



Document Identifier: DSP2048

Date: 2017-03-03

Version: 0.1.0

Redfish Baseline Switch Model

Supersedes: None

Document Class: Informative

Document Status: Published

Document Language: en-US

Copyright Notice

Copyright © 2017 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. Members and non-members may reproduce DMTF specifications and documents, provided that correct attribution is given. As DMTF specifications may be revised from time to time, the particular version and release date should always be noted.

Implementation of certain elements of this standard or proposed standard may be subject to third party patent rights, including provisional patent rights (herein "patent rights"). DMTF makes no representations to users of the standard as to the existence of such rights, and is not responsible to recognize, disclose, or identify any or all such third party patent right, owners or claimants, nor for any incomplete or inaccurate identification or disclosure of such rights, owners or claimants. DMTF shall have no liability to any party, in any manner or circumstance, under any legal theory whatsoever, for failure to recognize, disclose, or identify any such third party patent rights, or for such party's reliance on the standard or incorporation thereof in its product, protocols or testing procedures. DMTF shall have no liability to any party implementing such standard, whether such implementation is foreseeable or not, nor to any patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a patent owner for such implementations.

For information about patents held by third-parties which have notified the DMTF that, in their opinion, such patent may relate to or impact implementations of DMTF standards, visit <http://www.dmtf.org/about/policies/disclosures.php>.

This document's normative language is English. Translation into other languages is permitted.

CONTENTS

What is a Redfish Baseline Switch? 4

Core YANG RFCs..... 5

1. RFC7223 provides: 6

2. RFC7277 adds: 6

3. RFC7224 provides: 8

4. RFC7317 provides: 8

Additional YANG models 11

5. VLAN and interface extensions: 11

6. ACL 11

7. Syslog..... 11

Applicable Redfish system management models..... 12

Overall Baseline Switch profile structure 13

What is a Redfish Baseline Switch?

The baseline switch profile contains basic system, interface, L2, and L3 configuration elements sufficient to set up the device for use in a controller-based converged infrastructure environment.

The following list of IETF drafts, RFCs, and Redfish models will constitute the management interface to the baseline switch.

Core YANG RFCs

[RFC6020](https://tools.ietf.org/html/rfc6020) provides the YANG modeling language definition.

[RFC6991](https://tools.ietf.org/html/rfc6991) provides the Common YANG Data Types used by many other IETF YANG modules.

Interface management requires at set of RFCs to provide all relevant capabilities:

```
https://tools.ietf.org/html/rfc7223  
https://tools.ietf.org/html/rfc7277  
https://tools.ietf.org/html/rfc7224  
https://tools.ietf.org/html/rfc7317
```

1. RFC7223 provides:

```

+--rw interfaces
| +--rw interface* [name]
|   +--rw name                string
|   +--rw description?        string
|   +--rw type                 identityref
|   +--rw enabled?            boolean
|   +--rw link-up-down-trap-enable? enumeration
+--ro interfaces-state
  +--ro interface* [name]
    +--ro name                string
    +--ro type                 identityref
    +--ro admin-status        enumeration
    +--ro oper-status         enumeration
    +--ro last-change?        YANG:date-and-time
    +--ro if-index            int32
    +--ro phys-address?       YANG:phys-address
    +--ro higher-layer-if*    interface-state-ref
    +--ro lower-layer-if*    interface-state-ref
    +--ro speed?              YANG:gauge64
    +--ro statistics
      +--ro discontinuity-time  YANG:date-and-time
      +--ro in-octets?          YANG:counter64
      +--ro in-unicast-pkts?    YANG:counter64
      +--ro in-broadcast-pkts?  YANG:counter64
      +--ro in-multicast-pkts?  YANG:counter64
      +--ro in-discards?        YANG:counter32
      +--ro in-errors?          YANG:counter32
      +--ro in-unknown-protos?  YANG:counter32
      +--ro out-octets?          YANG:counter64
      +--ro out-unicast-pkts?    YANG:counter64
      +--ro out-broadcast-pkts?  YANG:counter64
      +--ro out-multicast-pkts?  YANG:counter64
      +--ro out-discards?        YANG:counter32
      +--ro out-errors?          YANG:counter32

```

2. RFC7277 adds:

```

+--rw if:interfaces
  +--rw if:interface* [name]

```

```

...
+--rw ipv4!
| +--rw enabled?          boolean
| +--rw forwarding?      boolean
| +--rw mtu?             uint16
| +--rw address* [ip]
| | +--rw ip              inet:ipv4-address-no-zone
| | +--rw (subnet)
| |   +--:(prefix-length)
| |   | +--rw ip:prefix-length?  uint8
| |   +--:(netmask)
| |     +--rw ip:netmask?        YANG:dotted-quad
| +--rw neighbor* [ip]
|   +--rw ip                  inet:ipv4-address-no-zone
|   +--rw link-layer-address  YANG:phys-address
+--rw ipv6!
  +--rw enabled?            boolean
  +--rw forwarding?        boolean
  +--rw mtu?                uint32
  +--rw address* [ip]
  | +--rw ip                 inet:ipv6-address-no-zone
  | +--rw prefix-length     uint8
  +--rw neighbor* [ip]
  | +--rw ip                  inet:ipv6-address-no-zone
  | +--rw link-layer-address  YANG:phys-address
  +--rw dup-addr-detect-transmits?  uint32
  +--rw autoconf
    +--rw create-global-addresses?    boolean
    +--rw create-temporary-addresses?  boolean
    +--rw temporary-valid-lifetime?    uint32
    +--rw temporary-preferred-lifetime? uint32

```

AND

```

+--ro if:interfaces-state
  +--ro if:interface* [name]
    ...
    +--ro ipv4!
    | +--ro forwarding?  boolean
    | +--ro mtu?         uint16
    | +--ro address* [ip]
    | | +--ro ip          inet:ipv4-address-no-zone
    | | +--ro (subnet)?
    | | | +--:(prefix-length)
    | | | | +--ro prefix-length?  uint8
    | | | +--:(netmask)
    | | |   +--ro netmask?        YANG:dotted-quad

```

```

| | +--ro origin?          ip-address-origin
| +--ro neighbor* [ip]
|   +--ro ip              inet:ipv4-address-no-zone
|   +--ro link-layer-address? YANG:phys-address
|   +--ro origin?        neighbor-origin
+--ro ipv6!
  +--ro forwarding?      boolean
  +--ro mtu?             uint32
  +--ro address* [ip]
  | +--ro ip              inet:ipv6-address-no-zone
  | +--ro prefix-length  uint8
  | +--ro origin?        ip-address-origin
  | +--ro status?        enumeration
  +--ro neighbor* [ip]
    +--ro ip              inet:ipv6-address-no-zone
    +--ro link-layer-address? YANG:phys-address
    +--ro origin?        neighbor-origin
    +--ro is-router?     empty
    +--ro state?         enumeration

```

3. RFC7224 provides:

The set of YANG identity statement for the IANA defined interface types.

4. RFC7317 provides:

- System Identification
- System Time Date
- NTP
- DNS Client

System Identification

```

+--rw system
| +--rw contact?          string
| +--rw hostname?        inet:domain-name
| +--rw location?         string
+--ro system-state
  +--ro platform
    +--ro os-name?        string
    +--ro os-release?     string

```



```

+---ro os-version?    string
+---ro machine?      string

```

System Time

```

+---rw system
| +---rw clock
| | +---rw (timezone)?
| |   +---:(timezone-name)
| |     | +---rw timezone-name?    timezone-name
| |     +---:(timezone-utc-offset)
| |       +---rw timezone-utc-offset?  int16
| +---rw ntp!
|   +---rw enabled?    boolean
|   +---rw server* [name]
|     +---rw name      string
|     +---rw (transport)
|       +---:(udp)
|         +---rw udp
|           +---rw address    inet:host
|           +---rw port?     inet:port-number
|       +---rw association-type?  enumeration
|       +---rw iburst?         boolean
|       +---rw prefer?         boolean
+---ro system-state
  +---ro clock
    +---ro current-datetime?    YANG:date-and-time
    +---ro boot-datetime?      YANG:date-and-time

```

DNS Client

```

+---rw system
  +---rw dns-resolver
    +---rw search*    inet:domain-name
    +---rw server* [name]
      | +---rw name    string
      | +---rw (transport)
      |   +---:(udp-and-tcp)
      |     +---udp-and-tcp
      |       +---rw address    inet:ip-address
      |       +---rw port?     inet:port-number
    +---rw options

```

```
+++rw timeout?    uint8
+++rw attempts?  uint8
```

User Authentication

```
+++rw system
+++rw authentication
+++rw user-authentication-order*  identityref
+++rw user* [name]
+++rw name          string
+++rw password?    ianach:crypt-hash
+++rw authorized-key* [name]
+++rw name          string
+++rw algorithm     string
+++rw key-data      binary
```

Additional YANG models

In addition to the above RFCs, the baseline switch model needs to cover:

- VLANs
- ACLs
- Syslog

The following lists of IETF drafts sets our recommendation to cover the above three areas.

5. VLAN and interface extensions:

To handle VLANs and with related interface configuration the following YANG models are under evaluation.

- <https://tools.ietf.org/html/draft-ietf-netmod-intf-ext-yang-03>
- <https://tools.ietf.org/html/draft-wilton-intf-vlan-yang-00>

6. ACL

To handle ACL configuration the following YANG model is under consideration.

- <https://tools.ietf.org/html/draft-ietf-netmod-acl-model-09>

7. Syslog

To handle configuration and access to syslog the following YANG model is under consideration.

- <https://tools.ietf.org/html/draft-ietf-netmod-syslog-model-11>

Applicable Redfish system management models

The following standard Redfish systems management models apply to the baseline network switch profile. Reference: [Redfish schema index](#). The use of these Redfish management models allows a converged infrastructure manager to have a consistent view of server, storage and network systems.

- Chassis
- ComputerSystem
- Manager
- ManagerAccount
- Power
- Thermal
- SoftwareInventory plus UpdateService
- Event configuration using Event, EventDestination, and Event Service
- Access to logs using LogEntry, and LogService
- Management interface configuration using EthernetInterface and related
- Console configuration using SerialInterface
- PrivilegeRegistry and Privileges

Where YANG and Redfish overlap, the commonality of YANG vs Redfish is TBD.

Overall Baseline Switch profile structure

```
./redfish/v1/Systems  
./redfish/v1/Chassis  
./redfish/v1/NetworkDevices/BaselineSwitch/...  
... other redfish resource blocks...  
(resource from RFCs and Redfish bullet list, above)
```