

YANG Schema iDentifier,
YANG-CBOR 'instance-identifier' and similar
CoRE Interim 05

Vojtěch Vilímek

2026-04-22

Topics

- ▶ YANG–SID Issues
- ▶ YANG–CBOR 'instance-identifier' draft
- ▶ YANG Stand-ins
- ▶ Pyang maintenance
- ▶ CORECONF

YANG-SID Issues

1. SID assignment for choice and case schema nodes
Already resolved at [previous interim](#). No, there is no need for this.
2. 'item' path format: Should it include choice and case id?
Already resolved at [previous interim](#). No.
Carsten coined a NID term.
3. Sort order of list 'item' entries
Non issue. Enums are sorted by their values. Moreover sort order is not significant.
4. '.sid' File Example in [RFC9595](#)
With changes above only changes needed: some `rpc/action` miss the `input` and `output` node SID assignment.

YANG-SID Issues

Issues raised by Andy Bierman:

1. '.sid' extension was already registered
2. [IANA SID Registry](#) does not have any '.sid' Files
3. `iana-if-type` [has too small SID range allocated](#)
4. `ietf-yang-types` and `ietf-inet-types` update to current [RFC9911](#)
5. YANG Doctor review for entries of IANA YANG SID Registry
6. remove `ietf-coreconf` reference from entry for RFC9595

Other

- ▶ Mention file name conversion for '.sid' Files

YANG-SID Issues

Plan

- ▶ Finalize the text of the erratum. Ask AD to change the [old errata \(EID 8629\)](#).
- ▶ Wait for pyang implementation (upstream or core-wg fork)
There is a repo '[ltn22/pyang](#)', only a partial implementation.
- ▶ Populate [IANA SID Registry](#)

Discussion

YANG-CBOR 'instance-identifier' draft

YANG built-in type 'instance-identifier' is type for indirect references. It must point to a specific instance node in data tree.

```
// file ex.yang
container system {
  container services {
    container ssh {
      leaf port { type uint16; }
    }
  }
  list server {
    key "ip port";
    leaf ip { type inet:ip-address; }
    leaf port { type uint16; }
    leaf service { type string; }
  }
}
```

YANG-CBOR 'instance-identifier' draft

```
/* instance-identifier for a container */
```

```
N: /ex:system/services/ssh (SID 1200)
```

```
S: 1200
```

```
/* instance-identifier for a leaf */
```

```
N: /ex:system/services/ssh/port (SID 1201)
```

```
S: 1201
```

```
/* instance-identifier for a list entry */
```

```
N: /ex:system/server[ip='192.0.0.1'][port='80']\  
  (SID 1230)
```

```
S: [1230, "192.0.0.1", 80]
```

```
/* instance-identifier for a leaf in a list entry */
```

```
N: /ex:system/user[ip='192.0.0.1'][port='80']\  
  /service (SID 1237)
```

```
S: [1237, "192.0.0.1", 80]
```

YANG-CBOR 'instance-identifier' draft

Not covered cases

```
// file ex2.yang
container system {
  list logged-users {
    config false;
    // no key stmt;
    leaf name { type string; }
    leaf number-of-sessions { type uint32; }
  }
  leaf-list if-names { type string; }
  leaf-list measurements { type uint64; }
}
```

YANG-CBOR 'instance-identifier' draft

Proposed YANG-CBOR encoding using SIDs:

```
/* instance-identifier for a list entry with two keys */
```

```
N: /ex2:system/logged-users[2] (SID 1241)
```

```
draft S: [1241, 2]
```

```
/* instance-identifier for a leaf-list entry */
```

```
N: /ex2:system/if-names[.='eth0'] (SID 1260)
```

```
draft S: [1260, "eth0"]
```

```
N: /ex2:system/measurements[.='1440'] (SID 1261)
```

```
draft S: [1261, 1440]
```

For more details see [draft text](#).

Intermezzo

```
container system {  
  list users {  
    key "name";  
    leaf name { type string; }  
    leaf type { type string; /* admin, normal, ... */ }  
    leaf created-at { type yang:date-and-time; }  
  }  
}
```

YANG 'instance-identifier's

```
/ex:system  
/ex:system/ex:users[ex:name="Vojtech"]  
/ex:system/ex:users[ex:name="Vojtech"]/type  
  
/ex:system/ex:users // NOT VALID !!
```

YANG Stand-ins

Most of YANG types are based on string built-in type. This is not ideal for constrained environments. We create an optimization for types that have more natural and/or efficient representation in CBOR.

Examples:

ip4-address, ip6-address, uuid, mac-address, date, date-and-time.

All currently considered types are in [draft text](#).

YANG Stand-ins

Standin file

```
module ietf-cbor-standin-file { container standin-file {  
  list encoding {  
    config false;  
    leaf standin { type identityref; }  
    choice encoding-variant {  
      list types { key "type"; leaf type { type string; }}  
      list sids { key "sid"; leaf sid { type sid:sid; }}  
    }  
  }  
} }
```

YANG Stand-ins

PR: #15

Encode simple cases of enumeration and bits inside union as CBOR unsigned integers. Only if the values are unique.

```
leaf unioned {  
  type union {  
    type string;  
    type enumeration { enum first; enum second; enum third;  
    type bits { bit b1; bit b2; bit b3; }  
  }  
}
```

OLD	NEW
"string-value"	"string-value"
"first"	44(0)
"b1 b3"	43(h'05')

Pyang maintenance

GitHub repo `core-wg/pyang` has some pending PRs.

- ▶ [#20](#) '.sid' File description field
- ▶ [#22](#) Core fix submod name
- ▶ [#23](#) Unify updated 'dependency-revision' list

Code of all PRs is already merged into upstream master and will be part of next version of pyang.

Also there are a lot of open issues that was already been fixed.

More work to be done.

Section 3.4 Event stream

Reception of notification instances is enabled with the CoAP Observe [RFC7641] function. Clients subscribe to the notifications by sending a GET request with an "Observe" option to the stream resource.

Each response payload carries one or multiple notifications.

The number of notifications reported, and the conditions used to remove notifications from the reported list are left to implementers...

Idea: Add text for High Performance Computing? like allow TCP transport in the doc. explicitly.

There are different requirements but I think they lead to similar design decisions.

Data tree walking

- ▶ Instance data (instances in the data tree) discovery. Currently the draft allows only for schema tree discovery.
- ▶ I wrote on the mailing list that this is a problem. [Andy Bierman eventually agreed](#) with me that instance discovery is problem in for the RESTCONF protocol.
- ▶ netmod/netconf WG are working on list pagination draft. A possibility for iterator creation for list nodes.
- ▶ RESTCONF has ability to create filter selecting only specified fields of given container/list.

Misc discussion

Thanks for your attention