

Internet Engineering Task Force (IETF)
Request for Comments: 8526
Updates: 6241, 7950
Category: Standards Track
ISSN: 2070-1721

M. Bjorklund
Tail-f Systems
J. Schoenwaelder
Jacobs University
P. Shafer
Juniper Networks
K. Watsen
Watsen Networks
R. Wilton
Cisco Systems
March 2019

NETCONF Extensions to Support the Network Management Datastore Architecture

Расширения NETCONF для поддержки архитектуры хранилищ управления сетью

Аннотация

Этот документ расширяет протокол настройки сети (Network Configuration Protocol или NETCONF), определённый в RFC 6241, для поддержки архитектуры хранилищ данных управления сетью (Network Management Datastore Architecture или NMDA), определённой в RFC 8342.

Документ обновляет RFC 6241 и RFC 7950. Обновление RFC 6241 добавляет операции `<get-data>` и `<edit-data>`, а также обновляет имеющиеся операции `<lock>`, `<unlock>` и `<validate>`. Обновление RFC 7950 требует использования библиотеки YANG (описана в RFC 8525) серверами NETCONF, реализующими NMDA.

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. Дополнительную информацию о стандартах Internet можно найти в разделе 2 в RFC 7841.

Информацию о текущем статусе документа, ошибках и способах обратной связи можно найти по ссылке <https://www.rfc-editor.org/info/rfc8526>.

Авторские права

Авторские права (Copyright (c) 2019) 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
1.1. Терминология.....	2
1.2. Диаграммы деревьев.....	2
2. Хранилище и требования к библиотеке YANG.....	2
3. Расширения NETCONF.....	2
3.1. Новые операции NETCONF.....	2
3.1.1. Операция <code><get-data></code>	2
3.1.1.1. Аннотация метаданных <code>origin</code>	3
3.1.1.2. Взаимодействия <code>with-defaults</code>	3
3.1.1.3. Пример нахождения полного субдерева из хранилища <code><running></code>	3
3.1.1.4. Пример нахождения фильтрованного субдерева из хранилища <code><operational></code>	4
3.1.2. Операция <code><edit-data></code>	4
3.1.2.1. Пример установки листа на интерфейсе в хранилище <code><running></code>	5
3.2. Дополнения операций NETCONF.....	5
4. Модуль YANG хранилищ NETCONF.....	5
5. Взаимодействие с IANA.....	10
6. Вопросы безопасности.....	10
7. Литература.....	10
7.1. Нормативные документы.....	10
7.2. Дополнительная литература.....	10
Адреса авторов.....	10

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

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

1. Введение

Этот документ расширяет протокол NETCONF [RFC6241] для поддержки архитектуры хранилищ данных (Network Management Datastore Architecture или NMDA), определённой в [RFC8342].

Документ обновляет [RFC6241] для поддержки взаимодействия клиентов NETCONF со всеми хранилищами данных, поддерживаемыми серверами, реализующими NMDA. Обновление добавляет операции `<get-data>` и `<edit-data>` и дополняет имеющиеся операции `<lock>`, `<unlock>` и `<validate>`.

Документ также обновляет [RFC7950] для поддержки клиентами NETCONF обнаружения хранилищ данных, поддерживаемых серверами NETCONF, и определения модулей, которые поддерживаются в каждом хранилище. Это требует от серверов NETCONF, реализующих NMDA, поддержки библиотеки YANG [RFC8525].

1.1. Терминология

В этом документе используется терминология, определённая в NMDA [RFC8342]. Термин «идентификатор содержимого библиотеки YANG» (YANG library content identifier) определён в [RFC8525].

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.

1.2. Диаграммы деревьев

Диаграммы деревьев в этом документе используют обозначения из [RFC8340].

2. Хранилище и требования к библиотеке YANG

Поддерживающий NMDA сервер NETCONF **должен** реализовать модуль `ietf-netconf-nmda`, определённый в этом документе, **должен** поддерживать хранилище рабочего состояния (operational) и **должен** реализовать последний выпуск (2019-01-04) модуля `ietf-yang-library`, определённого в [RFC8525].

Клиент NETCONF может раскрыть поддерживаемые сервером хранилища и модули YANG путём чтения данных библиотеки YANG из хранилища данных рабочего состояния.

Сервер **должен** анонсировать в своём сообщении `<hello>` указанные ниже возможности (перевод строки и пробелы используются лишь для форматирования).

```
urn:ietf:params:netconf:capability:yang-library:1.1?
  revision=<date>&content-id=<content-id-value>
```

Параметр `revision` имеет такое же значение как дата выпуска модуля `ietf-yang-library`, реализуемого сервером. Этот параметр **должен** присутствовать.

Параметр `content-id` содержит идентификатор содержимого библиотеки YANG [RFC8525]. Этот параметр должен присутствовать.

С помощью этого механизма клиент может кэшировать поддерживаемые сервером хранилища и модули YANG и обновлять кэш лишь при смене значения `content-id` в сообщении `<hello>`.

Этот документ обновляет параграф 5.6.4 [RFC7950], чтобы позволить серверам анонсировать возможность `:yang-library:1.1` вместо `:yang-library:1.0` и реализовать subtree `/yang-library` [RFC8525] взамен `/modules-state`.

3. Расширения NETCONF

В этом разделе описаны расширения NETCONF, требуемые для поддержки NMDA. Эти расширения определены в новом модуле YANG `ietf-netconf-nmda` [RFC7950].

Изменения включают использование исходных и целевых параметров на основе отождествления datastore, определённого в модуле `ietf-datastores` [RFC8342]. Применения отождествления позволят вносить расширения, которые не разрешала исходная стратегия на основе выбора (например, `<get-config>` и `<edit-config>`).

3.1. Новые операции NETCONF

В этом документе определены операции `<get-data>` и `<edit-data>` для поддержки NMDA. Операции похожи на `<get-config>` и `<edit-config>`, но могут работать с расширяемыми наборами хранилищ.

3.1.1. Операция `<get-data>`

Операция `<get-data>` извлекает данные из указанного хранилища NMDA. Операция похожа на NETCONF `<get-config>`, определённую в [RFC6241], но позволяет более гибко указывать исходное хранилище данных.

```
+---x get-data
  +---w input
  | +---w datastore                               ds:datastore-ref
  | +---w (filter-spec)?
  | | +--- (subtree-filter)
  | | | +---w subtree-filter?                   <anydata>
  | | +--- (xpath-filter)
  | | +---w xpath-filter?                       yang:xpath1.0 {nc:xpath}?
  | +---w config-filter?                         boolean
  | +---w (origin-filters)? {origin}?
  | | +--- (origin-filter)
  | | | +---w origin-filter*                    or:origin-ref
  | | +--- (negated-origin-filter)
```

		+---w negated-origin-filter*	or:origin-ref
		+---w max-depth?	union
		+---w with-origin?	empty {origin}?
		+---w with-defaults?	with-defaults-mode
+--ro output			
+---ro data? <anydata>			

Параметр `datastore` указывает хранилище для извлечения искомых данных. Это отождествление `datastore`.

Операция `<get-data>` принимает содержимое параметра фильтрации, подобно параметру `filter` в `<get-config>`, но использует явные узлы для фильтрации поддерева (`subtree-filter`) и фильтрацию XPath (`xpath-filter`).

Параметр `config-filter` может служить для извлечения лишь узлов `config true` или `config false`.

Параметр `origin-filter`, которые может указываться неоднократно, выбирает совпадающие узлы или узлы, выведенные из любого данного значения. Параметр `negated-origin-filter`, которые может указываться несколько раз, выбирает совпадающие узлы или узлы, выведенные из любого данного значения. Параметры `origin-filter` и `negated-origin-filter` не могут применяться совместно.

Параметр `max-depth` может применяться клиентом для ограничения числа уровней поддерева, возвращаемых в отклике.

3.1.1.1. Аннотация метаданных `origin`

Операция `<get-data>` определяет параметр `with-origin`, наличие которого запрашивает у сервера включение аннотации метаданных `metadata` в свой отклик, как задано в NMDA. Этот параметр действителен лишь для хранилища данных рабочего состояния и других хранилищ с отождествлениями, выведенными из `operational`. В противном случае возвращается ошибка, если указано не подходящее хранилище данных, как определено в модуле `ietf-netconf-nmda` (раздел 4). Отметим, что аннотации метаданных `origin` не включаются в отклик, пока клиент явно не запрашивает их.

Данные в хранилище рабочего состояния могут поступать из разных источников. Серверу следует возвращать значение аннотации метаданных `origin`, наиболее точно указывающее источник рабочего значения, как указано в параграфе 5.3.4 [RFC8342].

При кодировании аннотации метаданных `origin` для иерархии возвращаемых узлов, аннотация может пропускаться для дочерних узлов, когда значение соответствует родительскому узлу, как описано в модуле YANG `ietf-origin` [RFC8342].

Поддержка параметра `with-origin` **необязательна**. Параметр идентифицируется свойством `origin`.

3.1.1.2. Взаимодействия `with-defaults`

Если сервер поддерживает свойство `with-defaults`, параметр `with-defaults`, определённый в [RFC6243], поддерживается для операций `<get-data>`, нацеленных на унаследованные хранилища конфигурации.

Поддержка параметра `with-defaults` **необязательна** для операций `<get-data>` с целью `<operational>`. Связанная возможность для указания поддержки сервером идентифицируется URI

```
urn:ietf:params:netconf:capability:with-operational-defaults:1.0
```

Если параметр `with-defaults` поддерживается для операций `<get-data>` над хранилищем `<operational>`, все режимы извлечения, заданные в параметре `basic-mode` или `also-supported` со свойством `with-defaults`, будут разрешены. Поведение параметра `with-defaults` для хранилища `<operational>` описано ниже.

- Если параметр `with-defaults` не задан или имеет значение `explicit`, `report-all` или `report-all-tagged`, значения `in use`, как указано в параграфе 5.3 [RFC8342], возвращаются из хранилища рабочего состояния, даже если узел имеет оператор `default` в модуле YANG и это принятое по умолчанию значение используется сервером. Если для параметра `with-defaults` установлено значение `report-all-tagged`, любые значения, соответствующие принятым в схеме по умолчанию, помещаются дополнительными метаданными, как указано в параграфе 3.4 [RFC6243].
- Если для параметра `with-defaults` установлено значение `trim`, возвращаются все значения `in use`, за исключением фильтруемых на выходе для исключения всех значений, которые соответствуют принятым по умолчанию в схеме YANG.

Поддержку `with-defaults` в операциях `<get-data>` на любых хранилищах данных, не определённых в [RFC8342], следует определять в спецификациях хранилищ данных.

3.1.1.3. Пример нахождения полного поддерева из хранилища `<running>`

Ниже приведён пример, показывающий вариант `<get-data>` примера `<get-config>` из параграфа 7.1 в [RFC6241] с выбором всего поддерева `/users`.

```
<rpc message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-data xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-nmda"
    xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores">
    <datastore>ds:running</datastore>
    <subtree-filter>
      <top xmlns="http://example.com/schema/1.2/config">
        <users/>
      </top>
    </subtree-filter>
  </get-data>
</rpc>

<rpc-reply message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-nmda">
    <top xmlns="http://example.com/schema/1.2/config">
```

```

<users>
  <user>
    <name>root</name>
    <type>superuser</type>
    <full-name>Charlie Root</full-name>
    <company-info>
      <dept>1</dept>
      <id>1</id>
    </company-info>
  </user>
  <!-- Здесь указываются дополнительные элементы <user> ... -->
</users>
</top>
</data>
</rpc-reply>

```

3.1.1.4. Пример нахождения фильтрованного субдерева из хранилища <operational>

Ниже приведён пример, показывающий использование `origin-filter` для извлечения узлов из хранилища `<operational>`. В примере используются синтезированная модель данных из приложения С к [RFC8342].

```

<rpc message-id="102"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-data xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-nmda"
    xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores"
    xmlns:or="urn:ietf:params:xml:ns:yang:ietf-origin">
    <datastore>ds:operational</datastore>
    <subtree-filter>
      <bgp xmlns="http://example.com/ns/bgp" />
    </subtree-filter>
    <origin-filter>or:intended</origin-filter>
    <origin-filter>or:system</origin-filter>
    <with-origin/>
  </get-data>
</rpc>

<rpc-reply message-id="102"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-nmda">
    <bgp xmlns="http://example.com/ns/bgp"
      xmlns:or="urn:ietf:params:xml:ns:yang:ietf-origin"
      or:origin="or:intended">
      <peer>
        <name>2001:db8::2:3</name>
        <local-port or:origin="or:system">60794</local-port>
        <state>established</state>
      </peer>
    </bgp>
  </data>
</rpc-reply>

```

Чтобы не извлекать узлы данных состояния системы можно применить фильтр `config-filter`.

```

<rpc message-id="103"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-data xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-nmda"
    xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores"
    xmlns:or="urn:ietf:params:xml:ns:yang:ietf-origin">
    <datastore>ds:operational</datastore>
    <subtree-filter>
      <bgp xmlns="http://example.com/ns/bgp" />
    </subtree-filter>
    <config-filter>true</config-filter>
    <origin-filter>or:intended</origin-filter>
    <origin-filter>or:system</origin-filter>
    <with-origin/>
  </get-data>
</rpc>

<rpc-reply message-id="103"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-nmda">
    <bgp xmlns="http://example.com/ns/bgp"
      xmlns:or="urn:ietf:params:xml:ns:yang:ietf-origin"
      or:origin="or:intended">
      <peer>
        <name>2001:db8::2:3</name>
        <local-port or:origin="or:system">60794</local-port>
      </peer>
    </bgp>
  </data>
</rpc-reply>

```

3.1.2. Операция <edit-data>

Операция `<edit-data>` меняет содержимое доступного для записи хранилища, подобно операции `<edit-config>`, определённой в [RFC6241], но с большей гибкостью в именовании целевого хранилища данных. Если операция `<edit-data>` вызывается для хранилища без возможности записи, возвращается ошибка, как указано в модуле `ietf-netconf-nmda` (см. раздел 4).

```

+---x edit-data
  +---w input
    +---w datastore                ds:datastore-ref
    +---w default-operation?      enumeration
    +---w (edit-content)
      +--:(config)
        | +---w config?           <anydata>
        +--:(url)
          +---w url?              inet:uri {nc:url}?

```

Параметр `datastore` является отождествлением `datastore`, указывающим желаемое целевое хранилище, куда следует внести изменения.

Параметр `default-operation` задаёт используемую по умолчанию информацию. Он является копией параметра `default-operation` операции `<edit-config>`.

Параметр `edit-content` задаёт содержимое операции редактирования. Он отражает выбор `edit-content` операции `<edit-config>`. Отметим однако, что элемент `config` в выборе `edit-content` операции `<edit-data>` использует `anydata` (введён в YANG 1.1 [RFC7950]), тогда как элемент `config` в выборе `edit-content` операции `<edit-config>` использует `anyxml`.

Операция `<edit-data>` не поддерживает параметры `error-option` и `test-option`, которые были частью операции `<edit-config>`. Поведение `<edit-data>` при ошибках соответствует значению `rollback-on-error` в параметре `error-option`.

Если сервер поддерживает свойство `with-defaults`, семантика режимов редактирования такая же как для операции `<edit-config>`, описанной в параграфе 4.5.2 [RFC6243].

Семантику `with-defaults` в операциях `<edit-data>` на любом нетрадиционном хранилище данных конфигурации следует задавать в спецификации хранилища данных.

3.1.2.1. Пример установки листа на интерфейсе в хранилище `<running>`

Ниже приведён вариант `<edit-data>` для первого примера `<edit-config>` из параграфа 7.2 в [RFC6241]. Здесь установлено `MTU = 1500` на интерфейсе `Ethernet0/0` в хранилище рабочей конфигурации (`running`).

```

<rpc message-id="103"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-data xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-nmda"
    xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores">
    <datastore>ds:running</datastore>
    <config>
      <top xmlns="http://example.com/schema/1.2/config">
        <interface>
          <name>Ethernet0/0</name>
          <mtu>1500</mtu>
        </interface>
      </top>
    </config>
  </edit-data>
</rpc>

<rpc-reply message-id="103"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>

```

Другие примеры `<edit-config>` из параграфа 7.2 в [RFC6241] можно преобразовать в `<edit-data>` аналогичным путём.

3.2. Дополнения операций NETCONF

Некоторые из операций, определённых в базовом модуле NETCONF YANG `ietf-netconf` [RFC6241], можно использовать с новыми хранилищами данных. Операции `<lock>`, `<unlock>` и `<validate>` дополнены листом `datastore`, который позволяет выбрать желаемое хранилище. Если операция `<lock>`, `<unlock>`, `<validate>` не поддерживается конкретным хранилищем данных, возвращается ошибка, как указано в модуле `ietf-netconf-nmda` (раздел 4).

4. Модуль YANG хранилищ NETCONF

Этот модуль импортирует определения из [RFC6991], [RFC6241], [RFC6243], [RFC8342].

```

<CODE BEGINS> file "ietf-netconf-nmda@2019-01-07.yang"

module ietf-netconf-nmda {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-netconf-nmda";
  prefix ncds;

  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 6991: Common YANG Data Types";
  }
  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Data Types";
  }
  import ietf-datastores {
    prefix ds;
    reference

```

```
"RFC 8342: Network Management Datastore Architecture
(NMDA)";
}
import ietf-origin {
  prefix or;
  reference
    "RFC 8342: Network Management Datastore Architecture
    (NMDA)";
}
import ietf-netconf {
  prefix nc;
  reference
    "RFC 6241: Network Configuration Protocol (NETCONF)";
}
import ietf-netconf-with-defaults {
  prefix ncwd;
  reference
    "RFC 6243: With-defaults Capability for NETCONF";
}

organization
  "IETF NETCONF Working Group";

contact
  "WG Web: <https://datatracker.ietf.org/wg/netconf/>

  WG List: <mailto:netconf@ietf.org>

  Author: Martin Bjorklund
    <mailto:mbj@tail-f.com>

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

  Author: Phil Shafer
    <mailto:phil@juniper.net>

  Author: Kent Watsen
    <mailto:kent+ietf@watsen.net>

  Author: Robert Wilton
    <mailto:rwilton@cisco.com>";

description
  "Этот модуль YANG определяет набор операций NETCONF для поддержки
  архитектуры хранилищ данных управления сетью (NMDA).

  Ключевые слова MUST (ДОЛЖНО), MUST NOT (НЕДОПУСТИМО), REQUIRED
  (ТРЕБУЕТСЯ), SHALL (НУЖНО), SHALLNOT (НЕ НУЖНО), SHOULD (СЛЕДУЕТ,
  SHOULD NOT (НЕ СЛЕДУЕТ), RECOMMENDED (РЕКОМЕНДУЕТСЯ), NOT RECOMMENDED
  (НЕ РЕКОМЕНДУЕТСЯ), MAY (МОЖНО), OPTIONAL (НЕОБЯЗАТЕЛЬНО) в этом
  документе трактуются в соответствии с ВСП 14 (RFC 2119) (RFC 8174)
  тогда и только тогда, когда они выделены шрифтом, как показано здесь.

  Авторские права (Copyright (c) 2019) принадлежат IETF Trust и лицам,
  указанным как авторы. Все права защищены.

  Распространение и использование в исходной или двоичной форме в
  изменениями или без таковых разрешено в соответствии с условиями,
  указанными в Simplified BSD License раздела 4.c IETF Trust's Legal
  Provisions применительно к документам IETF
  (https://trustee.ietf.org/license-info).

  Эта версия модуля является частью RFC 8526, где правовые аспекты
  изложены более полно.";

revision 2019-01-07 {
  description
    "Исходный выпуск.";
  reference
    "RFC 8526: NETCONF Extensions to Support the Network Management
    Datastore Architecture";
}

feature origin {
  description
    "Указывает, что сервер поддерживает аннотации origin.";
  reference
    "RFC 8342: Network Management Datastore Architecture (NMDA)";
}

feature with-defaults {
  description
    "Свойство NETCONF :with-defaults. Если сервер анонсирует свойство
    :with-defaults для сессии, эта функция должна быть включена в
    сессии. В ином случае включение функции недопустимо.";
  reference
```

```

"RFC 6243: With-defaults Capability for NETCONF, Section 4; and
RFC 8526: NETCONF Extensions to Support the Network Management
Datastore Architecture, Section 3.1.1.2";
}

rpc get-data {
  description
    "Извлечение данных из хранилища NMDA. Содержимое, возвращаемое
    get-data, должно соответствовать всем фильтра (AND для фильтров).

    Все узлы-предки (включая ключи списков), выбранные фильтрами,
    включаются в отклик.

    Параметр with-origin действителен лишь для рабочего (operational)
    хранилища. Если with-origin используется с непригодным хранилищем,
    сервер ДОЛЖЕН возвращать элемент <rpc-error> с <error-tag> =
    invalid-value.

    Параметр with-defaults применяется к рабочему хранилищу, если
    анонсируется NETCONF со свойствами :with-defaults и
    :with-operational-defaults. Если параметр with-defaults
    присутствует в запросе, для которого он не поддерживается, сервер
    ДОЛЖЕН возвращать элемент <rpc-error> с <error-tag> =
    invalid-value.";
  input {
    leaf datastore {
      type ds:datastore-ref;
      mandatory true;
      description
        "Хранилище, из которого извлекаются данные.

        Если сервер не поддерживает хранилище, он ДОЛЖЕН возвращать
        элемент <rpc-error> с <error-tag> = invalid-value.";
    }
    choice filter-spec {
      description
        "Спецификация фильтра содержимого для запроса.";
      anydata subtree-filter {
        description
          "Указывает части целевого хранилища для извлечения.";
        reference
          "RFC 6241: Network Configuration Protocol (NETCONF),
          Section 6";
      }
      leaf xpath-filter {
        if-feature "nc:xpath";
        type yang:xpath1.0;
        description
          "Выражение XPath, указывающее часть хранилища для извлечения.

          Если выражение возвращает node-set, все узлы набора
          выбираются фильтром. В противном случае, операция
          <get-data> завершается отказом.

          Выражение оценивается в приведённом ниже контексте XPath.

          o Набор объявлений пространств имён в области действия
            листа xpath-filter.

          o Пустой набор привязок переменных.

          o Библиотекой функций служит библиотека ядра и функции
            XPath, заданные в разделе 10 RFC 7950.

          o Узлом контекста является корневой узел целевого хранилища.";
      }
    }
  }
  leaf config-filter {
    type boolean;
    description
      "Фильтр для узлов с данным значением свойства config. При
      значении true выбираются лишь узлы config true, при значении
      false – только config false. При отсутствии этого листа
      узлы не фильтруются.";
  }
  choice origin-filters {
    when 'derived-from-or-self(datastore, "ds:operational)';
    if-feature "origin";
    description
      "Фильтрация узлов конфигурации по аннотации origin. Узлы
      конфигурации без аннотации origin, считаются имеющими
      аннотацию origin со значением or:unknown.

      Узлы состояния системы не обрабатываются origin-filters.
      Эти узлы могут фильтроваться листом config-filter.";
  }
}

```

```
leaf-list origin-filter {
  type or:origin-ref;
  description
    "Фильтр по аннотации origin. Узел конфигурации соответствует
    фильтру, если его аннотация origin совпадает или выводится
    из заданных значений фильтра.";
}
leaf-list negated-origin-filter {
  type or:origin-ref;
  description
    "Фильтр по аннотации origin. Узел конфигурации соответствует
    фильтру, если его аннотация origin не совпадает и не
    выводится из заданных значений фильтра.";
}
}
leaf max-depth {
  type union {
    type uint16 {
      range "1..65535";
    }
    type enumeration {
      enum unbounded {
        description
          "Включаются все узлы-наследники.";
      }
    }
  }
  default "unbounded";
  description
    "Для каждого узла, выбранного фильтрами, этот параметр указывает,
    сколько уровней концептуального субдерева следует возвращать в
    отклике. Если задана глубина 1, отклик включает лишь выбранные
    узлы без потомков. При неограниченной (unbounded) глубине
    включаются все узлы-наследники.";
}
leaf with-origin {
  when 'derived-from-or-self(../datastore, "ds:operational)';
  if-feature "origin";
  type empty;
  description
    "При наличии этого параметра сервер будет возвращать аннотацию
    origin для имеющих её узлов.";
}
uses ncwd:with-defaults-parameters {
  if-feature "with-defaults";
}
}
output {
  anydata data {
    description
      "Копирование подмножества исходного хранилища, соответствующего
      критериям фильтра (при наличии). Пустой контейнер данных
      указывает, что запрос не возвращает каких-либо результатов.";
  }
}
}
rpc edit-data {
  description
    "Редактирование данных в хранилище NMDA.

    При возникновении ошибки, такой как генерация элемента
    <rpc-error>, сервер будет прерывать обработку операции
    <edit-data> и восстанавливать для указанной конфигурации
    её полное состояние на момент запуска <edit-data>.";
  input {
    leaf datastore {
      type ds:datastore-ref;
      mandatory true;
      description
        "Хранилище для выполнения операции <edit-data>.

        Если запись в целевое хранилище невозможна или не
        поддерживается сервером, сервер ДОЛЖЕН возвращать
        элемент <rpc-error> с <error-tag> = invalid-value.";
    }
  }
  leaf default-operation {
    type enumeration {
      enum merge {
        description
          "По умолчанию выполняется операция слияния.";
      }
      enum replace {
        description
          "По умолчанию выполняется операция замены.";
      }
    }
  }
}
```



```

    enum none {
        description
            "Операция по умолчанию не задана.";
    }
}
default "merge";
description
    "Используемая по умолчанию операция.";
}
choice edit-content {
    mandatory true;
    description
        "Содержимое для операции редактирования.";
    anydata config {
        description
            "Встроенное содержимое конфигурации.";
    }
    leaf url {
        if-feature "nc:url";
        type inet:uri;
        description
            "URL содержимого конфигурации.";
    }
}
}
}
}
/*
 * Дополнение параметра datastore к операциям <lock> и <unlock>.
 */

augment "/nc:lock/nc:input/nc:target/nc:config-target" {
    description
        "Добавление хранилища NMDA в качестве цели.";
    leaf datastore {
        type ds:datastore-ref;
        description
            "Хранилище данных для операции lock.

            Операция <lock> поддерживается лишь для хранилищ с
            возможностью записи.

            Если сервер не поддерживает операцию <lock> на заданном
            целевом хранилище, он ДОЛЖЕН возвращать элемент
            <rpc-error> с <error-tag> = invalid-value.";
    }
}

augment "/nc:unlock/nc:input/nc:target/nc:config-target" {
    description
        "Добавление хранилища NMDA в качестве цели.";
    leaf datastore {
        type ds:datastore-ref;
        description
            "Хранилище данных для операции unlock.

            Операция <unlock> поддерживается лишь для хранилищ с
            возможностью записи.

            Если сервер не поддерживает операцию <unlock> на заданном
            целевом хранилище, он ДОЛЖЕН возвращать элемент
            <rpc-error> с <error-tag> = invalid-value.";
    }
}

/* Дополнение параметра datastore к операции <validate>. */

augment "/nc:validate/nc:input/nc:source/nc:config-source" {
    description
        "Добавление хранилища NMDA в качестве источника.";
    leaf datastore {
        type ds:datastore-ref;
        description
            "Хранилище данных для проверки.

            Операция <validate> поддерживается лишь на хранилищах
            конфигурации.

            Если сервер не поддерживает операцию <validate> на
            указанном целевом хранилище, он ДОЛЖЕН возвращать
            элемент <rpc-error> с <error-tag> = invalid-value.";
    }
}
}
}
}

```

<CODE ENDS>

5. Взаимодействие с IANA

Этот документ регистрирует два URN идентификаторов возможностей в реестре Network Configuration Protocol (NETCONF) Capability URNs.

Индекс

:yang-library:1.1

:with-operational-defaults

Документ регистрирует URI в реестре IETF XML Registry [RFC3688] в соответствии с форматом RFC 3688.

URI: urn:ietf:params:xml:ns:yang:ietf-netconf-nmda

Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace.

Документ регистрирует модуль YANG в реестре YANG Module Names [RFC6020].

name: ietf-netconf-nmda

namespace: urn:ietf:params:xml:ns:yang:ietf-netconf-nmda

prefix: ncds

reference: RFC 8526

Идентификатор возможности

urn:ietf:params:netconf:capability:yang-library:1.1

urn:ietf:params:netconf:capability:with-operational-defaults:1.0

6. Вопросы безопасности

Определённый здесь модуль YANG расширяет базовые операции протокола NETCONF [RFC6241]. Нижним уровнем NETCONF является защищённый транспортный уровень с обязательной реализацией защищённого транспорта Secure Shell (SSH) [RFC6242].

Модель управления доступом к конфигурации сети (Network Configuration Access Control Model или NACM) [RFC8341] обеспечивает способы ограничения доступа конкретным пользователям NETCONF к предопределённому набору доступных операций и содержимого протокола NETCONF.

Вопросы безопасности базовых операций протокола NETCONF (раздел 9 в [RFC6241]) применимы и к новым операциям NETCONF <get-data> и <edit-data>, определенным в этом документе.

7. Литература

7.1. 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>>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, [RFC 3688](#), DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.

[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>>.

[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>>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", [RFC 6242](#), DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.

[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>>.

[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>>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, [RFC 8341](#), DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.

[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>>.

7.2. Дополнительная литература

[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>>.

Адреса авторов

Martin Bjorklund

Tail-f Systems

Email: mbj@tail-f.com

Juergen Schoenwaelder

Jacobs University

Email: j.schoenwaelder@jacobs-university.de

Phil Shafer

Juniper Networks

Email: phil@juniper.net**Kent Watsen**

Watsen Networks

Email: kent+ietf@watsen.net**Robert Wilton**

Cisco Systems

Email: rwilton@cisco.com**Перевод на русский язык**

Николай Малых

nmalykh@protokols.ru