

Absolute Requirements

- RFC 7950, Section 5.6.5 says:
A server **MUST NOT** implement more than one revision of a module.
- draft-ietf-netmod-revised-datastores says:
The conventional configuration datastores [...] share exactly the same datastore schema
- draft-ietf-netmod-revised-datastores says:
The datastore schema for <operational> **MUST** be a superset of the combined datastore schema used in all configuration datastores except that YANG nodes supported in a configuration datastore **MAY** be omitted from <operational> if a server is not able to accurately report them.

Objectives 1(2)

- As efficient as possible for a client to consume.

Since the size of the yang library can be quite large, it should be possible for clients to cache the yang library information.

- A dynamic datastore must be able to implement a module or feature that is not implemented in the conventional datastores.
- It must be possible to NOT implement a module or feature in operational, even if it is implemented in some other datastore.

This is required for transition purposes; a server that wants to implement <operational> should not have to implement all modules at once.

Objectives 2(2)

- A given module can only be implemented in one revision in all datastores. If a module is implemented in more than one datastores, the same revision is implemented in all these datastores.
- Multiple revisions can be used for import, if import-by revision is used.
- Nice to have: make it possible to be used by schema mount

Alt A.

Each datastore refers to a schema, and each schema contains a flat list of all modules, features, etc.

```
+--ro yang-library
  +--ro schema* [name]
  |   +--ro name           string
  |   +--ro checksum       string
  |   +--ro module*
  |   ...
  +--ro datastore* [name]
  |   +--ro name           identityref
  |   +--ro schema         -> ../../schema/name
  +--ro checksum           string
```

Alt B.

Each datastore refers to a schema, and each schema contains a list of references to module-sets, and each module-set contains a flat list of all modules, features, etc.

```
+--ro yang-library
  +--ro module-set* [name]
    |   +--ro name           string
    |   +--ro checksum       string
    |   +--ro module* [name]
    |       ...
  +--ro schema* [name]
    |   +--ro name           string
    |   +--ro checksum       string
    |   +--ro module-set*   -> ../../module-set/name
  +--ro datastore* [name]
    |   +--ro name           identityref
    |   +--ro schema         -> ../../schema/name
  +--ro checksum            string
```

Alt C.

Each datastore refers to a schema, and each schema contains a list of references to each module it includes.

(this is the draft -02 model)

```
+--ro yang-library
  +--ro module* [id]
  |   ...
  +--ro schema* [name]
  |   +--ro name      string
  |   +--ro module*  -> ../../module/id
  +--ro datastore* [name]
  |   +--ro name      identityref
  |   +--ro schema    -> ../../schema/name
  +--ro checksum      string
```

Alt D. (from Vladimir)

```

module: ietf-datastores
  +--ro datastores-state
    +--ro datastore* [name]
    +--ro module-set-id          string
    +--ro (model)?
      +--:(same-as-operational)
      +--:(constrained-to-operational)
      |  +--ro not-implemented* [name revision]
      |  |  +--ro name          ->
      |  |  |  /yanglib:module-state/module/name
      |  |  +--ro revision     ->
      |  |  |  /yanglib:module-state/module/revision
      +--:(unconstrained-datastore-instance)
      |  +--ro yang-library-datastore  identityref
      +--:(unconstrained-anydata)
      |  +--ro yang-library?          <anydata>

augment /yanglib:yang-library-change:
  +---- datastore?  identityref

```

Other Alternatives

- One yang-library instance per data, stored in a per-schema meta-datastore.
- A <get-yang-library-data> RPC.