CCAMP Working Group
Internet-Draft
Intended status: Standards Track
Expires: 20 April 2026

S. Homma, Ed.
NTT
H. Irino, Ed.
NTT West
T. Mano
K. Anazawa
NTT
Y. Tochio
1Finity

17 October 2025

A YANG Data Model for CMIS Access and Control
draft-hi-ccamp-cmis-control-yang-01

Abstract

This document provides a YANG data model to access to and control CMIS for managing pluggable Digital Coherent Optics transceivers equipped in a router or a switch from outside. CMIS has custom pages which enables to be defined by the module vendor for its own usage, and allows to extend the uses of the optics devices.

Status of This Memo

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

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

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

This Internet-Draft will expire on 20 April 2026.

Copyright Notice

Table of Contents

1. Introduction

Pluggable Digital Coherent Optics (DCO) transceivers enable routers or switches to directly connect to optical network (e.g., DWDM or OTN). Pluggable DCO transceivers, such as CFP2-DCO and QSFP-DD DCO, implement optical connectors (i.e., Tx and Rx) and a Digital Signal Processor (DSP), and provide higher data rates (100 Gbps, 400 Gbps, and beyond) and flexible data transport.

Pluggable DCO transceivers, equiped by a platform device (e.g., a router or a switch), is generally controlled by network OS (NOS) running on the device with Content Management Interoperability Specificatins (CMIS) which is an open standard protocol designed to facilitate interoperability between management systems. The specification is defined in [OIF-CMIS].

Vender-specific differences in the interpretation of certain CMIS definitions and in the capabilities of pluggable DCO transceivers can lead to inconsistent behavior. Therefore, each NOS needs explicitly support each transceiver type. Additionally, existing data models such as OpenConfig and IETF YANG may not fully present or utilize all transceiver capabilities (Ref. [I-D.rokui-ccamp-actn-wdm-pluggable-modelling]).

CMIS also allows vendor-specific extensions of its transceiver features by using custom pages. For example, CMIS custom pages can be used for non-standardized functions. However, there is currently no standardized schema for external systems to access or manage CMIS custom pages.

This document defines a YANG data model that enables external systems to directly manage DSP registers in pluggable DCO transceivers installed in routers and switches, in compliance with the CMIS specification.

1.1. Terminology and Notations

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" this document are to be interpreted as described in BCP14, [RFC2119], [RFC8340] when, and only when, they appear in all capitals, as shown here.

The terms and their definitions used in this specification are described below:

* CMIS (Common Management Interface Specifications): A generic management communication interface together with a generic management interaction protocol between host and managed modules. The specification is defined in [OIF-CMIS];

* NACM (Network Configuration Access Control Model): A standard access control model to restrict NETCONF or RESTCONF protocol access for particular users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content. The specification is defined in [RFC8341].

* NETCONF (Network Configuration Protocol): Mechanisms to install, manipulate, and delete the configuration of network devices. The definitions and specification is described in [RFC6241];

* RESTCONF: An HTTP-based protocol that provides a programmatic interface for accessing data defined in YANG, using the datastore concepts defined in NETCONF. The specification is defined in [RFC8040].

The following terms of NETCONF defined in [RFC6241] are also used in this specification:

* (NETCONF) client
* configuration data
* datastore
* message
* remote procedure call (RPC)
* (NETCONF) server
* state data
* (NETCONF) user

This document makes use of the terms defined in [RFC7950].

1.2. Acronyms

The following acronyms are used in this document:

CE	Customer Edge
CDB	Command Data Block
DCO	Digital Coherent Optics
DSP	Digital Signal Processor
DWDM	Dense Wavelength Division Multiplexing
GSNR	Generalized Signal-to-Noise Ratio
i2c	Inter-Integrated Circuit
NOS	Network Operating System
NMS	Network Management System

OTN Optical Transport Network

QoT Quality of Transmission

TPA Third Party Application

WDM Wavelength Division Multiplexing

1.3. Tree Diagram

The tree diagrams used in this document follow the notation defined in [RFC8340].

1.4. Prefixes in Data Node Names

In this document, names of data nodes and other data model objects are prefixed using the standard prefix associated with the corresponding YANG imported modules. The proposed modules are augments to the ietf-interface [RFC7223]. The details of the modules are described in Section 3.

Prefix	YANG module	Reference
if	ietf-interfaces	[RFC7223]
cmis-ctrl	ietf-cmis-control	RFC XXXX
cmis-ctrl-pm	ietf-cmis-control-primitive	RFC XXXX
cmis-ctrl-act	ietf-cmis-control-action	RFC XXXX
cmis-ctrl-rpc	ietf-cmis-control-rpc	RFC XXXX
ietf-cmis-mon	cmis-monitor	RFC XXXX

Table 1: Prefixes and corresponding YANG module

Note: The RFC Editor will replace XXXX with the number assigned to the RFC once this draft becomes an RFC.

2. Usecases and Implementation Patterns

This section describes usecases of this YANG data model for accessing to and control CMIS and implementation patterns for using such YANG.

2.1. Usecases

2.1.1. Centralized Control of Pluggable DCO Transceivers

This YANG data model disaggregates management features for pluggable DCO transceivers which a platform device equips with from NOS, and enables centralized control of such transceivers. For example, in case that a customer's router (i.e., CE) is connected to DWDM/OTN provided by a service provider, this enables pluggable DCO transceivers installed to the customer's router to be controlled by a controller of the service provider as shown in Figure 1. The service provider can configure DCO transceivers installed in customer devices directly depending on its own policy. Then, some access control with NACM defined in [RFC8341] should be applied for preventing illegal configuration of the other than target modules or leak of information of the customer's devices.

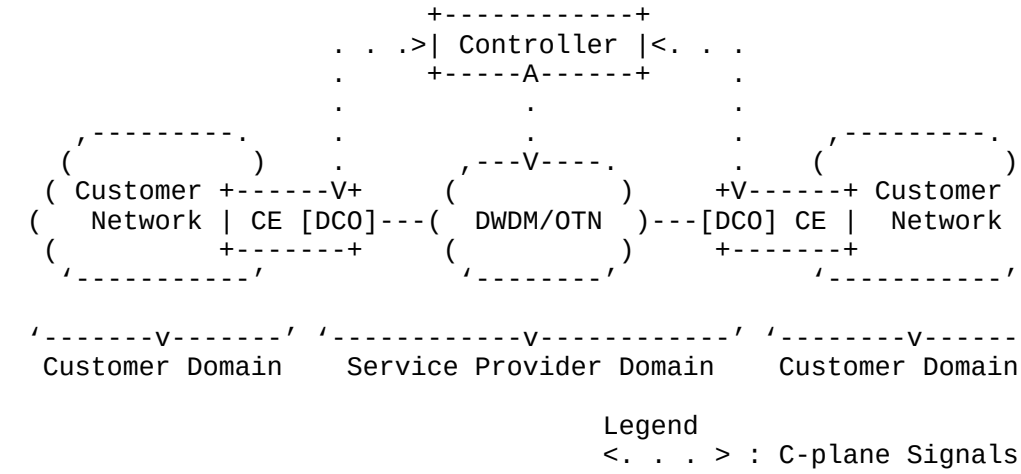

Figure 1: Centralized Control of Pluggable Modules

If CEs delegate whole the DCO transceivers management to the controller, the controller needs to monitor the DCO transceivers for detecting their failure occurred. For this case, notification-based YANG would be used Section 3.5.

On the other hand, in case that NOS of the platform devide manages pluggable modules, CMIS control from remote controller might cause inconsistency problem of pluggable modules. More details of this prroblem is considered in Section 4.

2.1.2. Obtaining Non-standardized Information with Custom Pages

By using this YANG data model, operators can obtain detailed DSP information contained in CMIS custom pages even if the modeling of the data is not standardized. Example uses of such detailed DSP information including fiber sensing (Ref. [ECOC48923.2020.9333176]), physical layer monitoring (Ref. [JLT.2021.3139167]), and accurate estimation (e.g., GSNR). (Ref. [JOCN.505729])

2.2. Implementation Patterns

This document introduce two patterns to implement a client using an interface in which this YANG data model is available:

Pattern1: Controller/NMS on Remote Host

In this pattern, a controller or an NMS implements a client using this YANG data model, and control pluggable modules installed to a platform device. The overview is shown in Figure 2.

Pattern2: Application Running on the Platform Device

In this pattern, a 3rd party's application running on a platform device implements a client using this YANG data model, and control pluggable modules installed to the device. That application can behave as a server using this YANG data model, or provide more generic interface, such as REST API to remote systems. The overview is shown in Figure 3.

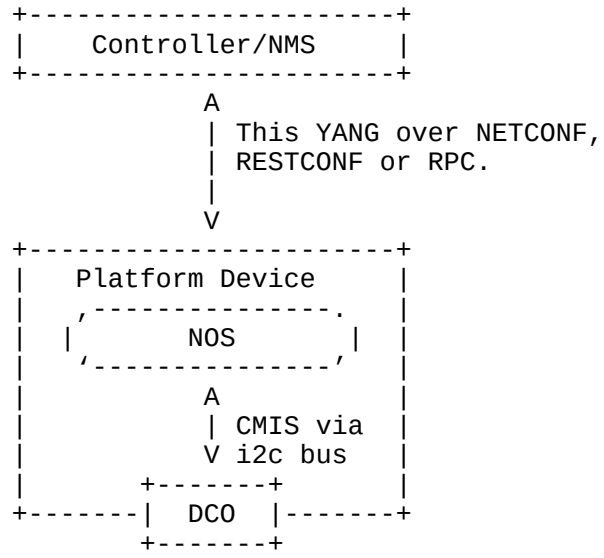

Figure 2: Implementation Pattern1 Overview

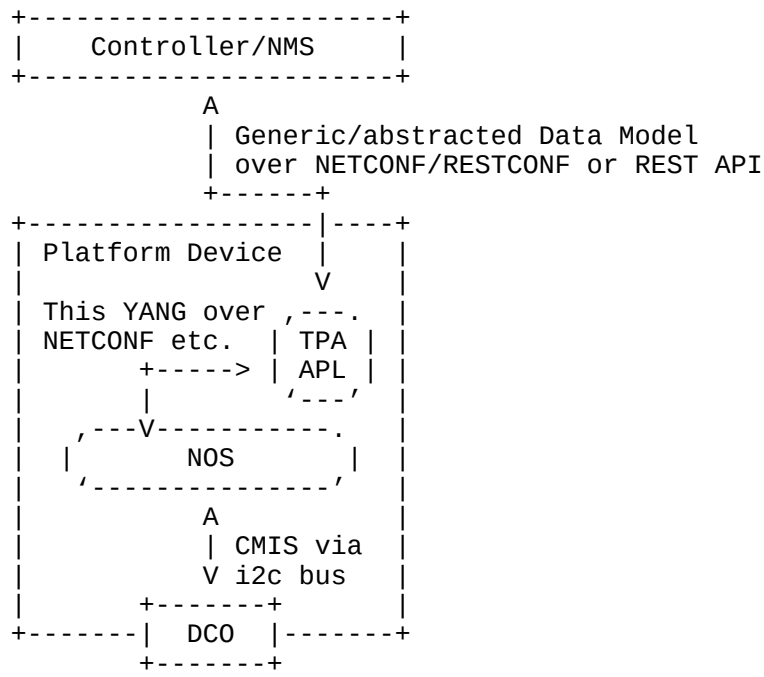

Figure 3: Implementation Pattern2 Overview

3. CMIS Access and Control Modules

This document defines the following YANG modules for the management of CMIS-capable pluggable DCO transceivers.

* ietf-cmis-control (base model, mandatory)

* ietf-cmis-control-primitive (optional)

* ietf-cmis-control-rpc (optional)

* ietf-cmis-control-action (optional)

* ietf-cmis-monitor (optional)

Each module is an augment to the ietf-interface. It allows the user to set the operating mode of CMIS for control pluggable devices as well as other operational parameters.

3.1. ietf-cmis-control

The structure of ietf-cmis-control is shown below:

```
module: ietf-cmis-control

  augment /if:interfaces/if:interface:
    +--rw cmis-control
       +--ro cmis-enabled?   boolean
       +--ro cmis-version?   string
       +--rw cmis-page* [page-num]
          +--rw page-num            uint8
          +--rw bank                uint8
          +--ro page-access-type?   access-type
          +--rw description?        string
          +--rw value* [offset]
             +--rw offset               uint8
             +--rw size                 uint8
             +--ro value-access-type?   access-type
             +--rw value-data           binary
             +--rw description?         string
```

Note that the values related to CMIS pages are defined in [OIF-CMIS].

The YANG module of "ietf-cmis-control" is defined as below.

```
<CODE BEGINS> file "ietf-cmis-control.yang"
module ietf-cmis-control {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-cmis-control";
  prefix cmis-ctrl;

  import ietf-interfaces {
    prefix if;
  }

  organization
    "IETF CCAMP Working Group";

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

    Editor:   Shunsuke Homma
      <mailto:shunsuke.homma.ietf@gmail.com>

    Editor:   Hitoshi Irino
```

```
        <mailto:hitoshi.irino.ge@west.ntt.co.jp>";

    description
      "This YANG module defines a data model for the management
      of CMIS (Common Management Interface Specification) pages
      as specified by OIF. It enables configuration and retrieval
      of CMIS page data, including access types and value fields,
      to support the management of pluggable optical modules via
      NETCONF or RESTCONF.

      Copyright (c) 2025 IETF Trust and the persons identified
      as authors of the code.  All rights reserved.

      Redistribution and use in source and binary forms, with or
      without modification, is permitted pursuant to, and
      subject to the license terms contained in, the Revised
      BSD License set forth in Section 4.c of the IETF Trust's
      Legal Provisions Relating to IETF Documents
      (http://trustee.ietf.org/license-info).

      This version of this YANG module is part of RFC XXXX
      (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
      for full legal notices.";

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

    revision "2025-04-21" {
      description
        "Initial revision.";
      reference
        "I-D.hi-ccamp-cmis-control-yang-00";
    }

    /*
     * CMIS control data nodes
     */

    typedef access-type {
        type enumeration {
          enum rw {
            description "A readable and writable element.";
          }
          enum rww {
            description "A readable and writable element that can be
            modified by the module.";
          }
          enum ro {
            description "A read-only element.";
```

```
          }
          enum wo {
            description "A write-only element.";
          }
          enum wo/sc {
            description "A write-only element with self-clearing side
            effect.";
          }
          enum ro/cor {
            description "A write-only element with celan-on-read side
            effect.";
          }
        }
        description "Defines access types for CMIS elements.";
      }

      grouping cmis-page {
        description
          "Parameters stored in the CMIS page";

        leaf page-num{
          type uint8 {
            range "0 .. 255";
          }
          mandatory true;
          description
            "The number of the CMIS page.";
        }

        leaf bank {
          type uint8;
          mandatory true;
          description
            "The banks corresponding to the CMIS page.";
        }

        leaf page-access-type {
          type access-type;
          config false;
          description "Access type of the CMIS page.";
        }

        leaf description {
          type string;
          description
            "The description of the CMIS page.";
        }
```

```
      list value {
        key "offset";
        description
        "The value contained in the CMIS page.";

        leaf offset {
          type uint8;
          mandatory true;
          description
            "The memory address of the value.";
        }

        leaf size {
          type uint8 {
            range "1 .. 128";
          }
          mandatory true;
          description
            "The memory size of the value.";
        }

        leaf value-access-type {
          type access-type;
          config false;
          description "Access type of the target value.";
        }

        leaf value-data {
          type binary;
          mandatory true;
          description
            "The data contained in the value. It is writable only
            when the access-type is not Read-Only or Read-Only with
            clean-on-read side effect.";
        }

        leaf description {
          type string;
          description
            "The description of the value.";
        }
      }
    }


    grouping cmis-pages {
      description
        "The list of the accessible CMIS pages supported by the
```

```
        pluggable device accommodated into the interface.";

      list cmis-page {
        key "page-num";
        uses cmis-page;
        }
    }

    grouping cmis-control {
      description
        "Parameters for primitive CMIS control of the pluggable device
        equipped in the interface.";

      leaf cmis-enabled {
        type boolean;
        default "false";
        config false;
        description
          "The availability of the CMIS for control the pluggable
          device equipped in the interface. If the device does not
          support CMIS, this value is false.";
      }

      leaf cmis-version {
        type string;
        config false;
        description
          "The version of the CMIS by the pluggable device.";
      }

      uses cmis-pages;

    }

    /*
     * Augment Interface
     */

    augment "/if:interfaces/if:interface" {
      container cmis-control {
        uses cmis-control;
      }
    }
  }
  <CODE ENDS>
```

3.2. ietf-cmis-control-primitive

This document provides a more primitive YANG data model for CMIS access and control. This is called as "ietf-cmis-control-primitive" or "primitive mode" and it doesn't manage supplemental information, such as access-types or description, of the fields in a CMIS page, and treat accessed memories as flat data structure.

This model enables implementation of server (i.e., network node) side to be simple, but on the other hand, client (i.e., controller) side is needed strict management of data of CMIS pages. For example, when a client sends a request to change any value, it needs to comprehend the page number, the offset, and the data size in which the data is contained.

The tree diagram of "ietf-cmis-control-primitive" is shown below:

```
module: ietf-cmis-control-primitive

  augment /if:interfaces/if:interface:
    +--rw cmis-control-primitive
       +--ro cmis-enabled?          boolean
       +--ro cmis-version?          string
       +--rw primitive-cmis-page* [page-num]
          +--rw page-num    uint8
          +--rw bank        uint8
          +--rw offset      uint8
          +--rw size        uint8
          +--rw value       binary
```

The "ietf-cmis-control-primitive" module is defined as below.

```
<CODE BEGINS> file "ietf-cmis-control-primitive.yang"
module ietf-cmis-control-primitive {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-cmis-control-primitive";
  prefix cmis-ctrl-pm;

  import ietf-interfaces {
    prefix if;
  }

  organization
    "IETF CCAMP Working Group";

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

```
       Editor:   Shunsuke Homma
         <mailto:shunsuke.homma.ietf@gmail.com>

       Editor:   Hitoshi Irino
         <mailto:hitoshi.irino.ge@west.ntt.co.jp>";

     description
       "This YANG module defines a data model for the management
       of CMIS (Common Management Interface Specification) pages
       as specified by OIF with RPC. It enables configuration and
       retrieval of CMIS page data, including access types and
       value fields, to support the management of pluggable optical
       modules via NETCONF or RESTCONF.

       Copyright (c) 2025 IETF Trust and the persons identified
       as authors of the code.  All rights reserved.

       Redistribution and use in source and binary forms, with or
       without modification, is permitted pursuant to, and
       subject to the license terms contained in, the Revised
       BSD License set forth in Section 4.c of the IETF Trust's
       Legal Provisions Relating to IETF Documents
       (http://trustee.ietf.org/license-info).

       This version of this YANG module is part of RFC XXXX
       (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
       for full legal notices.";

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

       revision "2025-4-21" {
         description
           "Initial revision.";
         reference
           "I-D.hi-ccamp-cmis-control-yang-00";
       }

       augment "/if:interfaces/if:interface" {
     action cmis-read {
       description "Read CMIS register under this interface.";
       input {
         leaf page {
           type uint8;
           mandatory true;
         }
         leaf bank {
           type uint8;
           mandatory true;
```

```
        }
        leaf offset {
          type uint8;
          mandatory true;
        }
        leaf size {
          type uint8;
          default 1;
        }
      }
      output {
        leaf data {
          type binary;
        }
      }
    }

    action cmis-write {
      description "Write CMIS register under this interface.";
      input {
        leaf page {
          type uint8;
          mandatory true;
        }
        leaf bank {
          type uint8;
          mandatory true;
        }
        leaf offset {
          type uint8;
          mandatory true;
        }
        leaf data {
          type binary;
          mandatory true;
        }
      }
    }
  }
  <CODE ENDS>
```

3.3. ietf-cmis-control-action

The "ietf-cmis-control-action" module defines actions-based controls of CMIS pages with NETCONF RPC.

```
module: ietf-cmis-control-action

  augment /if:interfaces/if:interface:
    +---x cmis-read
    |  +---w input
    |  |  +---w page      uint8
    |  |  +---w bank      uint8
    |  |  +---w offset    uint8
    |  |  +---w size      uint8
    |  +--rw output
    |     +--rw data?   binary
    +---x cmis-write
       +---w input
       |  +---w page      uint8
       |  +---w bank      uint8
       |  +---w offset    uint8
       |  +---w data      binary
       +--rw output
          +--rw status?             enumeration
          +--rw post-write-value?   binary
```

The YANG module of "ietf-cmis-control" is defined as below.

```
<CODE BEGINS> file "ietf-cmis-control-rpc"
module ietf-cmis-control-action {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-cmis-control-action";
  prefix cmis-ctrl-act;

  import ietf-interfaces {
    prefix if;
  }

  organization
    "IETF CCAMP Working Group";

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

    Editor:   Shunsuke Homma
      <mailto:shunsuke.homma.ietf@gmail.com>

    Editor:   Hitoshi Irino
      <mailto:hitoshi.irino.ge@west.ntt.co.jp>";

  description
    "This YANG module defines a data model for action-based
```

```
      management of CMIS (Common Management Interface Specification)
      pages as specified by OIF. It enables configuration and
      retrieval of CMIS page data, including access types and value
      fields, to support the management of pluggable optical modules
      via NETCONF or RESTCONF.

      Copyright (c) 2025 IETF Trust and the persons identified
      as authors of the code.  All rights reserved.

      Redistribution and use in source and binary forms, with or
      without modification, is permitted pursuant to, and
      subject to the license terms contained in, the Revised
      BSD License set forth in Section 4.c of the IETF Trust's
      Legal Provisions Relating to IETF Documents
      (http://trustee.ietf.org/license-info).

      This version of this YANG module is part of RFC XXXX
      (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
      for full legal notices.";

  // RFC Ed.: replace XXXX with actual RFC number and remove this note
    revision "2025-10-11" {
      description "Initial revision.";
    }

    augment "/if:interfaces/if:interface" {
      description
        "Add CMIS read/write actions under interface.";

      action cmis-read {
        description
          "Read CMIS register via action context.";
        input {
          leaf page {
            type uint8;
            mandatory true;
            description "The number of the CMIS page.";
          }
          leaf bank {
            type uint8;
            mandatory true;
            description "The banks corresponding to the CMIS page.";
          }
          leaf offset {
            type uint8;
            mandatory true;
            description "The memory address of the value.";
          }
```

```
            leaf size {
              type uint8{
                range "1 .. 128";
              }
              mandatory true;
              description "The memory size of the value.";
            }
          }
          output {
            leaf data {
              type binary;
              description "Raw register data.";
            }
          }
        }

        action cmis-write {
          description
            "Write CMIS register data via action context.";
          input {
            leaf page {
              type uint8;
              mandatory true;
              description "The number of the CMIS page.";
            }
            leaf bank {
              type uint8;
              mandatory true;
              description "The banks corresponding to the CMIS page.";
            }
            leaf offset {
              type uint8;
              mandatory true;
              description "The memory address of the value.";
            }
            leaf data {
              type binary;
              mandatory true;
              description "Data to write.";
            }
          }
          output {
            leaf status {
              type enumeration {
                enum success;
                enum not-permitted {
                  description "Write request was rejected due to access-type or policy";
                }
```

```
              enum io-error {
                description "I/O error during write";
              }
              enum invalid-params {
                description "Bad parameters";
              }
            }
            description "Result of the write operation.";
          }

          leaf post-write-value {
            type binary;
            description
              "Optional read-back of the target value after write.
               Present only if the implementation performed a read-back
               (e.g., for 'rw' registers). Not present for 'wo' registers
               or when no-readback was requested/possible.";
          }
        }
      }
    }
  }
  <CODE ENDS>
```

3.4. ietf-cmis-control-rpc

The "ietf-cmis-control-rpc" module provides a schema to control CMIS pages with NETCONF RPC.

The tree diagram of "ietf-cmis-control-rpc" is shown below.

```
module: ietf-cmis-control-rpc

  rpcs:
    +---x cmis-read
    |  +---w input
    |  |  +---w interface-name    -> /if:interfaces/interface/name
    |  |  +---w page              uint8
    |  |  +---w bank              uint8
    |  |  +---w offset            uint8
    |  |  +---w size?             uint8
    |  +--ro output
    |     +--ro data?   binary
    +---x cmis-write
       +---w input
       |  +---w interface-name    -> /if:interfaces/interface/name
       |  +---w page              uint8
       |  +---w bank              uint8
       |  +---w offset            uint8
       |  +---w data              binary
       +--ro output
          +--ro status?             enumeration
          +--ro post-write-value?   binary
```

The YANG module of "ietf-cmis-control-rpc" is defined as below.

```
<CODE BEGINS> file "ietf-cmis-control-rpc"
module ietf-cmis-rpc {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-cmis-rpc";
  prefix cmis-rpc;

  import ietf-interfaces {
    prefix if;
  }

  organization
    "IETF CCAMP Working Group";

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

    Editor:   Shunsuke Homma
      <mailto:shunsuke.homma.ietf@gmail.com>

    Editor:   Hitoshi Irino
      <mailto:hitoshi.irino.ge@west.ntt.co.jp>";
```

```
    description
      "This YANG module defines a data model for the management
      of CMIS (Common Management Interface Specification) pages
      as specified by OIF with RPC. It enables configuration and
      retrieval of CMIS page data, including access types and
      value fields, to support the management of pluggable
      optical modules via NETCONF or RESTCONF.

      Copyright (c) 2025 IETF Trust and the persons identified
      as authors of the code.  All rights reserved.

      Redistribution and use in source and binary forms, with or
      without modification, is permitted pursuant to, and
      subject to the license terms contained in, the Revised
      BSD License set forth in Section 4.c of the IETF Trust's
      Legal Provisions Relating to IETF Documents
      (http://trustee.ietf.org/license-info).

      This version of this YANG module is part of RFC XXXX
      (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
      for full legal notices.";

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

    revision "2025-10-11" {
      description
        "Initial revision.";
      reference
        "I-D.hi-ccamp-cmis-control-yang-01";
    }

    rpc cmis-read {
      description
        "Read a CMIS register from a pluggable DCO transceiver.";
      input {
        leaf interface-name {
          type leafref{
            path "/if:interfaces/if:interface/if:name";
          }
          mandatory true;
          description "Target interface name.";
        }
        leaf page {
          type uint8;
          mandatory true;
        }
        leaf bank {
          type uint8;
```

```
          mandatory true;
        }
        leaf offset {
          type uint8;
          mandatory true;
        }
        leaf size {
          type uint8;
          default 1;
          description "Number of bytes to read.";
        }
      }
      output {
        leaf data {
          type binary;
          description "Raw register data.";
        }
      }
    }

    rpc cmis-write {
      description
        "Write CMIS register data to a pluggable module.";
      input {
        leaf interface-name {
          type leafref{
            path "/if:interfaces/if:interface/if:name";
          }
          mandatory true;
          description "Target interface name.";
        }
        leaf page {
          type uint8;
          mandatory true;
          description "The number of the CMIS page.";
        }
        leaf bank {
          type uint8;
          mandatory true;
          description "The banks corresponding to the CMIS page.";
        }
        leaf offset {
          type uint8;
          mandatory true;
          description "The memory address of the value.";
        }
        leaf data {
          type binary;
```

```
            mandatory true;
            description "Data to write.";
          }
        }
        output {
          leaf status {
            type enumeration {
              enum success;
              enum not-permitted {
                description "Write request was rejected due to access-type or policy";
              }
              enum io-error {
                description "I/O error during write";
              }
              enum invalid-params {
                description "Bad parameters";
              }
            }
            description "Result of the write operation.";
          }

          leaf post-write-value {
            type binary;
            description
              "Optional read-back of the target value after write.
               Present only if the implementation performed a read-back
               (e.g., for 'rw' registers). Not present for 'wo' registers
               or when no-readback was requested/possible.";
          }
        }
      }
    }
    <CODE ENDS>
```

3.5. ietf-cmis-monitor

The "ietf-cmis-monitor" module provides monitoring capabilities for CMIS-based DCO transceivers.

The tree diagram of "ietf-cmis-monitor" is shown below.

```
module: ietf-cmis-monitor
  +--rw monitors
     +--rw monitor-rule* [id]
        +--rw id                string
        +--rw interface-name    -> /if:interfaces/interface/name
        +--rw monitor-target
        |  +--rw page      uint8
        |  +--rw bank      uint8
        |  +--rw offset    uint8
        |  +--rw size?     uint8
        +--rw condition
        |  +--rw condition-type    enumeration
        |  +--rw threshold?        decimal64
        |  +--rw delta-rate?       decimal64
        +--rw interval-ms?      uint32
        +--rw enabled?          boolean

  notifications:
    +---n cmis-monitor-event
       +--ro interface-name?   string
       +--ro rule-id?          string
       +--ro monitor-target
       |  +--ro page      uint8
       |  +--ro bank      uint8
       |  +--ro offset    uint8
       |  +--ro size?     uint8
       +--ro condition-type?   enumeration
       +--ro current-value?    binary
       +--ro threshold?        decimal64
       +--ro delta-rate?       decimal64
       +--ro timestamp?        yang:date-and-time
```

The YANG module of "ietf-cmis-monitor" is defined as below.

```
<CODE BEGINS> file "ietf-cmis-control-rpc"
module ietf-cmis-monitor {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-cmis-monitor";
  prefix cmis-mon;

  import ietf-interfaces {
    prefix if;
  }
  import ietf-yang-types {
    prefix yang;
  }

  organization
```

```
       "IETF CCAMP Working Group";

     contact
       "WG Web:   <https://datatracker.ietf.org/wg/ccamp/>
        WG List:  <mailto:ccamp@ietf.org>

       Editor:   Shunsuke Homma
         <mailto:shunsuke.homma.ietf@gmail.com>

       Editor:   Hitoshi Irino
         <mailto:hitoshi.irino.ge@west.ntt.co.jp>";

     description
       "This module provides monitoring capabilities for CMIS-based
        optical modules. Users can define monitor rules for CMIS
        registers identified by page/bank/offset/size. Notifications
        are generated when threshold or delta-rate conditions are met.";

     revision "2025-10-11" {
       description "Initial revision.";
     }

     grouping monitor-target {
       description
         "Target CMIS register to monitor.";
       leaf page {
         type uint8;
         mandatory true;
         description "The number of the CMIS page.";
       }
       leaf bank {
         type uint8;
         mandatory true;
         description "The bank of the CMIS page.";
       }
       leaf offset {
         type uint8;
         mandatory true;
         description "The memory address of the value.";
       }
       leaf size {
         type uint8 {
           range "1 .. 128";
         }
         description "The memory size of the monitored value.";
       }
     }
```

```
     container monitors {
       description "Container for all monitor rules.";

       list monitor-rule {
         key "id";
         description "Monitoring rule.";

         leaf id {
           type string;
           description "Unique identifier of the rule.";
         }

         leaf interface-name {
           type leafref {
             path "/if:interfaces/if:interface/if:name";
           }
           mandatory true;
           description "Target interface of the monitored CMIS module.";
         }

         container monitor-target {
           uses monitor-target;
         }

         container condition {
           description "Condition to trigger notification.";
           leaf condition-type {
             type enumeration {
               enum threshold;
               enum delta-rate;
             }
             mandatory true;
             description "Type of condition.";
           }
           leaf threshold {
             type decimal64 {
               fraction-digits 2;
             }
             description "Threshold value for triggering notification (only used for thre
shold type).";
           }
           leaf delta-rate {
             type decimal64 {
               fraction-digits 2;
             }
             description "Maximum allowed change per interval (only used for delta-rate t
ype).";
           }
         }
```

```
        leaf interval-ms {
          type uint32;
          default 1000;
          description "Monitoring interval in milliseconds.";
        }

        leaf enabled {
          type boolean;
          default true;
          description "Enable or disable this monitor rule.";
        }
      }
    }

    notification cmis-monitor-event {
      description "Notification raised when monitor rule condition is met.";

      leaf interface-name {
        type string;
        description "Interface name of the monitored module.";
      }

      leaf rule-id {
        type string;
        description "ID of the rule that triggered this notification.";
      }

      container monitor-target {
        uses monitor-target;
      }

      leaf condition-type {
        type enumeration {
          enum threshold;
          enum delta-rate;
        }
        description "The type of condition that was met.";
      }

      leaf current-value {
        type binary;
        description "Current value of the monitored register.";
      }

      leaf threshold {
        type decimal64 {
          fraction-digits 2;
        }
```

```
        description "Threshold value (present if threshold type).";
      }

      leaf delta-rate {
        type decimal64 {
          fraction-digits 2;
        }
        description "Delta-rate value (present if delta-rate type).";
      }

      leaf timestamp {
        type yang:date-and-time;
        description "Time when the notification was generated.";
      }
    }
  }
  <CODE ENDS>
```

4. Security Consideration

This YANG allows remote systems to control the equipped pluggable devices directly. It might cause conflict of management of the pluggable devices among the platform node and remote systems. For avoiding this problem, only the minimum necessary CMIS pages (e.g., custom pages) should be exposed to external by using access control features such as [RFC8341].

Regarding to use of the primitive mode, the control rights of the accessible pages are delegated to a controller. Therefore, it is recommended that the mode is used in case that the controller can be trusted, for example, the controlled device and controller are managed by the same operator. Otherwise, specific pages which may affect on data plane signaling should not exposed by using access control features such as [RFC8341]

5. IANA Considerations

This document requests IANA to register the following YANG modules in the "YANG Module Names" registry [RFC6020] within the "YANG Parameters" registry group.

```
   Name:  ietf-cmis-control
   Maintained by IANA?  N
   Namespace:  urn:ietf:params:xml:ns:yang:ietf-cmis-ctrl
   Prefix:  cmis-ctrl
   Reference:  RFC XXXX
```

RFC Editor Note: Please replace XXXX with the RFC number assigned to this document and remove this note.

6. References

6.1. Normative References

[OIF-CMIS] OIF, "Common Management Interface Specification (CMIS) Revision 5.2", April 2022.

[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>.

[RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, DOI 10.17487/RFC7223, May 2014, <https://www.rfc-editor.org/info/rfc7223>.

[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>.

[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>.

6.2. Informative References

[ECOC48923.2020.9333176] Cantono, M., Kamalov, V., Salsi, M., Newland, M., and Z. Zhan, "Sub-Hertz Spectral Analysis of Polarization of Light in a Transcontinental Submarine Cable", European Conference on Optical Communications ECOC 2020, DOI 10.1109/ECOC48923.2020.9333176, December 2020, <https://doi.org/10.1109/ECOC48923.2020.9333176>.

[I-D.rokui-ccamp-actn-wdm-pluggable-modelling] Rokui, R., Guo, A., Bedard, P., Balasundaram, S., and G. Grammel, "Data Modelling and Gap Analysis of Optical Pluggables in Packet Over Optical Network", Work in Progress, Internet-Draft, draft-rokui-ccamp-actn-wdm-pluggable-modelling-03, 9 September 2025, <https://datatracker.ietf.org/doc/html/draft-rokui-ccamp-actn-wdm-pluggable-modelling-03>.

[JLT.2021.3139167]
Sasai, T., Nakamura, M., Yamazaki, E., Yamamoto, S., Nishizawa, H., and Y. Kisaka, "Digital Longitudinal Monitoring of Optical Fiber Communication Link", Journal of Lightwave Technology volume:40, DOI 10.1109/JLT.2021.313917, April 2022, <https://doi.org/10.1109/JLT.2021.313917>.

[JOCN.505729]
Nishizawa, H., Mano, T., Ferreira de Lima, T., Huang, Y., Wang, Z., Ishida, W., Kawashima, M., Ip, E., D'Amico, A., Okamoto, S., Inoue, T., Anazawa, K., Curri, V., Zussman, G., Kilper, D., Chen, T., Wang, T., Asahi, K., and K. Takasugi, "Fast WDM provisioning with minimal probing: the first field experiments for DC exchanges", JOCN 505729, DOI 10.1364/JOCN.505729, February 2024, <https://doi.org/10.1364/JOCN.505729>.

[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>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[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>.

Authors' Addresses

Shunsuke Homma (editor)
NTT
Email: shunsuke.homma.ietf@gmail.com

Hitoshi Irino (editor)
NTT West
Email: hitoshi.irino.ge@west.ntt.co.jp

Toru Mano
NTT
Email: toru.mano@ntt.com

Kazuya Anazawa
NTT
Email: kazuya.anazawa@ntt.com

Yuji Tochio
1Finity
Email: tochio@fujitsu.com