# Redfish Tasks

Jeff Hilland
HPE

# Agenda

- Task Overview
- Task Resource Overview and Map
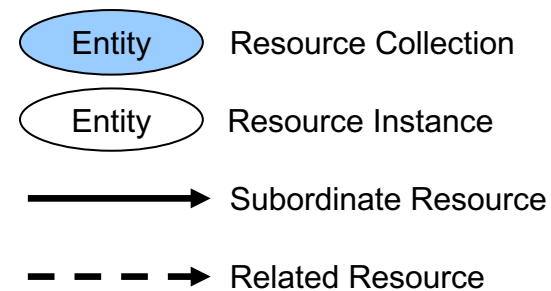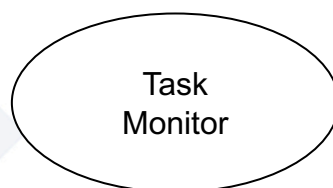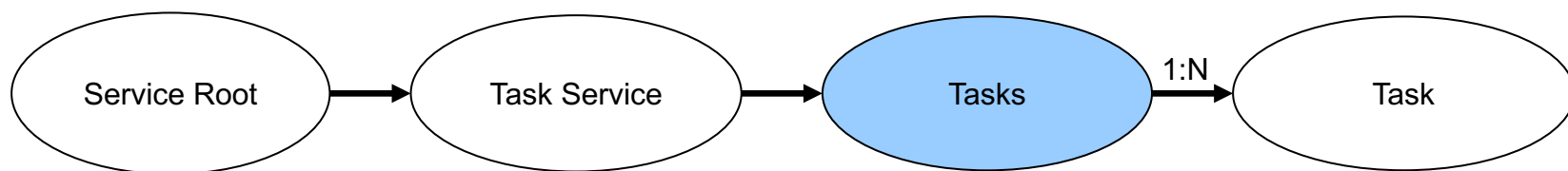- Examples
  - Task Service
  - Task

# Tasks

- Any operation could end up as an Asynchronous Operation
    - If the operation has been accepted for processing, a 202 can be returned. This mean a Task has been started to complete the request
        - The "Location" header is set to the URI of a Task Monitor.
        - There may also be a "Retry-After" header to specify the amount of time the client should wait before querying the Task Monitor
        - The Response Body should contain the Task Resource and not the expected Resource.
- What is a Task Monitor?
    - The Task Monitor is used by the Client to get the Response Body of the actual operation.
    - While the operation is going on, a 202 will be returned.
    - When the operation is finished, the Response Body will be returned with the content the same as if the Request had been synchronous.
- What's the Task Resource for then?
    - The Task Resource can be used by entities other than the client to track the operation.
        - The Client can use either the Task Resource or the Task Monitor, but the Task Monitor is probably preferable since it would have the expected Response Body but the Task Resource will not.
- Canceling is done via the DELETE operation
    - To either the Task Resource or the Task Monitor

# Resource Overview

- Task Service: A representation the event service inside of the Redfish Service
  - Contains Attributes of the service:
    - Such as status and overwrite policy.
  - Contains a collection of Tasks
    - The tasks returned to monitor asynchronous operations are the members of this collection
- Tasks: A representation of the "Task Resource"
  - Contains Attributes of the Task:
    - Started, Stopped times; completion state and any warning or error messages.
- Task Monitor.
  - What the Client "polls" on (if eventing isn't supported on task completion and the client has
  - Contains the completion format of the resource.  In other words, the expected Response Body from the initial request as if it had been performed synchronously.

# Task Service Model Map



**Diagram:**

Service Root → Task Service → Tasks —1:N→ Task

Task Monitor

**Legend:**
- Entity (blue filled ellipse) — Resource Collection
- Entity (white ellipse) — Resource Instance
- → Subordinate Resource
- - - → Related Resource

# Task Service

```
{
    "@odata.context": "/redfish/v1/$metadata/TaskService.TaskService",
    "@odata.id": "/redfish/v1/TaskService",
    "@odata.type": "#TaskService.v1_0_0.TaskService",
    "Id": "TaskService",
    "Name": "Tasks Service",
    "DateTime": "2015-03-13T04:14:33+06:00",
    "CompletedTaskOverWritePolicy": "Manual",          Overwrite Policy &
    "LifeCycleEventOnTaskStateChange": true,           Eventing
    "Status": {
        "State": "Enabled",
        "Health": "OK"
    },
    "ServiceEnabled": true,
    "Tasks": {                                          Tasks
        "@odata.id": "/redfish/v1/TaskService/Tasks"
    }
}
```

# Task

```json
{
    "@odata.context": "/redfish/v1/$metadata#Task.Task",
    "@odata.id": "/redfish/v1/TaskService/Tasks/1",
    "@odata.type": "#Task.v1_0_0.Task",
    "Id": "1",
    "Name": "Task 1",
    "TaskState": "Completed",
    "StartTime": "2012-03-07T14:44+06:00",
    "EndTime": "2012-03-07T14:45+06:00",
    "TaskStatus": "OK",
    "Messages" : [
        {
            "MessageId": "Base.1.0.PropertyNotWriteable",
            "RelatedProperties": [ "#/SKU" ],
            "MessageArgs": [ "SKU" ],
        }
    ]
}
```

**Task State**

**Start & Stop Time**

**Task Status**

**Message info that would have been returned in the ExtendedInfo section of a response**
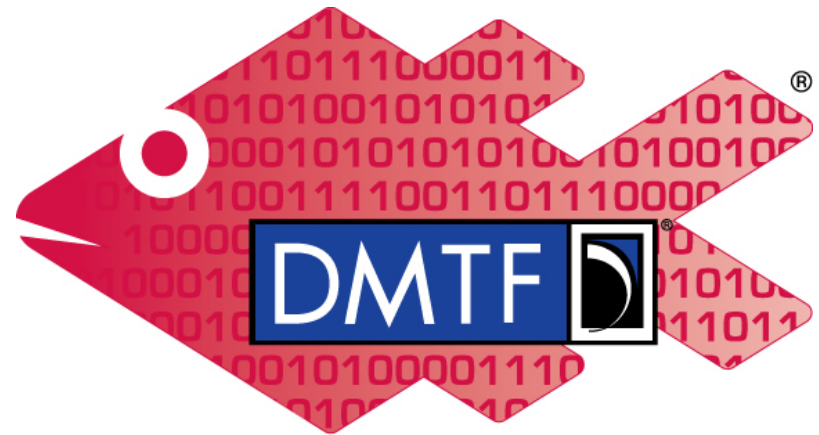
# Task Monitor

- We can't really show you a Task Monitor
  - The client only polls and looks for completion that is not a 202
  - When the return code is something other than a 202, normal completion processing is followed:
    - If it's a 200, the body will be just like the operation was synchronous.  So if it was a PATCH to a ComputerSystem resource, the body would be that ComputerSystem resource body.  It may also have an ExtendedInfo object, just like any other response.
    - If it's something else, like a 400, the client can parse the body for for any ExtendedInfo to find error and warning messages, just like with any other operation.
    - The Task Resource itself may be deleted by the service once the Task Monitor is read after the task has completed.

# Thank you for watching!

- Redfish Standards
  - Schemas, Specs, Mockups, White Papers, FAQ, Educational Material & more
  - http://www.dmtf.org/standards/redfish
- Redfish Developer Hub
  - Redfish Interactive Explorer, Hosted Schema at Namespace & other links
  - http://redfish.dmtf.org
- SPMF (WG that defines Redfish)
  - Companies involved, Upcoming Schedules & Future work, Charter, Information on joining.
  - http://www.dmtf.org/standards/spmf