|  |
| --- |
|  |

|  |  |
| --- | --- |
| CHANGE REQUEST | |
| Meeting ID:\* | RDM#49 |
| Source:\* | Cyrille Bareau, Orange, [cyrille.bareau@orange.com](mailto:cyrille.bareau@orange.com)  Andreas Kraft, Deutsche Telekom, [Andreas.Kraft@t-systems.com](mailto:Andreas.Kraft@t-systems.com)  Marianne Mohali, Orange, [marianne.mohali@orange.com](mailto:marianne.mohali@orange.com) |
| Date:\* | 2021-01-26 |
| Reason for Change/s:\* | See the introduction. |
| CR against: Release\* | Release 4 |
| CR against: WI\* | Active WI-0084  MNT maintenance / < Work Item number(optional)>  Is this a mirror CR? Yes  No  mirror CR number: (Note to Rapporteur - use latest agreed revision)  STE Small Technical Enhancements / < Work Item number (optional)>  Only ONE of the above shall be ticked |
| CR against: TS/TR\* | TS-0023 4.7.0 |
| Clauses \* | Clauses 5.8.3, 5.8.5, 5.8.6, 6.2.4, 6.3.4 |
| Type of change: \* | Editorial change  Bug Fix or Correction  Change to existing feature or functionality  New feature or functionality  Only ONE of the above shall be ticked |
| Other TS/TR(s) impacted | N/A |
| Post Freeze checking:\* | This CR contains only essential changes and corrections? YES  NO  This CR may break backwards compatibility with the last approved version of the TS? YES  NO |
| Template Version: January 2020 (do not modify) | |

**oneM2M Notice**

The document to which this cover statement is attached is submitted to oneM2M. Participation in, or attendance at, any activity of oneM2M, constitutes acceptance of and agreement to be bound by terms of the Working Procedures and the Partnership Agreement, including the Intellectual Property Rights (IPR) Principles Governing oneM2M Work found in Annex 1 of the Partnership Agreement.

GUIDELINES for Change Requests:

Provide an informative introduction containing the problem(s) being solved, and a summary list of proposals.

Each CR should contain changes related to only one particular issue/problem.

If this is a correction, and the change applies to previous releases, a separate “mirror CR” should be posted at the same time as this CR

Mirror CR: applies only when the text, including clause numbering are exactly the same.

Companion CR: applies when the change means the same but the baselines differ in some way (e.g. clause number).

Follow the principle of completeness, where all changes related to the issue or problem within a deliverable are simultaneously proposed to be made e.g. a change impacting 5 tables should not only include a proposal to change only 3 tables. Include any changes to references, definitions, and abbreviations in the same deliverable.

Follow the drafting rules.

All pictures must be editable.

Check spelling and grammar.

Use change bars for modifications.

The change should include the current and surrounding clauses to clearly show where a change is located and to provide technical context of the proposed change. Additions of complete clauses need not show surrounding clauses as long as the proposed clause number clearly shows where the proposed new clause is located.

Multiple changes in a single CR shall be clearly separated by horizontal lines with embedded text such as, start of change 1, end of change 1, start of new clause, end of new clause.

When subsequent changes are made to the content of a CR, then the accepted version should not show changes over changes. The accepted version of the CR should only show changes relative to the baseline approved text.

## Introduction

SDT Actions are defined with a DataType, i.e. they can return a single typed value, but this returned value is designed without any name.

When mapping an Action to a <flexContainer> resource, this return value must be mapped to a named custom attribute. We propose here to force this custom attribute name to ‘result’, rather than letting returned value names be defined for each specific Action.

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of change 1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### 5.8.3 dmAgent

This ModuleClass is the entry point module of [*flexNode*]; it provides capabilities to control and monitor the Device Management of the device.

Table 5.8.3-1 Actions of dmAgent ModuleClass

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Return Type | Name | Arguments | Optional | Description |
| none | reboot | rebootType: hd:enumRebootType | false | Execute a reboot or a factory reset |
| M2MID | deployPackage | name: xs:string  version: xs:string  url: xs:url | true | Create a dmPackage.  Return the ID of the created package. |

The *deployPackage* action allows creating a new [dmPackage] module class (see clause 5.8.9), child of this dmAgent’s parent *flexNode*. The returned value is the ID of this created <flexContainer>. The created dmPackage is in NotInstalled state.

The DataPoints of dmAgent Module Class are as follows:

‘state’ represents the state of the agent for DM purposes (ready, sleeping, etc.).

some optional device properties which can be used for Device Management purpose. The dmAgent can be seen as a ‘dashboard’ that gathers common information such as battery level, memory or CPU usage…

Table 5.8.3-2 DataPoints of dmAgent ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Description |
| state | hd:enumDmAgentState | R | false |  | The current state of the agent (ready, error, etc.) |
| storageAvailable | xs:integer | R | true | KB | The size of available storage memory. |
| storageTotal | xs:integer | R | true | KB | The size of total storage memory. |
| ramAvailable | xs:integer | R | true | KB | The size of available RAM memory. |
| ramTotal | xs:integer | R | true | KB | Total size of the RAM memory. |
| powerStatus | hd:enumPowerState | R | true |  | The status of the electrical power. |
| cpuUsage | xs:integer | R | true | % | Current CPU usage in percent. |
| systemTime | m2m:timestamp | RW | true |  | Reference time for the device. |

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of change 1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of change 2 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### 5.8.5 dmDataModelIO

This ModuleClass provides capabilities to handle the device’s Data Model for cases where the underlying Device Management technology supports APIs that are not directly reflected in the *flexNode* modules.

Table 5.8.5-1 Actions of dmDataModelIO ModuleClass

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Return Type | Name | Arguments | Optional | Description |
| xs:string | readIO | address: xs:string | true | Read the current values of parameters. Argument: the list of the parameter names.  Returns a JSON serialization of the parameters (see Rules 3-2 and 3-6 in clause 6.2.4). |
| xs:string | writeIO | address: xs:string  payload: xs:string | true | Update the current values of parameters. Arguments:  ‘address’: the list of the parameter names,  ‘payload’: the list of the parameter values.  Returns the list of the modified parameter names. |

<…>

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of change 2 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of change 3 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### 5.8.6 dmFirmware

This ModuleClass provides Device Management capabilities to control and monitor the firmware of a device.

The device can contain multiple components (a graphic card for instance) that can have individual firmwares, and they need to be managed separately. The [*flexNode*] allows one [*dmFirmware*] module per component plus one ‘major’ [*dmFirmware*] for the device itself.

Individual firmwares are managed using the [*dmFirmware*] actions presented in Table 5.8.6-1.

Table 5.8.6-1 Actions of dmFirmware ModuleClass

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Return Type** | **Name** | **Argument** | **Optional** | **Description** |
| xs:string | updateFirmware | url: xs:url  version: xs:string | true | Downloads a new firmware to the device / sub-component. In case of devices that do support toggling between multiple preinstalled firmware versions it also starts the firmware flashing/installation process.  The updateFirmware action as it results returns an AE/IPE message indicating if the action was successful or not. |
| xs:string | toggle | none | true | Toggles between the firmware versions installed on a device/sub-component. In case of devices that do not support such toggling, it triggers the firmware flashing/installation process.  The toggle action as it results returns an AE/IPE message indicating if the action was successful or not. |

<…>

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of change 3 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of change 4 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### 6.2.4 Resource mapping for Action

Actions defined as part of a ModuleClass model shall be mapped to the specializations of a <flexContainer> resource. The following rules shall be applied:

* Rule 3-1: The *containerDefinition* attribute shall be set according to 6.4.4.
* Rule 3-2: When the Action supports any 'Arguments', they are mapped to [customizedAttribute] with their variable names (short names are given in clause 6.3.4). When the Action supports a 'Return Type', it is mapped to a [customizedAttribute] named ‘result’ (short name ‘resut’). The keyword ‘result’ is reserved and cannot be used as an Argument name.
* Rule 3-3: XSD file for each Action shall be named according to 6.5.4.
* Rule 3-4: The Action shall be triggered:
  + by updating at least one of the Arguments custom attributes with any value, if the action has at least one argument, or
  + by updating the <flexContainer> resource with *empty content* if it has no argument
* Rule 3-5: The *resourceName* attribute for each Action model that appears as a child of a ModuleClass model shall be CREATED with the value set to “Action name”.
* Rule 3-6: If an action returns a value that is of a complex data type, i.e. not one of the standard scalar types, then this value shall be encoded as a JSON structure and returned serialized in an xs:string.

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of change 4 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of change 5 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### 6.3.4 Resource attributes for actions arguments

In protocol bindings resource attributes names for arguments of actions shall be translated into short names of Table 6.3.4-1.

Table 6.3.4‑1: Resource attribute short names (Action arguments)

|  |  |  |
| --- | --- | --- |
| Argument Name | Occurs in | Short Name |
| address | readIO, writeIO | ***addrs*** |
| name | deployPackage | ***name*** |
| payload | writeIO | ***payld*** |
| rebootType | reboot | ***rebTe*** |
| url | deployPackage, updateFirmware, update | ***url*** |
| version | deployPackage, updateFirmware, update | ***versn*** |



### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of change 5 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*