

CORECONF

- SID drive
- new CORECONF comi issue
- RFC 9254 instance identifier issue

SID drive

Milestone: generate official SIDs for all YANG RFCMODs

Hackathon: had to merge various development branches

→ 19718 SIDs used in total

Now:

- exclude modules that need manual allocation
- allocate ranges (with margin) → IANA range registry
- generate SID files → IANA module registry

New CORECONF comi issue: still using **rc:yang-data**?

rc:yang-data coreconf-error {

<https://github.com/core-wg/comi/pull/17>: move to **sx:structure**?

- + Use of rc:yang-data is innovation (not copying RESTCONF)
- - sx:structure has been tool support disaster
- - late change — implementation state?
- ? cf. RFC 9290 (problem details)
 - doesn't use identityref, though

RFC 9254 (YANG-CBOR) Instance-Identifier

RFC 9254 derives from RFC 7951 (YANG-JSON)

Can use **SIDs** in various places, e.g., **instance identifiers**:
combine SIDs with predicates (e.g., keys for lists)

Vojtech Vilimek ([draft-vilimek-yang-cbor-inst-id](#)):

RFC 9254 instance identifier text makes no explicit mention of

- leaf-list items (key in JSON = leaf value itself)
- keyless list items (key in JSON = index number, starting at 1)

Clarification? — "do as in YANG-JSON." Put this where?
(Details: e.g., type "empty" — is this all clear from this?)

Corr-Clar

2025-06-20 draft-ietf-core-corr-clar-02

Constrained Application Protocol (CoAP): Corrections and Clarifications

Plan: Work on issues via PRs:

<https://github.com/core-wg/corrclar/issues>

<https://github.com/core-wg/corrclar/issues/51>

May a proxy turn a Uri-Path into a Proxy-Uri?

<https://github.com/core-wg/corrclar/issues/52>

"Critical Option" 4.02 behavior not usable across proxies

👁️ May a proxy turn a Uri-Path into a Proxy-Uri? #51

Proxy-Uri is for talking to **forward** proxies.

Is it "normal" for a client to use Proxy-Uri?

Or last resort if cannot be expressed by Uri-Path etc.?

What if the client is a proxy?

💣 Uri-Path-Abbrev does not use unsafe

(trick: wouldn't otherwise fit in the right sequence)

?!?!?! **unaware** proxy's request might forward UPA along with new Proxy-Uri constructed from other Uri-* options

Hard to instil new/reinforced recommendation into **unaware** proxies

👁️ "Critical Option" 4.02 behavior not usable across proxies #52

NON requests are "rejected" via RST -- can't send 4.02 problem details
Leaves Proxies holding the beer -- can't relay back RST (plus information loss)

Maybe:

Request messages only need to be "rejected" when there is no error response they can be processed into.

RST is really about not having any context → **Correction?**
make that in which document?