Additional Examples

This chapter contains the following sections:

Information About the API Examples

In the examples, the JSON and XML structures have been expanded with line feeds, spaces, and indentations for readability.

Example: Using the JSON API to Get Top Level System Elements

This example shows how to query the APIC to determine what system devices are present.

General information about the system elements (APICs, spines, and leafs) is contained in an object of class top:System.

This example shows the API query message:

```
GET http://192.0.20.123/api/class/topSystem.json
```

A successful operation returns a response body similar to this example:

```
{
  "imdata" :
  [{
      "topSystem" : {
        "attributes" : {
          "instanceId" : "0:0",
          "address" : "10.0.0.32",
          "childAction" : "",
          "currentTime" : "2013-06-14T04:13:05.584",
          "currentTimeZone" : "",
```

```
            "dn" : "topology/pod-1/node-17/sys",
            "fabricId" : "0",
            "id" : "17",
            "inbMgmtAddr" : "0.0.0.0",
            "lcOwn" : "local",
            "mode" : "unspecified",
            "name" : "leaf0",
            "nodeId" : "0",
            "oobMgmtAddr" : "0.0.0.0",
            "podId" : "1",
            "replTs" : "never",
            "role" : "leaf",
            "serial" : "FOX-270308",
            "status" : "",
            "systemUpTime" : "00:00:02:03"
          }
        }
      }, {
        "topSystem" : {
          "attributes" : {
            "instanceId" : "0:0",
            "address" : "10.0.0.1",
            "childAction" : "",
            "currentTime" : "2013-06-14T04:13:29.301",
            "currentTimeZone" : "PDT",
            "dn" : "topology/pod-1/node-1/sys",
            "fabricId" : "0",
            "id" : "1",
            "inbMgmtAddr" : "0.0.0.0",
            "lcOwn" : "local",
            "mode" : "unspecified",
            "name" : "apic0",
            "nodeId" : "0",
            "oobMgmtAddr" : "0.0.0.0",
            "podId" : "0",
            "replTs" : "never",
            "role" : "apic",
            "serial" : "",
            "status" : "",
            "systemUpTime" : "00:00:02:26"
          }
        }
      }
    ]
}
```


This response indicates that the system consists of one APIC (node-1) and one leaf (node-17).

Example: Using the JSON API to Get Information About a Node

This example shows how to query the APIC to access a node in the system.

To direct an API operation to a specific node device in the fabric, the resource path consists of `/mo/topology/pod-`*number*`/node-`*number*`/sys/` followed by the node component. For example, this URI accesses board sensor 3 in chassis slot B of node 1:

```
GET http://192.0.20.123/api/mo/topology/pod-1/node-1/sys/ch/bslot/board/sensor-3.json
```

A successful operation returns a response body similar to this example:


```
{
  "imdata" :
  [{
      "eqptSensor" : {
        "attributes" : {
          "instanceId" : "0:0",
```

```
            "childAction" : "",
            "dn" : "topology/pod-1/node-1/sys/ch/bslot/board/sensor-3",
            "id" : "3",
            "majorThresh" : "0",
            "mfgTm" : "not-applicable",
            "minorThresh" : "0",
            "model" : "",
            "monPolDn" : "",
            "rev" : "0",
            "ser" : "",
            "status" : "",
            "type" : "dimm",
            "vendor" : "Cisco Systems, Inc."
          }
        }
      }
    ]
}
```

Example: Using the JSON API to Get Running Firmware

This example shows how to query the APIC to determine which firmware images are running.

The detailed information on running firmware is contained in an object of class firmware:IfcRunning, which is a child class (subtree) of the APIC firmware status container class firmware:IfcFwStatusCont. Because there can be multiple running firmware instances (one per APIC instance), you can query the container class and filter the response for the subtree of running firmware objects.

This example shows the API query message:

```
GET http://192.0.20.123/api/class/firmwareIfcFwStatusCont.json?
        query-target=subtree
        &
        target-subtree-class=firmwareIfcRunning
```

A successful operation returns a response body similar to this example:

```
{
  "imdata" : [{
      "firmwareIfcRunning" : {
        "attributes" : {
          "instanceId" : "0:0",
          "applId" : "3",
          "childAction" : "",
          "dn" : "ifcfwstatuscont/ifcrunning-3",
          "lcOwn" : "local",
          "replTs" : "never",
          "rn" : "",
          "status" : "",
          "ts" : "2012-12-31T16:00:00.000",
          "type" : "ifc",
          "version" : "1.1"
        }
      }
    }, {
      "firmwareIfcRunning" : {
        "attributes" : {
          "instanceId" : "0:0",
          "applId" : "1",
          "childAction" : "",
          "dn" : "ifcfwstatuscont/ifcrunning-1",
          "lcOwn" : "local",
          "replTs" : "never",
          "rn" : "",
          "status" : "",
```

```
          "ts" : "2012-12-31T16:00:00.000",
          "type" : "ifc",
          "version" : "1.1"
        }
      }
    }, {
      "firmwareIfcRunning" : {
        "attributes" : {
          "instanceId" : "0:0",
          "applId" : "2",
          "childAction" : "",
          "dn" : "ifcfwstatuscont/ifcrunning-2",
          "lcOwn" : "local",
          "replTs" : "never",
          "rn" : "",
          "status" : "",
          "ts" : "2012-12-31T16:00:00.000",
          "type" : "ifc",
          "version" : "1.1"
        }
      }
    }
  ]
}
```

This response describes three running instances of APIC firmware version 1.1.

Example: Using the JSON API to Add a Leaf Port Selector Profile

This example shows how to add a leaf port selector profile.

As shown in the *Cisco APIC Management Information Model Reference*, this hierarchy of classes forms a leaf port selector profile:

- fabric:LePortP — A leaf port profile is represented by a managed object (MO) of this class, which has a distinguished name (DN) format of `uni/fabric/leportp-[name]`, in which `leportp-[name]` is the relative name (RN). The leaf port profile object is a template that can contain a leaf port selector as a child object.
  - fabric:LFPortS — A leaf port selector is represented by an MO of this class, which has a RN format of `lefabports-[name]-typ-[type]`. The leaf port selector object can contain one or more ports or ranges of ports as child objects.
    - fabric:PortBlk — A leaf port or a range of leaf ports is represented by an MO of this class, which has a RN format of `portblk-[name]`.

The API command that creates the new leaf port selector profile MO can also create and configure the child MOs.

This example creates a leaf port selector profile with the name "MyLPSelectorProf." The example profile contains a selector named "MySelectorName" that selects leaf port 1 on leaf switch 1 and leaf ports 3 through 5 on leaf switch 1. To create and configure the new profile, send this HTTP POST message:

```
POST http://192.0.20.123/api/mo/uni/fabric/leportp-MyLPSelectorProf.json

{
  "fabricLePortP" : {
    "attributes" : {
      "descr" : "Selects leaf ports 1/1 and 1/3-5"
    },
    "children" : [{
```

```
        "fabricLFPortS" : {
          "attributes" : {
            "name" : "MySelectorName",
            "type" : "range"
          },
          "children" : [{
              "fabricPortBlk" : {
                "attributes" : {
                  "fromCard" : "1",
                  "toCard" : "1",
                  "fromPort" : "1",
                  "toPort" : "1",
                  "name" : "block2"
                }
              }
            }, {
              "fabricPortBlk" : {
                "attributes" : {
                  "fromCard" : "1",
                  "toCard" : "1",
                  "fromPort" : "3",
                  "toPort" : "5",
                  "name" : "block3"
                }
              }
            }
          ]
        }
      }
    ]
  }
}
```

A successful operation returns this response body:

```
{
  "imdata" : [{
      "fabricLePortP" : {
        "attributes" : {
          "instanceId" : "0:0",
          "childAction" : "deleteNonPresent",
          "descr" : "Select leaf ports 1/1 and 1/3-5",
          "dn" : "uni/fabric/leportp-MyLPSelectorProf",
          "lcOwn" : "local",
          "name" : "MyLPSelectorProf",
          "replTs" : "never",
          "rn" : "",
          "status" : "created"
        },
        "children" : [{
            "fabricLFPortS" : {
              "attributes" : {
                "instanceId" : "0:0",
                "childAction" : "deleteNonPresent",
                "dn" : "",
                "lcOwn" : "local",
                "name" : "MySelectorName",
                "replTs" : "never",
                "rn" : "lefabports-MySelectorName-typ-range",
                "status" : "created",
                "type" : "range"
              },
              "children" : [{
                  "fabricPortBlk" : {
                    "attributes" : {
                      "instanceId" : "0:0",
                      "childAction" : "deleteNonPresent",
                      "dn" : "",
                      "fromCard" : "1",
                      "fromPort" : "3",
                      "lcOwn" : "local",
                      "name" : "block3",
```

```
                    "replTs" : "never",
                    "rn" : "portblk-block3",
                    "status" : "created",
                    "toCard" : "1",
                    "toPort" : "5"
                  }
                }
              }, {
                "fabricPortBlk" : {
                  "attributes" : {
                    "instanceId" : "0:0",
                    "childAction" : "deleteNonPresent",
                    "dn" : "",
                    "fromCard" : "1",
                    "fromPort" : "1",
                    "lcOwn" : "local",
                    "name" : "block2",
                    "replTs" : "never",
                    "rn" : "portblk-block2",
                    "status" : "created",
                    "toCard" : "1",
                    "toPort" : "1"
                  }
                }
              }
            ]
          }
        }
      ]
    }
  }
}
```

To delete the new profile, send an HTTP POST message with a fabricLePortP attribute of `"status":"deleted"`, as in this example:

```
POST http://192.0.20.123/api/mo/uni/fabric/leportp-MyLPSelectorProf.json

{
  "fabricLePortP" : {
    "attributes" : {
      "status" : "deleted"
    }
  }
}
```

Alternatively, you can send this HTTP DELETE message:

```
DELETE http://192.0.20.123/api/mo/uni/fabric/leportp-MyLPSelectorProf.json
```