Monitoring Management Information Base Version 2", RFC 4502, May 2006.
- [RFC4960] Stewart, R., "Stream Control Transmission Protocol", [RFC 4960](#), September 2007.
- [RFC5017] McWalter, D., "MIB Textual Conventions for Uniform Resource Identifiers (URIs)", RFC 5017, September 2007.
- [RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.
- [RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, August 2010.
- [RFC6021] Schoenwaelder, J., "Common YANG Data Types", RFC 6021, October 2010.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.
- [RFC6793] Vohra, Q. and E. Chen, "BGP Support for Four-Octet Autonomous System (AS) Number Space", [RFC 6793](#), December 2012.
- [XSD-TYPES] Biron, P. and A. Malhotra, "XML Schema Part 2: Datatypes Second Edition", World Wide Web Consortium Recommendation REC-xmlschema-2-20041028, October 2004, <<http://www.w3.org/TR/2004/REC-xmlschema-2-20041028>>.

## Приложение А. Отличия от RFC 6021

В этой версии добавлены новые определения типов для модулей YANG. Ниже перечислены типы, добавленные в модуль ietf-yang-types.

- yang-identifier
- hex-string
- uuid
- dotted-quad

Ниже перечислены типы, добавленные в модуль ietf-inet-types.

- ip-address-no-zone
- ipv4-address-no-zone
- ipv6-address-no-zone

### Адрес автора

**Juergen Schoenwaelder (редактор)**

Jacobs University

E-Mail: [j.schoenwaelder@jacobs-university.de](mailto:j.schoenwaelder@jacobs-university.de)

### Перевод на русский язык

**Николай Малых**

[nmalykh@protocols.ru](mailto:nmalykh@protocols.ru)