

## A File Format for YANG Instance Data

Формат файла для экземпляра данных YANG

### Аннотация

Требуется документировать данные, определённые в моделях YANG, в процессе разработки, при реализации или недоступности действующего сервера. Этот документ задаёт стандартный формат файла для экземпляра данных YANG, который следует синтаксису и семантике имеющихся моделей YANG и аннотируется метаданными.

### Статус документа

Документ относится к категории Internet Standards Track.

Документ является результатом работы IETF<sup>1</sup> и представляет согласованный взгляд сообщества IETF. Документ прошёл открытое обсуждение и был одобрен для публикации IESG<sup>2</sup>. Дополнительную информацию о стандартах Internet можно найти 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/rfc9195>.

### Авторские права

Авторские права (Copyright (c) 2022) 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 упрощённой лицензией BSD, как указано в параграфе 4.e документа IETF Trust Legal Provisions, без каких-либо гарантий (как указано в Simplified BSD License).

## Оглавление

1. Введение.....	2
1.2. Терминология.....	2
1.2. Принципы.....	2
1.3. Доставка данных экземпляра.....	3
1.4. Жизненный цикл данных.....	3
2. Формат файла данных экземпляра.....	3
2.1. Задание схемы содержимого.....	3
2.1.1. Встраивание (Inline).....	4
2.1.2. Упрощённое встраивание.....	4
2.1.3. Метод URI.....	4
2.2. Примеры.....	4
2.2.1. Документирование возможностей сервера.....	4
2.2.2. Предварительно загруженные данные конфигурации по умолчанию.....	5
2.2.3. Сохранение данных диагностики.....	6
3. Модель YANG для данных экземпляра.....	6
3.1. Диаграмма дерева.....	6
3.2. Модель YANG.....	6
4. Вопросы безопасности.....	9
5. Взаимодействие с IANA.....	10
5.1. Регистрация URI.....	10
5.2. Регистрация модуля YANG.....	10
6. Литература.....	10
6.1. Нормативные документы.....	10
6.2. Дополнительная литература.....	10
Приложение А. Совместимость с прежними версиями.....	11
Приложение В. Варианты применения.....	11
В.1. Пример 1. Ранее документирование возможностей сервера.....	11
В.2. Пример 2. Предварительная загрузка данных.....	11
В.3. Пример 3. Документирование заводских установок.....	12
Благодарности.....	12
Адреса авторов.....	12

<sup>1</sup>Internet Engineering Task Force - комиссия по решению инженерных задач Internet.

<sup>2</sup>Internet Engineering Steering Group - комиссия по инженерным разработкам Internet.

## 1. Введение

Требуется документировать данные, определённые в моделях YANG, когда работающий сервер недоступен. Данные часто нужны в процессе разработки, при реализации или даже позже при недоступности работающего сервера. Для обеспечения такой (offline) доставки данных этот документ задаёт стандартный формат для наборов данных и файлов экземпляров данных YANG. Формат набора данных экземпляра задан в модуле YANG `ietf-yang-instance-data` (3. Модель YANG для данных экземпляра). Модель данных YANG в этом документе соответствует архитектуре хранилища данных управления сетью (Network Management Datastore Architecture или NMDA), заданной в [RFC8342].

Ниже приведён список уже реализованных и возможных вариантов применения.

### UC1

Документирование возможностей сервера.

### UC2

Предварительная загрузка заданных по умолчанию данных.

### UC3

Документирование заданных на производстве принятых по умолчанию данных.

### UC4

Сохранение конфигурации устройства, например, для резервной копии, архива или аудита.

### UC5

Сохранение данных диагностики.

### UC6

Возможность передачи экземпляра данных YANG в сообщениях обмена между процессами (inter-process communication или IPC).

### UC7

Использование заданных по умолчанию данных в качестве шаблона.

### UC8

Предоставление примеров данных в RFC и Internet-draft.

Первые три варианта применения описаны в Приложении В.

Имеется много вариантов применения экземпляров данных YANG. Этот документ не ограничивает будущее применения наборов данных экземпляров, поэтому не задаёт, когда и как используются экземпляры данных YANG. Предполагается, что будущие документы определяют конкретные варианты применения, а здесь лишь даны примеры.

## 1.2. Терминология

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не следует** (SHALL NOT), **следует** (SHOULD), **не нужно** (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.

### **Instance Data - данные экземпляра**

Набор созданных узлов данных.

### **Instance Data Set - набор данных экземпляра**

Именованный набор аннотированных метаданными элементов данных, которые могут служить экземпляром данных в дереве YANG.

### **Instance Data File - файл данных экземпляра**

Файл, содержащий набор данных экземпляра в формате, соответствующем правилам этого документа.

### **Content-schema - схема содержимого**

Набор модулей YANG с их выпусками, поддерживаемыми функциями (feature) и отклонениями (deviation) для которых набор данных экземпляра содержит данные.

### **Content-defining YANG Module - определяющий содержимое модуль YANG**

Отдельный модуль YANG из content-schema.

Термин «сервер» определён в [RFC8342].

## 1.2. Принципы

Ниже указаны базовые принципы форматирования данных экземпляра.

### P1

Нужно определить стандартные форматы кодирования на основе XML и JSON.

### P2

В данных экземпляра нужно применять имеющиеся правила кодирования данных YANG.

### P3

Нужно определить метаданные для набора данных экземпляра (2. Формат файла данных экземпляра).

### P4

Нужно разрешать включение в набор данных экземпляра YANG данных из разных модулей YANG.

### P5

Нужно разрешать включение в набор данных конфигурации, состояния и их комбинаций.

### P6

Нужно разрешать частичные наборы данных.

### P7

Формату экземпляра данных YANG нужно подходить для любых данных, которые могут быть определены в модулях YANG, и быть доступным для читателя независимо от реализации модуля сервером.

### P8

Нужна возможность сообщить идентификатор хранилища данных, с которым связан набор данных экземпляра.

### 1.3. Доставка данных экземпляра

Наборы данных экземпляров, созданные в какой-либо спецификации или проекте могут быть доступны без работающего сервера, например, путём загрузки с web-сайта производителя или через иной способ распространения документов.

Другие наборы данных экземпляра могут предоставляться самим сервером YANG, например, при документировании данных диагностики (UC5).

### 1.4. Жизненный цикл данных

Набор данных экземпляра YANG создаётся в конкретный момент времени. Если данные меняются после этого, набор больше не будет представлять их, пока не будет обновлён. Текущие значения можно извлечь в процессе работы через NETCONF/RESTCONF или получить, например, в уведомлении YANG-Push.

Время и способ возможного изменения данных следует указывать в операторе description набора данных экземпляра или иным способом, зависящим от реализации.

## 2. Формат файла данных экземпляра

Файл данных экземпляра YANG должен содержать один набор данных экземпляра без дополнительных данных.

Формат набора данных экземпляра задан в модуле YANG ietf-yang-instance-data. Набор состоит из заголовка и содержимого. Заголовок содержит метаданные для набора данных экземпляра, а content-data, заданные как узел anydata, - данные экземпляра, которые пользователь хочет документировать и/или предоставить. Синтаксис и семантику content-data определяет content-schema.

Заданы два формата на основе кодировок YANG XML и JSON. Форматы файлов обеспечиваются путём применения соответствующих правил кодирования XML или JSON к включённой в этот документ структуре YANG. По мере определения иных кодировок YANG (например, CBOR) могут быть заданы другие форматы данных экземпляра.

Элемент content-data **должен** соответствовать content-schema с учётом приведённых ниже исключений. Для content-data **нужно** следовать правилам кодирования XML [RFC7950] или JSON [RFC7951] и **должны** применяться символы UTF-8. В content-data **можно** включить:

- метаданные, определённые в [RFC7952];
- источник метаданных, как указано в [RFC8526] и [RFC8527];
- зависящие от реализации метаданные для отдельных узлов данных, неизвестные метаданные **должны** игнорироваться пользователями данных экземпляра, оставляя их для будущих целей.

Набор данных экземпляра **может** содержать данные для любого числа модулей YANG, а при необходимости **может** включать полные данные конфигурации и состояния для сервера. Принятые по умолчанию значения следует исключать по причине их бесполезности.

Данные конфигурации (config true) и рабочего состояния (config false) **можно** смешивать в файле данных экземпляра.

Файлы данных экземпляра **могут** содержать частичные наборы данных. Это означает, что ограничения mandatory, min-elements, require-instance true, must, when **можно** не соблюдать.

Для имени файла данных экземпляра **следует** использовать форму (нотация ABNF [RFC5234])

```
instance-data-set-name ["@" ( revision-date / timestamp ) ]
                        ( ".xml" / ".json" )
```

Примерами имён служат acme-router-modules.xml, acme-router-modules@2018-01-25.xml, acme-router-modules@2018-01-25T15\_06\_34\_3+01\_00.json.

При наличии узла name в заголовке данных экземпляра его значение **следует** использовать как instance-data-set-name в имени файла. При наличии revision-date в имени файла значение **должно** соответствовать формату листа revision-date в модели YANG. При наличии revision-date и имени файла и заголовке данных экземпляра дата выпуска в имени файла **должна** соответствовать последнему выпуску в наборе данных экземпляра. При наличии timestamp в имени файла метка **должна** соответствовать формату листа timestamp в модели YANG за исключением описанной ниже замены двоеточий (:). При наличии timestamp в заголовке и имени файла для метки в имени **следует** использовать timestamp в наборе данных экземпляра с заменой двоеточий символом подчёркивания (\_).

Метаданные со сведениями о наборе данных **должны** быть включены. Некоторые элементы метаданных определены в модуле YANG ietf-yang-instance-data, но **можно** применять другие элементы.

Метаданные **должны** включать версию формата данных экземпляра YANG (если она не задана явно, используется принятое по умолчанию значение). В метаданные **следует** включить:

- имя набора данных;
- спецификацию content-schema (т. е. узел content-schema);
- описание набора данных экземпляра, в которое **следует** включать сведения о возможности, условиях и способе изменения данных с течением срока работы сервера;
- индикацию включения заданных по умолчанию данных; для обработки таких данных применяются концепции [RFC6243], однако пользователи набора данных экземпляра не обязаны поддерживать [RFC6243].

### 2.1. Задание схемы содержимого

Для корректного понимания и применения набора данных экземпляра пользователю нужно знать схему содержимого (content-schema), которая может быть задана в отдельных документах или в самом наборе данных экземпляра. В последнем случае **должен** применяться один из указанных ниже методов.

**Встраивание (Inline)**

Включение нужной информации в набор данных экземпляра.

**Упрощённое встраивание**

Включение информации в набор данных экземпляра с использованием лишь name и revision-date из модулей.

**URI**

Включения URI со ссылкой на дугой файл данных экземпляра YANG. Этот файл будет использовать ту же content-schema, что и указанный файл данных экземпляра YANG (чтобы не повторять снова и снова).

Позднее могут быть добавлены другие методы, например, решения на основе пакетов YANG.

Отметим, что заданная схема содержимого (content-schema) лишь указывает набор модулей, использованных для определения этого набора данных экземпляра YANG. Иногда данные экземпляра могут применяться для сервера, поддерживающего иной набор модулей YANG (например, для предварительной загрузки принятых по умолчанию данных - UC2, где набор данных экземпляра YANG может не меняться при обновлении модулей на сервере). Возможность использования набора данных экземпляра с другой схемой содержимого зависит от многих факторов, включая различия в совместимости между схемами при рассмотрении модулей, выпусков, свойств (feature), отклонений (deviation), области действия данных экземпляра и т. п.

**2.1.1. Встраивание (Inline)**

Узел данных anydata inline-yang-library содержит данные экземпляра (в соответствии с ietf-yang-library@2019-01-04) [RFC8525], которые задают определяющие содержимое модули YANG, включая выпуск, поддерживаемые функции (feature), отклонения (deviation) и любые дополнительные данные. Пример встраивания приведён в параграфе 2.2.1.

**2.1.2. Упрощённое встраивание**

Набор данных экземпляра содержит список определяющих содержимое модулей YANG, включая даты их выпуска. Применение этого метода предполагает, что модули применяются без отклонений и поддерживаются все функции. Модули YANG, нужные лишь для соблюдения зависимостей импорта, **могут** не включаться в leaf-list. При их исключении потребителю набора данных экземпляра нужно применять правила языка YANG для восстановления импорта. Пример упрощённого встраивания приведён в параграфе 2.2.2.

**2.1.3. Метод URI**

В лист same-schema-as-file **нужно** включить URI другого файла данных экземпляра YANG. Текущий файл данных экземпляра будет использовать ту же схему content-schema, что и указанный файл. Указанный файл данных экземпляра **может** не включать content-data, если он служит лишь для задания content-schema. Если указанный ссылкой файл недоступен, схема содержимого будет неизвестна.

Метод URI удобен, когда пользователь хочет сократить издержки, связанные с заданием content-schema в каждом файле данных экземпляра, например, в варианте UC6, когда система создаёт файл диагностики каждую минуту для документирования состояния сервера. Пример метода URI представлен в параграфе 2.2.3.

**2.2. Примеры****2.2.1. Документирование возможностей сервера**

Пример файла acme-router-modules@2022-01-20.xml отражает вариант UC1 из раздела 1 и предоставляет список поддерживаемых модулей YANG и возможностей NETCONF на сервере. Для задания content-schema служит метод inline. В примере применяется фальцовка, описанная в [RFC8792].

===== NOTE: '\ ' line wrapping per RFC 8792 =====

```
<?xml version="1.0" encoding="UTF-8"?>
<instance-data-set xmlns=\
  "urn:ietf:params:xml:ns:yang:ietf-yang-instance-data">
  <name>acme-router-modules</name>
  <content-schema>
    <inline-yang-library>
      <modules-state \
        xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-library">
        <module>
          <name>ietf-yang-library</name>
          <revision>2019-01-04</revision>
        </module>
        <module>
          <name>ietf-netconf-monitoring</name>
          <revision>2010-10-04</revision>
        </module>
      </modules-state>
    </inline-yang-library>
  </content-schema>
  <revision>
    <date>2020-10-23</date>
    <description>Исходный выпуск</description>
  </revision>
  <description>Задаёт минимальный набор модулей, которые будет \
    включать каждый маршрутизатор acme-router. Этот набор будет \
    меняться лишь при обновлении выпуска программ.</description>
  <contact>info@acme.example.com</contact>
  <content-data>
    <modules-state \
      xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-library">
      <module>
        <name>ietf-yang-library</name>
```

```

<revision>2019-01-04</revision>
<namespace>\
  urn:ietf:params:xml:ns:yang:ietf-yang-library\
</namespace>
<conformance-type>implement</conformance-type>
</module>
<module>
  <name>ietf-system</name>
  <revision>2014-08-06</revision>
  <namespace>urn:ietf:params:xml:ns:yang:ietf-system</namespace>
  <feature>sys:authentication</feature>
  <feature>sys:local-users</feature>
  <deviation>
    <name>acme-system-ext</name>
    <revision>2018-08-06</revision>
  </deviation>
  <conformance-type>implement</conformance-type>
</module>
<module>
  <name>ietf-netconf-monitoring</name>
  <revision>2010-10-04</revision>
  <namespace>\
    urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring\
  </namespace>
  <conformance-type>implement</conformance-type>
</module>
<module>
  <name>ietf-yang-types</name>
  <revision>2013-07-15</revision>
  <namespace>urn:ietf:params:xml:ns:yang:ietf-yang-types\
  </namespace>
  <conformance-type>import</conformance-type>
</module>
<module>
  <name>acme-system-ext</name>
  <revision>2018-08-06</revision>
  <namespace>\
    urn:rdns:acme.example.com:oammodel:acme-system-ext\
  </namespace>
  <conformance-type>implement</conformance-type>
</module>
</modules-state>
<netconf-state \
  xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring">
  <capabilities>
    <capability>\
      urn:ietf:params:netconf:capability:validate:1.1\
    </capability>
  </capabilities>
</netconf-state>
</content-data>
</instance-data-set>

```

Рисунок 1.

## 2.2.2. Предварительно загруженные данные конфигурации по умолчанию

Пример файла `read-only-acm-rules@2022-01-20.xml` отражает вариант UC2 из раздела 1 и представляет принятый по умолчанию набор правил для оператора, которому доступно лишь считывание. Для указания `content-schema` применяется метод `simplified-inline`.

```

===== NOTE: '\ ' line wrapping per RFC 8792 =====
<?xml version="1.0" encoding="UTF-8"?>
<instance-data-set
  xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-instance-data">
  <name>read-only-acm-rules</name>
  <content-schema>
    <module>ietf-netconf-acm@2018-02-14</module>
  </content-schema>
  <revision>
    <date>2018-07-04</date>
    <description>Исходный выпуск</description>
  </revision>
  <description>Принятые по умолчанию правила доступа, разрешающие \
  лишь чтение. Набор правил будет меняться лишь при установке \
  нового выпуска программ.</description>
  <content-data>
    <nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
      <enable-nacm>true</enable-nacm>
      <read-default>deny</read-default>
      <exec-default>deny</exec-default>
      <rule-list>
        <name>read-only-role</name>
        <group>read-only-group</group>
        <rule>

```

```

<name>read-all</name>
<module-name>*</module-name>
<access-operation>read</access-operation>
<action>permit</action>
</rule>
</rule-list>
</nacm>
</content-data>
</instance-data-set>

```

Рисунок 2.

### 2.2.3. Сохранение данных диагностики

Пример файла `acme-router-netconf-diagnostics@2018-01-25T17_00_38Z.json` отражает вариант UC5 из раздела 1. Набор данных экземпляра содержит статистику сервера NETCONF, отдаваемую сервером каждые 15 минут. Новый набор данных создаётся периодически много раз в день и дата выпуска не имеет смысла, поэтому включена метка времени.

```

===== NOTE: '\ ' line wrapping per RFC 8792 =====
{
  "ietf-yang-instance-data:instance-data-set": {
    "name": "acme-router-netconf-diagnostics",
    "content-schema": {
      "same-schema-as-file": "file:///acme-diagnostics-schema.json"
    },
    "timestamp": "2018-01-25T17:00:38Z",
    "description": ["Статистика NETCONF. \
      Данные могут меняться каждый раз."],
    "content-data": {
      "ietf-netconf-monitoring:netconf-state": {
        "statistics": {
          "netconf-start-time": "2018-12-05T17:45:00Z",
          "in-bad-hellos": "32",
          "in-sessions": "397",
          "dropped-sessions": "87",
          "in-rpcs": "8711",
          "in-bad-rpcs": "408",
          "out-rpc-errors": "408",
          "out-notifications": "39007"
        }
      }
    }
  }
}

```

Рисунок 3.

## 3. Модель YANG для данных экземпляра

### 3.1. Диаграмма дерева

Диаграмма дерева использует нотацию, определённую в [RFC8340].

```

module: ietf-yang-instance-data
  structure instance-data-set:
    +--name? string
    +--format-version? string
    +--includes-defaults? enumeration
    +--content-schema
    | +--(content-schema-spec)?
    | | +--:(simplified-inline)
    | | | +--module* module-with-revision-date
    | | +--:(inline)
    | | | +--inline-yang-library <anydata>
    | | +--:(uri)
    | | +--same-schema-as-file? inet:uri
    +--description* string
    +--contact? string
    +--organization? string
    +--datastore? ds:datastore-ref
    +--revision* [date]
    | +--date string
    | +--description? string
    +--timestamp? yang:date-and-time
    +--content-data? <anydata>

```

### 3.2. Модель YANG

Этот модуль YANG импортирует определения типов из [RFC6991], [RFC6243], идентификаторы из [RFC8342] и расширение structure из [RFC8791]. Модуль также ссылается на [RFC8525].

```

<CODE BEGINS> file "ietf-yang-instance-data@2022-02-17.yang"
module ietf-yang-instance-data {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-yang-instance-data";
  prefix yid;

```

```

import ietf-yang-structure-ext {
  prefix sx;
  reference
    "RFC 8791: YANG Data Structure Extensions";
}
import ietf-datastores {
  prefix ds;
  reference
    "RFC 8342: Network Management Datastore Architecture (NMDA)";
}
import ietf-inet-types {
  prefix inet;
  reference
    "RFC 6991: Common YANG Data Types";
}
import ietf-yang-types {
  prefix yang;
  reference
    "RFC 6991: Common YANG Data Types";
}
import ietf-netconf-with-defaults {
  prefix ncwd;
  reference
    "RFC 6243: With-defaults Capability for NETCONF";
}

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

  Author: Balazs Lengyel
    <mailto:balazs.lengyel@ericsson.com>

  Author: Benoit Claise
    <mailto:benoit.claise@huawei.com>;
description
  "The module defines the structure and content of YANG
  instance data sets.

  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) 2022) принадлежат IETF Trust и
  лицам, указанным как авторы. Все права защищены.

  Распространение и применение модуля в исходной или двоичной
  форме с изменениями или без таковых разрешено в соответствии с
  лицензией Simplified BSD License, изложенной в параграфе 4.c
  IETF Trust's Legal Provisions Relating to IETF Documents
  (https://trustee.ietf.org/license-info).

  Эта версия модуля YANG является частью RFC 9195, где правовые
  аспекты приведены более полно.";

revision 2022-02-17 {
  description
    "Исходный выпуск.";
  reference
    "RFC 9195: YANG Instance Data File Format";
}

typedef module-with-revision-date {
  type string {
    pattern '[a-zA-Z][a-zA-Z0-9\-\_\.]*'
      + '(@\d{4}-(1[0-2]|0[1-9])-(0[1-9]|[1|2][0-9]|3[0-1]))?';
    pattern '\.\.\.[^xX].*\.[^mM].*\.\.[^lL].*';
  }
  description
    "Тип, задающий имя модуля и необязательную дату выпуска,
    например, ietf-yang-library@2019-01-04.";
}

sx:structure instance-data-set {
  description
    "Структура, задающая формат данных экземпляра YANG. Большинство
    узлов YANG предоставляет метаданные об экземпляре данных, а
    сам экземпляр содержится лишь в узле content-data.";
  leaf name {
    type string;
    description

```

```

"Произвольное имя набора данных экземпляра YANG, применяемое
в основном для описания. Однако при сохранении набора данных
экземпляра в файле имя файла ДОЛЖНО кодировать значение name
в соответствии с разделом 2 в RFC 9195.";
}
leaf format-version {
  type string {
    pattern '\d{4}-(1[0-2]|0[1-9])-(0[1-9]|[1|2][0-9]|3[0-1])';
  }
  default "2022-01-20";
  description
    "Для кодирования этого instance-data-set применяется revision
    из модуля ietf-yang-instance-data.";
}
leaf includes-defaults {
  type ncvd:with-defaults-mode;
  default "report-all";
  description
    "Указывает, как узлы с заданными по умолчанию значениями
    представляются для узлов данных из instance-data-set.

    Используются некоторые определения из раздела 3 в RFC 6243,
    но значение применяется в контексте файла данных экземпляра,
    а не запроса NETCONF с параметром <with-defaults>.

    Для файлов JSON кодирование опции report-all-tagged задано в
    параграфе 4.8.9 RFC 8040.";
  reference
    "RFC 6243: With-defaults Capability for NETCONF";
}
container content-schema {
  description
    "Схема содержимого (т. е. модули YANG), используемая для
    создания набора данных экземпляра. При отсутствии
    пользователь должен предоставить сведения из внешних
    документов.";
  choice content-schema-spec {
    description
      "Спецификация content-schema.";
    case simplified-inline {
      leaf-list module {
        type module-with-revision-date;
        min-elements 1;
        description
          "Список определяющих содержимое модулей YANG. Значению
          НУЖНО начинаться с имени модуля. Если модуль включает
          оператор revision, НУЖНО включать дату выпуска в
          запись списка, например, ietf-yang-library@2019-01-04.

          Использование этого leaf-list предполагает, что модули
          применяются без отклонений и поддерживаются все
          функции. Указывать несколько выпусков одного модуля
          НЕДОПУСТИМО.";
      }
    }
    case inline {
      anydata inline-yang-library {
        mandatory true;
        description
          "Данные экземпляра, соответствующие определению
          ietf-yang-library@2019-01-04 для набора задающих
          содержимое этого instance-data-set модулей YANG.";
      }
    }
    case uri {
      leaf same-schema-as-file {
        type inet:uri;
        description
          "Ссылка на другой файл данных экземпляра YANG с такой
          же схемой содержимого. ДОЛЖНЫ поддерживаться файлы,
          применяющие методы inline и simplified-inline. МОГУТ
          поддерживаться файлы, применяющие метод URI. ДОЛЖНЫ
          поддерживаться схемы URL file:// и https:// и МОГУТ
          поддерживаться иные схемы.";
      }
    }
  }
}
leaf-list description {
  type string;
  description
    "Описание набора данных экземпляра.";
}
leaf contact {
  type string;
  description

```

```

"Контактные данные человека или организации, куда следует
направлять запросы, связанные с набором данных экземпляра.";
}
leaf organization {
  type string;
  description
  "Организация, отвечающая за набор данных экземпляра.";
}
leaf datastore {
  type ds:datastore-ref;
  description
  "Идентификатор хранилища данных, которым связан набор данных
экземпляра, например, хранилище, откуда данные считаны или
куда данные были загружены, документированное хранилище
данных. Если невозможно указать одно конкретное хранилище,
лист ДОЛЖЕН отсутствовать. Если этого листа нет, хранилище
остаётся незадаанным.";
}
list revision {
  key "date";
  description
  "Наборам данных экземпляров, созданным при спецификации или
разработке, СЛЕДУЕТ иметь хотя бы одну запись revision. Для
каждого опубликованного изменения СЛЕДУЕТ указывать новый
оператор revision перед имеющимися (обратный порядок).

  Когда набор данных экземпляра, считан с сервера или создан
сервером или по иным причинам обновляется часто, НЕ СЛЕДУЕТ
применять revision.";
  leaf date {
    type string {
      pattern '\d{4}-(1[0-2]|0[1-9])-(0[1-9]|[12][0-9]|3[0-1])';
    }
    description
    "Дата последнего изменения набора данных экземпляра в
формате YYYY-MM-DD.";
  }
  leaf description {
    type string;
    description
    "Описание выпуска этого набора данных экземпляра.";
  }
}
leaf timestamp {
  type yang:date-and-time;
  description
  "Дата и время последнего изменения набора данных экземпляра.

  Когда набор данных экземпляра, считан с сервера или создан
сервером или по иным причинам обновляется часто, СЛЕДУЕТ
указывать timestamp.

  При наличии записей revision и timestamp в метке СЛЕДУЕТ
указывать ту же дату, как в последнем операторе revision.";
}
anydata content-data {
  description
  "Фактические данные экземпляра. Данные ДОЛЖНЫ соответствовать
модулям YANG, заданным в content-schema или иным способом.";
}
}
}
<CODE ENDS>

```

## 4. Вопросы безопасности

Заданный здесь модуль YANG определяет лишь структуру, задающую формат и заголовок метаданных для данных экземпляра YANG, указанных content-schema, поэтому шаблон вопросов безопасности для моделей YANG из параграфа 3.7.1 в [RFC8407] не применяется. Данные экземпляра предназначены для доступа через сохранённый файл или по протоколу или методу доступа к файлам.

Документ не задаёт методов, влияющих на поведение сервера.

Заголовок обычно не связан с безопасностью, однако может включать конфиденциальные сведения и в этом случае нужна защищённая обработка для:

- userinfo, если применяется метод URI для задания content-schema и URI включает сведения о пользователе;
- текст описания.

Содержимое может включать конфиденциальные данные, связь которых с безопасностью полностью определяет content-schema. В зависимости от природы данных экземпляра для ниже может требоваться защищённая обработка. Для сохранённого и передаваемого файла следует применять такую же обработку, как к результату операции чтения, возвращающей эти же данные. Механизмы защиты при передаче смягчат проблемы целостности.

Файлы данных экземпляров следует защищать от изменения и несанкционированного доступа с использованием обычных методов обработки файлов. При копировании исходных файлов или предоставлении доступа к ним

дополнительным пользователям следует принимать меры предосторожности, чтобы информация не была нечаянно раскрыта.

При использовании метода URI для задания content-schema возникает риск, что раздел конфигурации схемы в указанном файле данных экземпляра YANG может быть злонамеренно изменён даже в процессе нормальной обработки. В этом случае схема содержимого может отличаться от ожидаемой. Следует обеспечивать защиту целостности и стабильности файла, на который даётся ссылка.

## 5. Взаимодействие с IANA

Документ регистрирует URI и модуль YANG.

### 5.1. Регистрация URI

Этот документ регистрирует URI в реестре IETF XML Registry [RFC3688]

```
URI: urn:ietf:params:xml:ns:yang:ietf-yang-instance-data
Registrant Contact: The IESG.
XML: N/A, запрошенный URI является пространством имён XML.
```

### 5.2. Регистрация модуля YANG

Этот документ регистрирует модуль YANG в реестре YANG Module Names [RFC6020]

```
Name: ietf-yang-instance-data
Namespace: urn:ietf:params:xml:ns:yang:ietf-yang-instance-data
Prefix: yid
Reference: RFC 9195
```

## 6. Литература

### 6.1. Нормативные документы

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, [RFC 5234](#), DOI 10.17487/RFC5234, January 2008, <<https://www.rfc-editor.org/info/rfc5234>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](#), DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6243] Bierman, A. and B. Lengyel, "With-defaults Capability for NETCONF", [RFC 6243](#), DOI 10.17487/RFC6243, June 2011, <<https://www.rfc-editor.org/info/rfc6243>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", [RFC 6991](#), DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [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>>.
- [RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", [RFC 7951](#), DOI 10.17487/RFC7951, August 2016, <<https://www.rfc-editor.org/info/rfc7951>>.
- [RFC7952] Lhotka, L., "Defining and Using Metadata with YANG", [RFC 7952](#), DOI 10.17487/RFC7952, August 2016, <<https://www.rfc-editor.org/info/rfc7952>>.
- [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>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", [RFC 8342](#), DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8525] Bierman, A., Bjorklund, M., Schoenwaelder, J., Watsen, K., and R. Wilton, "YANG Library", [RFC 8525](#), DOI 10.17487/RFC8525, March 2019, <<https://www.rfc-editor.org/info/rfc8525>>.
- [RFC8526] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "NETCONF Extensions to Support the Network Management Datastore Architecture", [RFC 8526](#), DOI 10.17487/RFC8526, March 2019, <<https://www.rfc-editor.org/info/rfc8526>>.
- [RFC8527] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "RESTCONF Extensions to Support the Network Management Datastore Architecture", [RFC 8527](#), DOI 10.17487/RFC8527, March 2019, <<https://www.rfc-editor.org/info/rfc8527>>.
- [RFC8791] Bierman, A., Bjorklund, M., and K. Watsen, "YANG Data Structure Extensions", [RFC 8791](#), DOI 10.17487/RFC8791, June 2020, <<https://www.rfc-editor.org/info/rfc8791>>.

### 6.2. Дополнительная литература

- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, [RFC 3688](#), DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, [RFC 8340](#), DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8407] Bierman, A., "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", BCP 216, [RFC 8407](#), DOI 10.17487/RFC8407, October 2018, <<https://www.rfc-editor.org/info/rfc8407>>.

- [RFC8632] Vallin, S. and M. Bjorklund, "A YANG Data Model for Alarm Management", [RFC 8632](#), DOI 10.17487/RFC8632, September 2019, <<https://www.rfc-editor.org/info/rfc8632>>.
- [RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", [RFC 8641](#), DOI 10.17487/RFC8641, September 2019, <<https://www.rfc-editor.org/info/rfc8641>>.
- [RFC8792] Watson, K., Auerswald, E., Farrel, A., and Q. Wu, "Handling Long Lines in Content of Internet-Drafts and RFCs", [RFC 8792](#), DOI 10.17487/RFC8792, June 2020, <<https://www.rfc-editor.org/info/rfc8792>>.
- [RFC8808] Wu, Q., Lengyel, B., and Y. Niu, "A YANG Data Model for Factory Default Settings", [RFC 8808](#), DOI 10.17487/RFC8808, August 2020, <<https://www.rfc-editor.org/info/rfc8808>>.

## Приложение А. Совместимость с прежними версиями

Концепция совместимости с прежними версиями (backwards compatibility) и совместимые с ними изменения не определены для наборов данных экземпляра, поскольку они сильно зависят от конкретного варианта применения и схемы содержимого (content-schema).

Для наборов данных экземпляров, созданных при разработке или спецификации некоторых изменений, указанных ниже, следует избегать. YANG использует концепцию управляемых элементов (сущностей), идентифицируемых ключами. Если связь между сущностью и ключом не сохраняется при обновлении, могут возникать указанные ниже проблемы.

- Если меняется значение ключа записи списка того же элемента, что был раньше, пользователь может ошибочно идентифицировать запись как новую.
- Если значение записи меняется, а ключ остаётся (например, при переопределении alarm-type без смены alarm-type-id), изменение может остаться незамеченным.
- Если значение ключа удалённой записи применяется снова для иной записи, изменение может быть ошибочно истолковано как восстановление прежнего элемента.

## Приложение В. Варианты применения

Это приложение не является нормативным.

### В.1. Пример 1. Раннее документирование возможностей сервера

Сервер имеет много возможностей, определённых в модулях YANG, которые можно узнать от сервера по протоколу NETCONF или RESTCONF. Возможности сервера включают:

- данные, определённые в ietf-yang-library: модули и submodule YANG, функции (feature), отклонения (deviation), точки монтирования схемы (schema-mount), хранилища данных ([RFC8525]);
- поддерживаемые сигналы тревоги ([RFC8632]);
- узлы данных и ветви с поддержкой и без поддержки уведомлений при изменении ([RFC8641]);
- возможности NETCONF из ietf-netconf-monitoring.

Считается верным разрешать клиенту запрашивать эти данные у работающего сервера, но это возможно не всегда.

Зачастую при выпуске узла сети выпускается и связанная с ним система управления сетью (Network Management System или NMS). NMS зависит от возможностей сервера и в процессе реализации NMS нужны сведения об этих возможностях. Если такие сведения недоступны заранее в каком-либо документе и их можно получить только от работающего узла сети, реализация NMS задерживается, поскольку приходится ожидать готовности узла. Кроме того, допущение о том, что у всех разработчиков NMS имеется корректно настроенный узел сети, от которого можно получить данные, является слишком дорогостоящим (NMS может работать с десятками типов узлов).

Операторы часто создают свои системы NMS, в которые нужно интегрировать узлы производителей оборудования. Оператору нужно знать серверные возможности узлов сети для создания системы управления. Более того, на решение оператора о приобретении оборудования может влиять набор функций эксплуатации, администрирования и поддержки (Operations, Administration, and Maintenance или OAM), документированных как возможности сервера.

Системным интеграторам и многим другим также заранее нужны сведения о возможностях. Примерами может служить тестирование моделей, разработка документации и т. п.

Большинство возможностей сервера сравнительно стабильны и меняются лишь при обновлении или в связи с лицензированием или изменением состава оборудования. Обычно возможности задаёт производитель на этапе разработки до выпуска продукции. Целесообразно и желательно задать и документировать их заранее, например в файле данных экземпляра YANG.

Предполагается, что будет создан специальный документ IETF, подробно указывающий, какие возможности серверов следует документировать.

### В.2. Пример 2. Предварительная загрузка данных

Некоторые части конфигурации должны быть полностью настраиваемыми оператором, однако зачастую достаточно простой конфигурации, принятой по умолчанию. Одним из примеров служат группы управления (ролей) для доступа и связанные с ними правила. Хотя изошёренные операторы могут задавать десятки различных групп, зачастую достаточно базовой тройки - только чтение для оператора, чтение и запись для системного администратора и роль администратора безопасности. Производители часто предоставляют такую базовую конфигурацию по умолчанию, чтобы упростить работу оператора.

Производитель устройства может задать набор принятых по умолчанию групп (/nacm:nacm/groups) и правил для доступа этих групп к конкретным частям базовых моделей (/nacm:nacm/rule-list/rule).

Файлы данных экземпляров YANG можно использовать для документирования и/или предварительной загрузки принятой по умолчанию конфигурации.

### В.3. Пример 3. Документирование заводских установок

Почти каждый сервер имеет заданную при производстве конфигурацию. Если система настроена с ошибками или текущая конфигурация была повреждена, систему можно сбросить к заданным при производстве настройкам.

Файлы данных экземпляров YANG можно использовать для документирования заводских настроек (см. [RFC8808]).

### Благодарности

Спасибо Andy Bierman, Juergen Schoenwaelder, Rob Wilton, Joe Clarke, Kent Watsen, Martin Bjorklund, Ladislav Lhotka, Qin Wu и другим участникам рабочей группы Netmod за полезные комментарии, дискуссии и предложения.

### Адреса авторов

**Balazs Lengyel**

Ericsson

Budapest

Magyar Tudosok korutja 11

1117

Hungary

Email: [balazs.lengyel@ericsson.com](mailto:balazs.lengyel@ericsson.com)**Benoit Claise**

Huawei

Email: [benoit.claise@huawei.com](mailto:benoit.claise@huawei.com)

### Перевод на русский язык

Николай Малых

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