

# ietf-core-comi: CORECONF

## IETF 126 CoRE WG

Vojtěch Vilímek

2026-07-23

# CORECONF

RFC 8342: NMDA [2] defines revised datastore architecture.

The architecture is relatively complex.

In its current specification, the CORECONF protocol does not support the Network Management Datastore Architecture (NMDA).

CORECONF defines a "unified" datastore.

The CORECONF I-D does not specify the unified datastore behavior clearly. This mostly applies to messages writing YANG data instances into the datastore.

# CORECONF unified datastore

## 2.4 Unified datastore

CORECONF supports a simple datastore model consisting of a single unified datastore. This datastore provides access to both configuration and operational data. Configuration updates performed on this datastore are reflected immediately or with a minimal delay as operational data.

More complex datastore models such as the Network Management Datastore Architecture (NMDA) as defined by [[RFC8342](#)] are out of scope of the present specification.

(Copied from [[1](#)])

## CORECONF unified datastore – cont.

Characteristics of the unified datastore are summarized in the table below:

| NAME         | VALUE                                                           |
|--------------|-----------------------------------------------------------------|
| Name         | unified                                                         |
| YANG modules | all modules                                                     |
| YANG nodes   | all data nodes<br>("config true", and "config false")           |
| Access       | read-write                                                      |
| How applied  | changes applied in place immediately<br>or with a minimal delay |
| Protocols    | CORECONF                                                        |
| Defined in   | "ietf-coreconf"                                                 |

Table 2: Characteristics of the Unified Datastore

(Copied from [1])

# CORECONF write – CoAP iPatch

iPATCH /c

Content-Format: TBD1 (application/yang-instances+cbor-seq)

```
{ 1755 : true                                }, / enabled SID 1755 /
{ [1756, "tac.nrc.ca"] : null }, / server SID 1756 /
{
  1756 : {                                    / server SID 1756 /
    3 : "tic.nrc.ca",                        / name SID 1759 /
    4 : true,                                / prefer SID 1760 /
    5 : {                                    / udp SID 1761 /
      1 : "132.246.11.231"                  / address SID 1762 /
    }
  }
}
```

## 3.2.3.1. iPATCH example [1]

# Lack of transactions, atomicity

The iPatch used for writes may write to several points in the tree "at the same time".

We propose explicit groups for related writes that require joint processing. These groups are process sequentially using make-before-break semantics; however this approach does not provide atomicity.

The requests are **not** processed as a transaction.

We focus only on the in-message dependencies.

# unified Behaviour Declarations

## Simple

If an error occurs during processing, the server simply continues to the next item. This approach works well for unrelated data nodes, such as when attempting to configure five independent radios while aiming to push as much configuration as possible.

## Make-before-break group

We place dependent actions within the same group. If any action within the group fails, all remaining group items are skipped. If some action fails, we skip all remaining group items.

This can cover simple dependencies between writes/reads.

## Full atomicity support

To be done in extension draft.

# Make-before-break Request Example

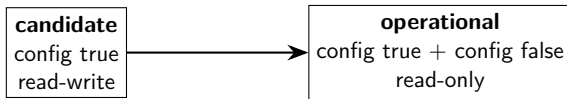
iPATCH /c

Content-Format: TBD1 (application/yang-instances+cbor-seq)

```
{ 1720 : 12                                }, / timeout SID 1720 /  
[                                           / CBOR array defines make-before-break group /  
{ 1755 : true                              }, / enabled SID 1755 /  
{ 1760 : 5683                             }, / port      SID 1760 /  
{ 1770 : [ 1, 2, 3 ]                      }, / values   SID 1770 /  
{ [1756, "tac.nrc.ca"] : null             }, / server   SID 1756 /  
]                                           / CBOR array end /  
{ 1790 : -67                              }, / limit    SID 1790 /
```

Proposed encoding for dependent writes

# Future work – Datastore Extension



**Idea:** Usage of datastores on constrained server

## Links

1. I-D ietf-core-comi (rev 21)
2. RFC 8342: NMDA