Telemetry 2.0: Report Profiles

Telemetry 2 brings multi-profile support to RDK telemetry. This allows for multiple report profiles to be active on a device at one time, each with their own data and reporting configuration. Specific profiles may be targeted to data for specific purposes, such as WiFi Happiness, DOCSIS information or many other areas. Telemetry 2.0 Report Profiles are supported along with legacy Telemetry 1.0/1.1 configurations, so that T2.0 Report Profiles can be actively added and removed without disturbing legacy telemetry.

In addition, Telemetry 2 adds support for RDK Components to send their telemetry data to the T2 component for reporting. This allows for moving away from the CPU-intensive process of searching through log files for telemetry strings.

Telemetry 2.0 allows for an RDK device to be updated with a set of T2 profiles at any time. See Profile Sets for more information.

Telemetry 2.0 profiles are authored in JSON. Detailed descriptions of the JSON format and properties can be found here: Telemetry 2.0 Report Profile Description

There is a web-based tool called Report Profile Generator to help with creation of Report Profiles. It will take inputs and generate the properly formatted JSON. (You must be on the VPN to access GitHub Enterprise to view Report Profile Generator.)

> Tip: Online tools like https://jsonlint.com/ (for simple JSON syntax validation) and https://www.jsonschemavalidator.net/ (for validating the profile JSON against the schema) are great for helping with JSON validation and formatting. See GHE for the T2 schema to use, or use the XConf CI UI that will automatically validate the JSON against the schema.

Telemetry 2.0 Report Profile schemas are in GitHub Enterprise (You must be on the VPN to access GitHub Enterprise) .

Anatomy of A Report Profile

A Telemetry 2 profile contains a set of properties at the root. These properties define information for the processing of the profile on the CPE, such as:

- When the report should be generated (ReportingInterval, ReportingAdjustments, ActivationTimeOut, DeleteOnTimeOut, GenerateNow, TriggerCondition and TimeReference)
- How the report should be sent to the cloud (Protocol)
- How the report should be encoded and compressed (RootName and EncodingType)
- Descriptive information about the profile (Description and Version)
- Parameters describe the data that should be gathered and included in the generated report (Parameter)

The generated report contains no default data. Any information desired in the final report must be defined as a parameter object in the report profile.

Telemetry 2.0 Generated Reports

A report generated by Telemetry 2.0 will be in the following format, where each name value pair contained in the report corresponds to a parameter defined within the corresponding report profile.

```
{
    "Report": [{"name":"value"}]
}
```

For instance, a report profile that defines a Parameter array as follows:

```
"Parameter": [{
            "type": "dataModel", "name": "CmMac",
            "reference": "Device.DeviceInfo.X_COMCAST-COM_CM_MAC"
        }, {
            "type": "dataModel", "name": "ClearResetCount",
            "reference": "Device.DeviceInfo.ClearResetCount"
        }, {
            "type": "dataModel", "name": "Desc",
            "reference": "Device.DeviceInfo.Description"
        }, {
            "type": "dataModel",
            "reference": "Device.DeviceInfo.FactoryResetCount"
        }, {
            "type": "dataModel", "name": "MemFree",
            "reference": "Device.DeviceInfo.Hardware_MemFree"
        }, {
            "type": "dataModel", "name": "MemUsed",
            "reference": "Device.DeviceInfo.Hardware_MemUsed"
        }, {
            "type": "dataModel", "name": "HwVer",
            "reference": "Device.DeviceInfo.HardwareVersion"
        }, {
            "type": "dataModel", "name": "MemStatFree",
            "reference": "Device.DeviceInfo.MemoryStatus.Free"
        }, {
            "type": "dataModel", "name": "MemStatTotal",
            "reference": "Device.DeviceInfo.MemoryStatus.Total"
        }, {
            "type": "dataModel", "name": "MemStatUsed",
            "reference": "Device.DeviceInfo.MemoryStatus.Used"
        }, {
            "type": "dataModel", "name": "FrMemThrshld",
            "reference": "Device.DeviceInfo.MemoryStatus.X_RDKCENTRAL-COM_FreeMemThreshold"
        }, {
            "type": "dataModel", "name": "CpuUsage",
            "reference": "Device.DeviceInfo.ProcessStatus.CPUUsage"
        }, {
            "type": "dataModel", "name": "Uptime",
            "reference": "Device.DeviceInfo.UpTime"
        }]
```

Would generate a report that looks like this:

```
{
    "Report": [
        { "CmMac": "XX:XX:XX:XX:XX:XX" },
        { "ClearResetCount": "false" },
        { "Desc": "DOCSIS 3.1 2-port Voice Gateway" },
        { "Device.DeviceInfo.FactoryResetCount": "0" },
        { "MemFree": "1876" },
        { "MemUsed": "172" },
        { "HwVer": "2.2" },
        { "MemStatFree": "432" },
        { "MemStatTotal": "1024" },
        { "MemStatUsed": "592" },
        { "FrMemThrshld": "10" },
        { "CpuUsage": "8" },
        { "Uptime": "30837" }
    ]
}
```

See Telemetry 2.0 Report Profile Description for more information.