

Common YANG Data Types

Аннотация

Этот документ вводит набор типов данных общего назначения для использования в языке моделирования данных YANG. Он включает несколько определений новых типов и отменяет RFC 6991.

Статус документа

Документ относится к категории Internet Standards Track.

Документ является результатом работы IETF¹ и представляет согласованный взгляд сообщества IETF. Документ прошёл открытое обсуждение и был одобрен для публикации IESG². Дополнительные сведения о документах Internet Standard приведены в разделе 2 RFC 7841.

Информацию о текущем статусе документа, ошибках и способах обратной связи можно найти по ссылке <https://www.rfc-editor.org/info/rfc9911>.

Авторские права

Авторские права (Copyright (c) 2025) принадлежат IETF Trust и лицам, указанным в качестве авторов документа. Все права защищены.

К документу применимы права и ограничения, указанные в BCP 78 и IETF Trust Legal Provisions и относящиеся к документам IETF (<http://trustee.ietf.org/license-info>), на момент публикации данного документа. Прочтите упомянутые документы внимательно. Фрагменты программного кода, включённые в этот документ, распространяются в соответствии с пересмотренной лицензией BSD (Revised BSD License), как указано в параграфе 4.e документа IETF Trust Legal Provisions, без каких-либо гарантий (как указано в Revised 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 изменён 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.....	3
4. Типы для стека протоколов Internet.....	11
5. Взаимодействие с IANA.....	19
6. Вопросы безопасности.....	19
7. Литература.....	19
7.1. Нормативные документы.....	19
7.2. Дополнительная литература.....	20
Благодарности.....	21
Адрес автора.....	21

1. Введение

YANG [RFC7950] является языком моделирования данных, служащим для создания моделей конфигурации и состояния, которыми манипулирует протокол NETCONF³ [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 определяет набор типов данных общего назначения. Определения представлены в форме двух модулей YANG:

- модуль `ietf-yang-types` определяет типы данных общего назначения, такие как счётчики и измерители (датчики), типы, связанные с датами и временем, а также типы для строковых значений (например, UUID, числа, разделённые точками, таги языка);

¹Internet Engineering Task Force - комиссия по решению инженерных задач Internet.

²Internet Engineering Steering Group - комиссия по инженерным разработкам Internet.

³Network Configuration Protocol - протокол настройки конфигурации сети.

- модуль `ietf-inet-types` определяет типы данных, связанные со стеком протоколов Internet, такие как адреса IP, доменные имена, имена хостов, URI, адреса электронной почты и типы для базовых полей протоколов (например, номера портов).

Первая версия этих модулей YANG была представлена в [RFC6021]. При пересмотре [RFC6021] в [RFC6991] были добавлены несколько определений типов. Данный пересмотр добавляет определения новых типов и решает проблемы, отмеченные в Erratum ID 4076 [Err4076] и 5105 [Err5105]. Кроме того, определение `yang-identifier` приведено в соответствии с YANG 1.1 [RFC7950], а также улучшены некоторые операторы `pattern`. Дополнительные сведения приведены в операторах `revision` модулей YANG разделов 3 и 4. Краткий обзор всех представленных в этом документе типов дан в разделе 2. В будущем могут быть добавлены новые определения типов путём представления предложений в рабочую группу NETMOD.

В документе используется терминология YANG, определённая в разделе 3 [RFC7950].

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не нужно** (SHALL NOT), **следует** (SHOULD), **не следует** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **не рекомендуется** (NOT RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с BCP 14 [RFC2119] [RFC8174] тогда и только тогда, когда они выделены шрифтом, как показано здесь.

2. Обзор

В таблицах 1 и 2 перечислены типы, определённые в модулях YANG `ietf-yang-types` и `ietf-inet-types`. Для каждого типа указано имя, базовый тип и RFC, где данный тип был определён.

Таблица 1. Типы, определённые в модуле `ietf-yang-types`.

Тип	Базовый тип	Документ
<code>counter32</code>	<code>uint32</code>	RFC 6021
<code>zero-based-counter32</code>	<code>uint32</code>	RFC 6021
<code>counter64</code>	<code>uint64</code>	RFC 6021
<code>zero-based-counter64</code>	<code>uint64</code>	RFC 6021
<code>gauge32</code>	<code>uint32</code>	RFC 6021
<code>gauge64</code>	<code>uint64</code>	RFC 6021
<code>object-identifier</code>	<code>string</code>	RFC 6021
<code>object-identifier-128</code>	<code>object-identifier</code>	RFC 6021
<code>date-and-time</code>	<code>string</code>	RFC 6021
<code>date</code>	<code>string</code>	RFC 9911
<code>date-no-zone</code>	<code>string</code>	RFC 9911
<code>time</code>	<code>string</code>	RFC 9911
<code>time-no-zone</code>	<code>string</code>	RFC 9911
<code>hours32</code>	<code>int32</code>	RFC 9911
<code>minutes32</code>	<code>int32</code>	RFC 9911
<code>seconds32</code>	<code>int32</code>	RFC 9911
<code>centiseconds32</code>	<code>int32</code>	RFC 9911
<code>milliseconds32</code>	<code>int32</code>	RFC 9911
<code>microseconds32</code>	<code>int32</code>	RFC 9911
<code>microseconds64</code>	<code>int64</code>	RFC 9911
<code>nanoseconds32</code>	<code>int32</code>	RFC 9911
<code>nanoseconds64</code>	<code>int64</code>	RFC 9911
<code>timeticks</code>	<code>int32</code>	RFC 6021
<code>timestamp</code>	<code>timeticks</code>	RFC 6021
<code>phys-address</code>	<code>string</code>	RFC 6021
<code>mac-address</code>	<code>string</code>	RFC 6021
<code>xpath1.0</code>	<code>string</code>	RFC 6021
<code>hex-string</code>	<code>string</code>	RFC 6991
<code>uuid</code>	<code>string</code>	RFC 6991
<code>dotted-quad</code>	<code>string</code>	RFC 6991
<code>language-tag</code>	<code>string</code>	RFC 9911
<code>yang-identifier</code>	<code>string</code>	RFC 6991

Таблица 2. Типы, определённые в модуле `ietf-inet-types`.

Тип	Базовый тип	Документ
<code>ip-version</code>	<code>enum</code>	RFC 6021
<code>dscp</code>	<code>uint8</code>	RFC 6021
<code>ipv6-flow-label</code>	<code>uint32</code>	RFC 6021
<code>port-number</code>	<code>uint16</code>	RFC 6021
<code>protocol-number</code>	<code>uint8</code>	RFC 9911
<code>upper-layer-protocol-number</code>	<code>protocol-number</code>	RFC 9911
<code>as-number</code>	<code>uint32</code>	RFC 6021
<code>ip-address</code>	<code>union</code>	RFC 6021
<code>ipv4-address</code>	<code>string</code>	RFC 6021
<code>ipv6-address</code>	<code>string</code>	RFC 6021
<code>ip-address-no-zone</code>	<code>union</code>	RFC 6991
<code>ipv4-address-no-zone</code>	<code>ipv4-address</code>	RFC 6991
<code>ipv6-address-no-zone</code>	<code>ipv6-address</code>	RFC 6991
<code>ip-address-link-local</code>	<code>union</code>	RFC 9911
<code>ipv4-address-link-local</code>	<code>ipv4-address</code>	RFC 9911
<code>ipv6-address-link-local</code>	<code>ipv6-address</code>	RFC 9911
<code>ip-prefix</code>	<code>union</code>	RFC 6021

ipv4-prefix	string	RFC 6021
ipv6-prefix	string	RFC 6021
ip-address-and-prefix	union	RFC 9911
ipv4-address-and-prefix	string	RFC 9911
ipv6-address-and-prefix	string	RFC 9911
domain-name	string	RFC 6021
host-name	domain-name	RFC 9911
host	union	RFC 6021
uri	string	RFC 6021
email-address	string	RFC 9911

Некоторые типы имеют эквиваленты в Structure of Management Information Version 2 (SMIv2) [RFC2578] [RFC2579]. Тип данных YANG является эквивалентным типу SMIv2, если оба типа имеют эквивалентные значения и их семантику.

В таблице 3 указаны типы из модуля ietf-yang-types и соответствующие типы SMIv2, а в таблице 4 - типы из ietf-inet-types и соответствующие типы SMIv2.

Таблица 3. Типы ietf-yang-types и их эквиваленты в SMIv2.

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-128	OBJECT IDENTIFIER
centiseconds32	TimeInterval (SNMPv2-TC)
timeticks	TimeTicks (SNMPv2-SMI)
timestamp	TimeStamp (SNMPv2-TC)
phys-address	PhysAddress (SNMPv2-TC)
mac-address	MacAddress (SNMPv2-TC)
language-tag	LangTag (LANGTAG-TC-MIB)

Таблица 4. Типы ietf-inet-types и их эквиваленты в SMIv2.

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)
uri	Uri (URI-TC-MIB)

3. Базовые производные типы YANG

Модуль YANG ietf-yang-types ссылается на [IEEE-802-2024], [ISO-8601], [ISO-9834-1], [RFC2578], [RFC2579], [RFC2856], [RFC3339], [RFC4502], [RFC5131], [RFC5646], [RFC7950], [RFC9557], [RFC9562], [XPATH], [XSD-TYPES].

```
<CODE BEGINS> file "ietf-yang-types@2025-12-22.yang"
module ietf-yang-types {
  namespace "urn:ietf:params:xml:ns:yang:ietf-yang-types";
  prefix yang;

  organization
    "IETF Network Modeling (NETMOD) Working Group";
  contact
    "WG Web: <https://datatracker.ietf.org/wg/netmod/>
    WG List: <mailto:netmod@ietf.org>

    Editor: Jürgen Schönwälder
           <mailto:jschoenwaelder@constructor.university>";
  description
    "Этот модуль содержит набор производных типов данных YANG
    общего назначения.
```

Ключевые слова ДОЛЖНО, НЕДОПУСТИМО, ТРЕБУЕТСЯ, НУЖНО, НЕ НУЖНО, СЛЕДУЕТ, НЕ СЛЕДУЕТ, РЕКОМЕНДУЕТСЯ, НЕ РЕКОМЕНДУЕТСЯ, МОЖНО, НЕОБЯЗАТЕЛЬНО в этом документе трактуются в соответствии с ВСП 14 (RFC 2119) (RFC 8174) тогда и только тогда, когда они указаны заглавными буквами, как показано здесь.

Авторские права (Copyright (c) 2025) принадлежат IETF Trust и лицам, указанным в качестве авторов кода. Все права защищены.

Распространение и использование в исходной или двоичной форме с изменениями или без таковых разрешено в соответствии с лицензией Simplified BSD, изложенной в разделе 4.с IETF Trust's Legal Provisions применительно к документам IETF (<http://trustee.ietf.org/license-info>).

Эта версия данного модуля YANG является частью RFC 9911, где правовые вопросы рассмотрены более полно.";

```
revision 2025-12-22 {
  description
    "Этот выпуск документа добавляет указанные ниже типы данных:
    - yang:date
    - yang:date-no-zone
    - yang:time
    - yang:time-no-zone
    - yang:hours32
    - yang:minutes32
    - yang:seconds32
    - yang:centiseconds32
    - yang:milliseconds32
    - yang:microseconds32
    - yang:microseconds64
    - yang:nanoseconds32
    - yang:nanoseconds64
    - yang:language-tag
    Определение yang-identifier приведено в соответствии с YANG
    1.1 и типами, представляющими время с високосными секундами.
    Представление часовых поясов приведено в соответствии с RFC
    9557. Улучшены некоторые операторы description и pattern.";
  reference
    "RFC 9911: Common YANG Data Types";
}
revision 2013-07-15 {
  description
    "Этот выпуск документа добавляет указанные ниже типы данных:
    - yang:yang-identifier
    - yang:hex-string
    - yang:uuid
    - yang:dotted-quad";
  reference
    "RFC 6991: Common YANG Data Types";
}
revision 2010-09-24 {
  description
    "Исходный выпуск.";
  reference
    "RFC 6021: Common YANG Data Types";
}

/**** Набор типов для счётчиков 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 счёт
    повторяется с нуля.

    Для счётчиков не определены 'начальные' значения, поэтому
    одиночное значение счётчика (обычно) не имеет смысла. Разрывы
    в монотонном росте значений счётчиков 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 счётчиков типа counter32 не в процессе инициализации
    следует определить соответствующий узел схемы подходящего типа
    для указания последнего разрыва счётчика.

    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 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 неопределённом 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 счёт повторяется с нуля.

Для счётчиков не определены начальные значения, поэтому одиночное значение счётчика (обычно) не имеет смысла. Разрывы в монотонном росте значений счётчиков 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 счётчиков типа counter64 не в процессе инициализации следует определить соответствующий узел схемы подходящего типа для указания последнего разрыва счётчика.

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 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 неопределённом 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 (растёт).

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 (растёт).

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

/** Набор типов, связанных с идентификаторами */

typedef object-identifier {
    type string {
        pattern '((([0-1](\.[1-3]?[0-9]))|(2\.(0|([1-9][0-9]*))))'
            + '(\.(0|([1-9][0-9]*)))+';
    }
    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
        "ISO 9834-1: Information technology -- Procedures for the
        operation of object identifier registration authorities --
        Part 1: General procedures and top arcs of the international
        object identifier tree";
}

typedef object-identifier-128 {
    type object-identifier {
        pattern '[0-9]*(\.[0-9]*){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)";
}

/** Набор типов для дат и времени */

typedef date-and-time {
    type string {
        pattern
            '[0-9]{4}-(1[0-2]|0[1-9])-(0[1-9]|[1-2][0-9]|3[0-1])'
            + 'T(0[0-9]|1[0-9]|2[0-3]):[0-5][0-9]:([0-5][0-9]|60)';
    }
}

```

```

+ '(\.[0-9]+)?'
+ '(Z|[\+\-]((1[0-3]|0[0-9]):([0-5][0-9])|14:00))?' ;
}
description
"Тип date-and-time является профилем стандарта ISO 8601
для представления дат и времени с использованием григорианского
календаря. Профиль определен date-time в параграфе 5.6 RFC 3339
и обновлен в разделе 2 RFC 9557. Значение 60 допускается лишь в
в случае високосных секунд.

Тип date-and-time совместим с типом dateTime схемы XML с учётом
приведённых ниже исключений.

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

(b) Значение time-offset Z указывает, что date-and-time
задано в UTC и локальный часовой пояс не известен.
Значение time-offset +00:00 указывает, что date-and-time
задано в UTC и локальным часовым поясом тоже служит UTC
(см. раздел 2 в RFC 9557).

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 (DST).
В каноническом формате значений date-and-time в UTC с
неизвестным часовым поясом СЛЕДУЕТ использовать
time-offset Z и МОЖНО MAY применять -00:00 для совместимости
с прежними версиями.";
reference
"ISO 8601: Data elements and interchange formats -- Information
interchange -- Representation of dates and times
RFC 3339: Date and Time on the Internet: Timestamps
RFC 9557: Date and Time on the Internet: Timestamps
with Additional Information
RFC 2579: Textual Conventions for SMIV2
XSD-TYPES: XML Schema Definition Language (XSD) 1.1
Part 2: Datatypes";
}

typedef date {
type string {
pattern '[0-9]{4}-(1[0-2]|0[1-9])-(0[1-9]|[1-2][0-9]|3[0-1])'
+ '(Z|[\+\-]((1[0-3]|0[0-9]):([0-5][0-9])|14:00))?' ;
}
description
"Тип данных date представляет временной интервал в сутках (24
24 часа) и может включать часовой пояс.

Тип date совместим с типом date в схеме XML с приведёнными
ниже ограничениями:

(a) Тип date не допускает отрицательные значения лет.

(b) Значение time-offset Z указывает, что date задано в UTC
и локальный часовой пояс не известен. Значение time-offset
+00:00 указывает, что date задано в UTC и локальным
часовым поясом тоже служит UTC (см. раздел 2 в RFC 9557).

Канонический формат для значений date 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 должным
образом. Такие изменения могут быть периодическими в результате
перехода на летнее время (DST). В каноническом формате значений
date в UTC с неизвестным часовым поясом используется
time-offset Z.";
reference
"RFC 3339: Date and Time on the Internet: Timestamps
RFC 9557: Date and Time on the Internet: Timestamps
with Additional Information
XSD-TYPES: XML Schema Definition Language (XSD) 1.1
Part 2: Datatypes";
}

typedef date-no-zone {
type date {
pattern '[0-9]{4}-(1[0-2]|0[1-9])-(0[1-9]|[1-2][0-9]|3[0-1])' ;
}
}

```

```
description
  "Тип date-no-zone представляет дату без часового пояса.";
}

typedef time {
  type string {
    pattern
      '(0[0-9]|1[0-9]|2[0-3]):[0-5][0-9]:([0-5][0-9]|60)'
      + '(\.[0-9]+)?'
      + '(Z|[\+\-]((1[0-3]|0[0-9]):[0-5][0-9]|14:00))?';
  }
  description
    "Тип time представляет момент времени, повторяющийся каждый
    день, и может включать часовой пояс. Значение 60 допускается
    только для високосных секунд.

    Тип time совместим с типом time схемы XML с одним исключением:

    (a) Значение time-offset Z указывает, что time задано в UTC
    и локальный часовой пояс не известен. Значение time-offset
    +00:00 указывает, что time задано в UTC и локальным
    часовым поясом тоже служит UTC (см. раздел 2 в RFC 9557).

    Канонический формат для значений 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 time должным
    образом. Такие изменения могут быть периодическими в результате
    перехода на летнее время (DST). В каноническом формате значений
    time в UTC с неизвестным часовым поясом используется
    time-offset Z.";
  reference
    "RFC 3339: Date and Time on the Internet: Timestamps
    RFC 9557: Date and Time on the Internet: Timestamps
    with Additional Information
    XSD-TYPES: XML Schema Definition Language (XSD) 1.1
    Part 2: Datatypes";
}

typedef time-no-zone {
  type time {
    pattern
      '(0[0-9]|1[0-9]|2[0-3]):[0-5][0-9]:([0-5][0-9]|60)'
      + '(\.[0-9]+)?';
  }
  description
    "Тип time-no-zone представляет время без часового пояса.";
}

typedef hours32 {
  type int32;
  units "hours";
  description
    "Период времени, заданный в часах.

    Допустимые значения лежат в диапазоне [-89478485 дней
    08:00:00 часов - 89478485 дней 07:00:00 часов].

    Для этого типа следует ограничивать диапазон, если приемлемы
    только неотрицательные интервалы (0..max).";
}

typedef minutes32 {
  type int32;
  units "minutes";
  description
    "Период времени, заданный в минутах.

    Допустимые значения лежат в диапазоне [-1491308 дней 2:08:00 -
    1491308 дней 2:07:00].

    Для этого типа следует ограничивать диапазон, если приемлемы
    только неотрицательные интервалы (0..max).";
}

typedef seconds32 {
  type int32;
  units "seconds";
  description
    "Период времени, заданный в секундах.

    Допустимые значения лежат в диапазоне [-24855 дней 03:14:08 -
    24855 дней 03:14:07].

    Для этого типа следует ограничивать диапазон, если приемлемы
    только неотрицательные интервалы (0..max).";
}
```

```
}

typedef centiseconds32 {
    type int32;
    units "centiseconds";
    description
        "Период времени, заданный в сотых долях секунды.

        Допустимые значения лежат в диапазоне [-248 дней 13:13:56 -
        248 дней 13:13:56].

        Для этого типа следует ограничивать диапазон, если приемлемы
        только неотрицательные интервалы (0..max).";
}

typedef milliseconds32 {
    type int32;
    units "milliseconds";
    description
        "Период времени, заданный в миллисекундах.

        Допустимые значения лежат в диапазоне [-24 дня 20:31:23 -
        24 дня 20:31:23].

        Для этого типа следует ограничивать диапазон, если приемлемы
        только неотрицательные интервалы (0..max).";
}

typedef microseconds32 {
    type int32;
    units "microseconds";
    description
        "Период времени, заданный в микросекундах.

        Допустимые значения лежат в диапазоне [-00:35:47 to 00:35:47].

        Для этого типа следует ограничивать диапазон, если приемлемы
        только неотрицательные интервалы (0..max).";
}

typedef microseconds64 {
    type int64;
    units "microseconds";
    description
        "Период времени, заданный в микросекундах.

        Допустимые значения лежат в диапазоне [-106751991 дней 04:00:54
        - 106751991 дней 04:00:54].

        Для этого типа следует ограничивать диапазон, если приемлемы
        только неотрицательные интервалы (0..max).";
}

typedef nanoseconds32 {
    type int32;
    units "nanoseconds";
    description
        "Период времени, заданный в наносекундах.

        Допустимые значения лежат в диапазоне [-00:00:02 to 00:00:02].

        Для этого типа следует ограничивать диапазон, если приемлемы
        только неотрицательные интервалы (0..max).";
}

typedef nanoseconds64 {
    type int64;
    units "nanoseconds";
    description
        "Период времени, заданный в наносекундах.

        Допустимые значения лежат в диапазоне [-106753 дней 23:12:44 -
        106752 дней 0:47:16].

        Для этого типа следует ограничивать диапазон, если приемлемы
        только неотрицательные интервалы (0..max).";
}

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 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,
        определённого с использованием этого типа. Когда конкретное
        вхождение происходит до того, как связанный атрибут timeticks
        в последний раз был равен нулю, 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.
        Отметим, что этот тип не может представлять адреса с размером,
        отличным от IEEE 802 MAC и для них можно применять тип
        phys-address.

        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";
}

/**** Тип, связанный с XML ****/

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";
}

/**** Строковые типы ****/

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, разделёнными 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, определённом в RFC 4122. В каноническом варианте
    применяются строчные буквы.

    Ниже приведён пример строкового представления UUID
    f81d4fae-7dec-11d0-a765-00a0c91e6bf6";
  reference
    "RFC 9562: Universally Unique IDentifiers (UUIDs)";
}

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, разделёнными точками (full stop).";
}

typedef language-tag {
  type string;
  description
    "Тег языка в соответствии с RFC 5646 (BCP 47). В каноническом
    представлении применяются строчные буквы.

    Значения этого типа должны быть корректно сформированными
    тегами языка в соответствии с BCP 47. Реализации МОГУТ
    дополнительно ограничивать принимаемые значения 'проверяющим'
    процессором, как указано в BCP 47.

    Каноническое представление значений этого типа соответствует
    текстовому соглашению SMiv2 LangTag с учётом ограничений по
    длине, принятых для LangTag.";
  reference
    "RFC 5646: Tags for Identifying Languages
    RFC 5131: A MIB Textual Convention for Language Tags";
}

/** Набор типов, связанных с YANG */

typedef yang-identifier {
  type string {
    length "1..max";
    pattern '[a-zA-Z_][a-zA-Z0-9\-\_]*';
  }
  description
    "Строка идентификатора YANG в соответствии с правилом
    'identifier' раздела 14 в RFC 7950. Идентификатор должен
    начинаться с буквы или символа подчёркивания, за которым
    следует произвольный набор букв, цифр, символов подчёркивания,
    дефисов и точек.

    Это определение соответствует YANG 1.1 (RFC 7950). В RFC 6991
    определение исключало строки, начинающиеся с xml (в любой
    комбинации регистров), как было задано для YANG 1 в RFC 6020.
    Если данный тип применяется в контексте YANG 1, следует
    соблюдать данное ограничение.";
  reference
    "RFC 7950: The YANG 1.1 Data Modeling Language
    RFC 6991: Common YANG Data Types
    RFC 6020: YANG - A Data Modeling Language for the
    Network Configuration Protocol (NETCONF)";
}
}
<CODE ENDS>

```

4. Типы для стека протоколов Internet

Модуль YANG ietf-inet-types ссылается на [RFC0768], [RFC0791], [RFC0952], [RFC1034], [RFC1123], [RFC1930], [RFC2317], [RFC2474], [RFC2780], [RFC2782], [RFC3289], [RFC3305], [RFC3595], [RFC3927], [RFC3986], [RFC4001],

[RFC4007], [RFC4271], [RFC4291], [RFC4340], [RFC4592], [RFC5017], [RFC5322], [RFC5890], [RFC5952], [RFC6532], [RFC6793], [RFC8200], [RFC9260], [RFC9293], [RFC9499].

```
<CODE BEGINS> file "ietf-inet-types@2025-12-22.yang"
module ietf-inet-types {
  namespace "urn:ietf:params:xml:ns:yang:ietf-inet-types";
  prefix inet;

  organization
    "IETF Network Modeling (NETMOD) Working Group";
  contact
    "WG Web: <https://datatracker.ietf.org/wg/netmod/>
    WG List: <mailto:netmod@ietf.org>

    Editor: Jürgen Schönwälder
           <mailto:jschoenwaelder@constructor.university>";
  description
    "Этот модуль содержит набор производных типов данных YANG
    для адресов Internet и связанных элементов.

    Ключевые слова ДОЛЖНО, НЕДОПУСТИМО, ТРЕБУЕТСЯ, НУЖНО, НЕ НУЖНО,
    СЛЕДУЕТ, НЕ СЛЕДУЕТ, РЕКОМЕНДУЕТСЯ, НЕ РЕКОМЕНДУЕТСЯ, МОЖНО,
    НЕОБЯЗАТЕЛЬНО в этом документе трактуются в соответствии с
    ВСП 14 (RFC 2119) (RFC 8174) тогда и только тогда, когда они
    указаны заглавными буквами, как показано здесь.

    Авторские права (Copyright (c) 2025) 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 упрощённой 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 9911, где
    правовые аспекты выражены более полно.";

  revision 2025-12-22 {
    description
      "В этом выпуске добавлены указанные ниже типы данных:
      - inet:ip-address-and-prefix
      - inet:ipv4-address-and-prefix
      - inet:ipv6-address-and-prefix
      - inet:protocol-number
      - inet:upper-layer-protocol-number
      - inet:host-name
      - inet:email-address
      - inet:ip-address-link-local
      - inet:ipv4-address-link-local
      - inet:ipv6-address-link-local
      Определение inet:host было изменено с использованием
      inet:host-name вместо inet:domain-name. Улучшены некоторые
      операторы pattern.";
    reference
      "RFC 9911: Common YANG Data Types";
  }
  revision 2013-07-15 {
    description
      "В этом выпуске добавлены типы данных:
      - inet:ip-address-no-zone
      - inet:ipv4-address-no-zone
      - inet:ipv6-address-no-zone";
    reference
      "RFC 6991: Common YANG Data Types";
  }
  revision 2010-09-24 {
    description
      "Исходный выпуск.";
    reference
      "RFC 6021: Common YANG Data Types";
  }
}

/*** Набор типов для полей протоколов ***/

typedef ip-version {
  type enumeration {
    enum unknown {
      value 0;
      description
        "An unknown or unspecified version of the Internet
        Protocol.";
    }
    enum ipv4 {
      value 1;
      description

```

```

    " Протокол IPv4 в соответствии с RFC 791.";
}
enum ipv6 {
    value 2;
    description
        " Протокол IPv6 в соответствии с RFC 8200.";
}
}
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 8200: 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 SMIPv2.";
    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 представляет идентификатор или метку
        потока в заголовке пакета IPv6, которые могут служить для
        различения потоков трафика.

        Набор значений и семантика этого типа эквивалентны
        текстовому соглашению IPv6FlowLabel в SMIPv2.";
    reference
        "RFC 3595: Textual Conventions for IPv6 Flow Label
        RFC 8200: Internet Protocol, Version 6 (IPv6) Specification";
}

typedef port-number {
    type uint16 {
        range "0..65535";
    }
    }
    description
        "Тип port-number представляет 16-битовый номер порта 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
        путём создания субтипа для port-number без 0.

        Набор значений и семантика для этого типа эквивалентны
        текстовому соглашению InetPortNumber в SMIPv2.";

    reference
        "RFC 768: User Datagram Protocol
        RFC 9293: Transmission Control Protocol (TCP)
        RFC 9260: Stream Control Transmission Protocol
        RFC 4340: Datagram Congestion Control Protocol (DCCP)
        RFC 4001: Textual Conventions for Internet Network Addresses";
}

typedef protocol-number {
    type uint8;
    description
        "Тип protocol-number представляет 8-битовый номер протокола
        Internet, передаваемый в поле protocol заголовка IPv4 или в
        поле next header заголовка IPv6.

```

Номера протоколов выделяются IANA. Текущий список номеров доступен по ссылке [<https://www.iana.org/>](https://www.iana.org/)."

reference

"RFC 791: Internet Protocol

RFC 8200: Internet Protocol, Version 6 (IPv6) Specification";

}

typedef upper-layer-protocol-number {

type protocol-number;

description

"Поле upper-layer-protocol-number представляет протокол вышележащего уровня в пакете IP. Для пакетов IPv6 с заголовками расширения этот номер передаётся в последнем поле next header цепочки заголовков расширения IPv6.";

reference

"RFC 791: Internet Protocol

RFC 8200: Internet Protocol, Version 6 (IPv6) Specification";

}

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

typedef as-number {

type uint32;

description

"Тип as-number представляет номера автономных систем (AS). 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 SMIPv2.";

reference

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

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 для адресов IP и имён хостов ***/

typedef ip-address {

type union {

type ipv4-address;

type ipv6-address;

}

description

"Тип ip-address представляет адрес IP независимо от версии IP. Формат текстового представления подразумевает 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])'
+ '(%.)?';

}

description

"Тип ipv4-address представляет адрес IPv4 в десятичной нотации с разделением точками. Адрес IPv4 может включать индекс зоны, отделённый символом %. Если в системе применяются имена зон, не представляемые символами UTF-8, реализации нужно иметь механизм преобразования локального имени в UTF-8. Задание такого механизма выходит за рамки этого документа.

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, используется принятая по умолчанию зона устройства.

Каноническим 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]))|'
            + '(%[A-Za-z0-9][A-Za-z0-9\-\.\_~/]*)?';
        pattern '(([^:]+:){6}([^\:]+\:[^\:]+|(\.\*\.\*\.\*\.\*\.\*)))|'
            + '(((^\:)+)*[^\:]+)?::(((^\:)+)*[^\:]+)?'
            + '(%\.)?';
    }
    description
        "Тип ipv6-address представляет адрес IPv6 в полной, сокращённой
        или сокращённой смешанной нотации. Адрес IPv6 может включать
        индекс зоны, отделённый символом %. Если в системе применяются
        имена зон, не представляемые символами UTF-8, реализации нужно
        иметь механизм преобразования локального имени в UTF-8.
        Задание такого механизма выходит за рамки этого документа.

        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, определённое 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 ipv4-address-no-zone;
        type ipv6-address-no-zone;
    }
    description
        "Тип ip-address-no-zone представляет адрес IP независимо от
        версии IP. Формат текстового представления подразумевает
        версию IP. Этот тип не поддерживает ограниченные (scoped)
        адреса, поскольку он не включает идентификатор зоны."
    reference
        "RFC 4007: IPv6 Scoped Address Architecture";
}

typedef ipv4-address-no-zone {
    type 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 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-address-link-local {
    type union {
        type ipv4-address-link-local;
        type ipv6-address-link-local;
    }
    description
        "Тип ip-address-link-local представляет адрес link-local
        независимо от версии IP. Формат текстового представления
        подразумевает версию IP."
}

typedef ipv4-address-link-local {

```

```

type ipv4-address {
    pattern '169\.254\..*';
}
description
    "Тип ipv4-address-link-local представляет адрес link-local IPv4
    в префиксе 169.254.0.0/16, как указано в параграфе 2.1
    RFC 3927.";
reference
    "RFC 3927: Dynamic Configuration of IPv4 Link-Local Addresses";
}

typedef ipv6-address-link-local {
    type ipv6-address {
        pattern '[fF][eE][89aAbB][0-9a-fA-F]:.*';
    }
    description
        "Тип ipv6-address-link-local представляет адрес link-local IPv6
        в префиксе fe80::/10, как указано в параграфе 2.4 RFC 4291.";
    reference
        "RFC 4291: IP Version 6 Addressing Architecture";
}

typedef ip-prefix {
    type union {
        type ipv4-prefix;
        type ipv6-prefix;
    }
    description
        "Тип ip-prefix представляет префикс IP независимо от версии IP.
        Формат текстового представления подразумевает 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.";

        Определение ipv4-prefix не требует установки значения 0 для
        битов, не являющихся частью префикса. Однако реализации должны
        возвращать значения в каноническом формате, который требует
        установки таких битов в 0. Это означает, что префикс
        192.0.2.1/24 должен восприниматься как корректный, но
        преобразовываться в каноническое значение 192.0.2.0/24.";
}

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 представляются в соответствии
        с разделом 4 RFC 5952.";

        Определение ipv6-prefix не требует установки значения 0 для

```


имеют более строгий синтаксис (см. RFC 952), чем рекомендации DNS в RFC 1034 и 1123. Узлам схемы, представляющим имена хостов, следует применять тип `host-name` взамен `domain-type`.

Размер имён 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 имён 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 2317: Classless IN-ADDR.ARPA delegation
RFC 2782: A DNS RR for specifying the location of services
(DNS SRV)
RFC 4592: The Role of Wildcards in the Domain Name System
RFC 5890: Internationalized Domain Names in Applications
(IDNA): Definitions and Document Framework
RFC 9499: DNS Terminology";
```

```
}
```

```
typedef host-name {
    type domain-name {
        length "2..max";
        pattern '[a-zA-Z0-9\-\.\.]+';
    }
    description
    "Тип host-name представляет полное доменное имя хоста. Размер
имени должен быть не менее 2 символов (см. RFC 952), которыми
могут быть буквы, цифры и символы дефиса, разделённые точками
(см. RFC 1123 и 952).";
    reference
    "RFC 952: DoD Internet Host Table Specification
RFC 1123: Requirements for Internet Hosts -- Application
and Support";
}
```

```
typedef host {
    type union {
        type ip-address;
        type host-name;
    }
    description
    "Тип host представляет адрес IP или доменное имя (FQDN).";
}
```

```
typedef uri {
    type string {
        pattern '[a-z][a-z0-9+.-]*:.*';
    }
    description
    "Тип uri представляет идентификаторы URI1, заданные в STD 66.
```

Объекты, использующие тип `uri`, ДОЛЖНЫ указываться в кодировке 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 заменяются символами, для независимых от регистра символов устанавливается нижний регистр, за исключением шестнадцатеричных цифр, которые нормализуются в верхний регистр как указано в параграфах 6.2.2.1 RFC 3986.

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 разрешённые схемы. Например, схемы `data:` и `urn:` могут не подойти.

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

¹Uniform Resource Identifier - унифицированный идентификатор ресурса.

```

Набор значений и семантика этого типа эквивалентны
текстовому соглашению Uri SMIV2, определённом 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)";
}

typedef email-address {
type string {
pattern '.*@.*';
}
description
"Тип email-address представляет адреса электронной почты
в международном формате (internationalized).

Формат адресов электронной почты задан правилом addr-spec
ABNF в параграфе 3.4.1 RFC 5322. Этот формат был расширен в
RFC 6532 для поддержки адресов на других языках. Реализации
ДОЛЖНЫ поддерживать расширения интернационализации RFC 6532.
Поддержка устаревших obs-local-part, obs-domain, obs-qttext
из RFC 5322 не требуется.

В доменной части могут применяться метки A и U (см RFC 5890).
В каноническом формате доменной части используются символы
нижнего регистра и метки U (RFC 5890), когда это применимо.";
reference
"RFC 5322: Internet Message Format
RFC 5890: Internationalized Domain Names in Applications
(IDNA): Definitions and Document Framework
RFC 6532: Internationalized Email Headers";
}
}
<CODE ENDS>

```

5. Взаимодействие с IANA

В этом документе используются URI для ietf-yang-types и ietf-inet-types в реестре IETF XML Registry [RFC3688].

В соответствии с этим документом агентство IANA обновило реестр YANG Module Names с указанием данного вместо [RFC6991] для модулей ietf-yang-types и ietf-inet-types. В соответствии с форматом [RFC6020] эти представления имеют вид:

```

Name: ietf-yang-types
Namespace: urn:ietf:params:xml:ns:yang:ietf-yang-types
Prefix: yang
Reference: RFC 9911

Name: ietf-inet-types
Namespace: urn:ietf:params:xml:ns:yang:ietf-inet-types
Prefix: inet
Reference: RFC 9911

```

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 [RFC7950], применимы и к этому документу.

7. Литература

7.1. Нормативные документы

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, [RFC 2119](https://www.rfc-editor.org/info/rfc2119), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", [RFC 3339](https://www.rfc-editor.org/info/rfc3339), DOI 10.17487/RFC3339, July 2002, <<https://www.rfc-editor.org/info/rfc3339>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, [RFC 3688](https://www.rfc-editor.org/info/rfc3688), DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, [RFC 3986](https://www.rfc-editor.org/info/rfc3986), DOI 10.17487/RFC3986, January 2005, <<https://www.rfc-editor.org/info/rfc3986>>.
- [RFC4007] Deering, S., Haberman, B., Jinmei, T., Nordmark, E., and B. Zill, "IPv6 Scoped Address Architecture", [RFC 4007](https://www.rfc-editor.org/info/rfc4007), DOI 10.17487/RFC4007, March 2005, <<https://www.rfc-editor.org/info/rfc4007>>.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", [RFC 4291](https://www.rfc-editor.org/info/rfc4291), DOI 10.17487/RFC4291, February 2006, <<https://www.rfc-editor.org/info/rfc4291>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](https://www.rfc-editor.org/info/rfc6020), DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.

- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", [RFC 7950](#), DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC9499] Hoffman, P. and K. Fujiwara, "DNS Terminology", BCP 219, [RFC 9499](#), DOI 10.17487/RFC9499, March 2024, <<https://www.rfc-editor.org/info/rfc9499>>.
- [RFC9557] Sharma, U. and C. Bormann, "Date and Time on the Internet: Timestamps with Additional Information", [RFC 9557](#), DOI 10.17487/RFC9557, April 2024, <<https://www.rfc-editor.org/info/rfc9557>>.
- [RFC9562] Davis, K., Peabody, B., and P. Leach, "Universally Unique Identifiers (UUIDs)", [RFC 9562](#), DOI 10.17487/RFC9562, May 2024, <<https://www.rfc-editor.org/info/rfc9562>>.
- [XPATH] Clark, J., Ed. and S. DeRose, Ed., "XML Path Language (XPath) Version 1.0", W3C Recommendation, 16 November 1999, <<http://www.w3.org/TR/xpath-10>>.
- [XSD-TYPES] Peterson, D., Ed., Gao, S., Ed., Malhotra, A., Ed., Sperberg-McQueen, C., Ed., and H. S. Thompson, Ed., "W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes", W3C Recommendation, 5 April 2012, <<https://www.w3.org/TR/xmlschema11-2/>>.

7.2. Дополнительная литература

- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, [RFC 768](#), DOI 10.17487/RFC0768, August 1980, <<https://www.rfc-editor.org/info/rfc768>>.
- [RFC0791] Postel, J., "Internet Protocol", STD 5, [RFC 791](#), DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/info/rfc791>>.
- [RFC0952] Harrenstien, K., Stahl, M., and E. Feinler, "DoD Internet host table specification", [RFC 952](#), DOI 10.17487/RFC0952, October 1985, <<https://www.rfc-editor.org/info/rfc952>>.
- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, [RFC 1034](#), DOI 10.17487/RFC1034, November 1987, <<https://www.rfc-editor.org/info/rfc1034>>.
- [RFC1123] Braden, R., Ed., "Requirements for Internet Hosts - Application and Support", STD 3, [RFC 1123](#), DOI 10.17487/RFC1123, October 1989, <<https://www.rfc-editor.org/info/rfc1123>>.
- [RFC1930] Hawkinson, J. and T. Bates, "Guidelines for creation, selection, and registration of an Autonomous System (AS)", BCP 6, [RFC 1930](#), DOI 10.17487/RFC1930, March 1996, <<https://www.rfc-editor.org/info/rfc1930>>.
- [RFC2317] Eidnes, H., de Groot, G., and P. Vixie, "Classless IN-ADDR.ARPA delegation", BCP 20, [RFC 2317](#), DOI 10.17487/RFC2317, March 1998, <<https://www.rfc-editor.org/info/rfc2317>>.
- [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](#), DOI 10.17487/RFC2474, December 1998, <<https://www.rfc-editor.org/info/rfc2474>>.
- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIPv2)", STD 58, [RFC 2578](#), DOI 10.17487/RFC2578, April 1999, <<https://www.rfc-editor.org/info/rfc2578>>.
- [RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIPv2", STD 58, [RFC 2579](#), DOI 10.17487/RFC2579, April 1999, <<https://www.rfc-editor.org/info/rfc2579>>.
- [RFC2780] Bradner, S. and V. Paxson, "IANA Allocation Guidelines For Values In the Internet Protocol and Related Headers", BCP 37, [RFC 2780](#), DOI 10.17487/RFC2780, March 2000, <<https://www.rfc-editor.org/info/rfc2780>>.
- [RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", [RFC 2782](#), DOI 10.17487/RFC2782, February 2000, <<https://www.rfc-editor.org/info/rfc2782>>.
- [RFC2856] Bierman, A., McCloghrie, K., and R. Presuhn, "Textual Conventions for Additional High Capacity Data Types", [RFC 2856](#), DOI 10.17487/RFC2856, June 2000, <<https://www.rfc-editor.org/info/rfc2856>>.
- [RFC3289] Baker, F., Chan, K., and A. Smith, "Management Information Base for the Differentiated Services Architecture", [RFC 3289](#), DOI 10.17487/RFC3289, May 2002, <<https://www.rfc-editor.org/info/rfc3289>>.
- [RFC3305] Mealling, M., Ed. and R. Denenberg, Ed., "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](#), DOI 10.17487/RFC3305, August 2002, <<https://www.rfc-editor.org/info/rfc3305>>.
- [RFC3595] Wijnen, B., "Textual Conventions for IPv6 Flow Label", [RFC 3595](#), DOI 10.17487/RFC3595, September 2003, <<https://www.rfc-editor.org/info/rfc3595>>.
- [RFC3927] Cheshire, S., Aboba, B., and E. Guttman, "Dynamic Configuration of IPv4 Link-Local Addresses", [RFC 3927](#), DOI 10.17487/RFC3927, May 2005, <<https://www.rfc-editor.org/info/rfc3927>>.
- [RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", [RFC 4001](#), DOI 10.17487/RFC4001, February 2005, <<https://www.rfc-editor.org/info/rfc4001>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", [RFC 4271](#), DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", [RFC 4340](#), DOI 10.17487/RFC4340, March 2006, <<https://www.rfc-editor.org/info/rfc4340>>.

- [RFC4502] Waldbusser, S., "Remote Network Monitoring Management Information Base Version 2", RFC 4502, DOI 10.17487/RFC4502, May 2006, <<https://www.rfc-editor.org/info/rfc4502>>.
- [RFC4592] Lewis, E., "The Role of Wildcards in the Domain Name System", [RFC 4592](#), DOI 10.17487/RFC4592, July 2006, <<https://www.rfc-editor.org/info/rfc4592>>.
- [RFC5017] McWalter, D., Ed., "MIB Textual Conventions for Uniform Resource Identifiers (URIs)", RFC 5017, DOI 10.17487/RFC5017, September 2007, <<https://www.rfc-editor.org/info/rfc5017>>.
- [RFC5131] McWalter, D., Ed., "A MIB Textual Convention for Language Tags", RFC 5131, DOI 10.17487/RFC5131, December 2007, <<https://www.rfc-editor.org/info/rfc5131>>.
- [RFC5322] Resnick, P., Ed., "Internet Message Format", [RFC 5322](#), DOI 10.17487/RFC5322, October 2008, <<https://www.rfc-editor.org/info/rfc5322>>.
- [RFC5646] Phillips, A., Ed. and M. Davis, Ed., "Tags for Identifying Languages", BCP 47, RFC 5646, DOI 10.17487/RFC5646, September 2009, <<https://www.rfc-editor.org/info/rfc5646>>.
- [RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, DOI 10.17487/RFC5890, August 2010, <<https://www.rfc-editor.org/info/rfc5890>>.
- [RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, DOI 10.17487/RFC5952, August 2010, <<https://www.rfc-editor.org/info/rfc5952>>.
- [RFC6021] Schoenwaelder, J., Ed., "Common YANG Data Types", [RFC 6021](#), DOI 10.17487/RFC6021, October 2010, <<https://www.rfc-editor.org/info/rfc6021>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", [RFC 6241](#), DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6532] Yang, A., Steele, S., and N. Freed, "Internationalized Email Headers", RFC 6532, DOI 10.17487/RFC6532, February 2012, <<https://www.rfc-editor.org/info/rfc6532>>.
- [RFC6793] Vohra, Q. and E. Chen, "BGP Support for Four-Octet Autonomous System (AS) Number Space", [RFC 6793](#), DOI 10.17487/RFC6793, December 2012, <<https://www.rfc-editor.org/info/rfc6793>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", [RFC 6991](#), DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, [RFC 8200](#), DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC9260] Stewart, R., Tüxen, M., and K. Nielsen, "Stream Control Transmission Protocol", [RFC 9260](#), DOI 10.17487/RFC9260, June 2022, <<https://www.rfc-editor.org/info/rfc9260>>.
- [RFC9293] Eddy, W., Ed., "Transmission Control Protocol (TCP)", STD 7, [RFC 9293](#), DOI 10.17487/RFC9293, August 2022, <<https://www.rfc-editor.org/info/rfc9293>>.
- [ISO-8601] ISO/IEC, "Data elements and interchange formats -- Information interchange -- Representation of dates and times", ISO/IEC 8601:2000, December 2008, <<https://www.iso.org/standard/26780.html>>.
- [ISO-9834-1] ISO/IEC, "Information technology -- Procedures for the operation of object identifier registration authorities -- Part 1: General procedures and top arcs of the international object identifier tree", ISO/IEC 9834-1:2012, 2012, <<https://www.iso.org/standard/58055.html>>.
- [IEEE-802-2024] IEEE, "IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture", IEEE Std 802-2024, DOI 10.1109/IEEESTD.2025.10935844, March 2025, <<https://doi.org/10.1109/IEEESTD.2025.10935844>>.
- [Err4076] RFC Errata, Erratum ID 4076, RFC 6991, <<https://www.rfc-editor.org/errata/eid4076>>.
- [Err5105] RFC Errata, Erratum ID 5105, RFC 6991, <<https://www.rfc-editor.org/errata/eid5105>>.

Благодарности

В создании исходного документа, опубликованного как [RFC6021] приняли участие: Andy Bierman, Martin Björklund, Balazs Lengyel, David Partain, Phil Shafer.

Полезные замечания к черновым вариантам этого документа представили: Andy Bierman, Martin Björklund, Benoît Claise, Joel M. Halpern, Ladislav Lhotka, Lars-Johan Liman, Dan Romascanu.

Адрес автора

Jürgen Schönwälder (editor)

Constructor University

Email: jschoenwaelder@constructor.university

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru