International Telecommunication Union

ITU-T Y.4500.6

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(03/2018)

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

Internet of things and smart cities and communities – Frameworks, architectures and protocols

oneM2M management enablement (BBF)

Recommendation ITU-T Y.4500.6

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

GLOBAL INFORMATION INFRASTRUCTURE	
General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899
IPTV over NGN	Y.1900–Y.1999
NEXT GENERATION NETWORKS	
Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Enhancements to NGN	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Network control architectures and protocols	Y.2500–Y.2599
Packet-based Networks	Y.2600–Y.2699
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899
Carrier grade open environment	Y.2900–Y.2999
FUTURE NETWORKS	Y.3000–Y.3499
CLOUD COMPUTING	Y.3500–Y.3999
INTERNET OF THINGS AND SMART CITIES AND COMMUNITIES	
General	Y.4000–Y.4049
Definitions and terminologies	Y.4050–Y.4099
Requirements and use cases	Y.4100–Y.4249
Infrastructure, connectivity and networks	Y.4250–Y.4399
Frameworks, architectures and protocols	**Y.4400–Y.4549**
Services, applications, computation and data processing	Y.4550–Y.4699
Management, control and performance	Y.4700–Y.4799
Identification and security	Y.4800–Y.4899
Evaluation and assessment	Y.4900–Y.4999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T Y.4500.6

oneM2M management enablement (BBF)

Summary

Recommendation ITU-T Y.4500.6 specifies the usage of the BBF TR-069 protocol and the corresponding message flows including normal as well as error cases to fulfil oneM2M management requirements.

- Protocol mapping between the oneM2M service layer and BBF TR-069 protocol. The Mca reference point, ms interface and la interface are possibly involved in this protocol mapping.
- Mapping between oneM2M management-related resources and the BBF TR-069 protocol remote procedure calls (RPCs) and BBF TR-181i2 data model.
- Specification of new BBF TR-181 data model elements to fulfil oneM2M specific management requirements that cannot be currently translated.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.4500.6	2018-03-01	20	11.1002/1000/13502

Keywords

Broadband Forum, oneM2M, device management.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

NOTE – This Recommendation departs slightly from the usual editorial style of ITU-T Recommendations to preserve existing cross-referencing from external documents.

Table of Contents

Recommendation ITU-T Y.4500.6

oneM2M management enablement (BBF)

1 Scope

This Recommendation describes the protocol mappings between management resources for oneM2M and the BBF TR-181i2 data model.

This Recommendation contains oneM2M Release 2 specification – oneM2M Management Enablement (BBF) V2.0.1 and is equivalent to standards of oneM2M partners including ARIB, ATIS [b-ATIS.oneM2M.TS0006V201], CCSA [b-CCSA M2M-TS-0006-V2.0.1], ETSI [b-ETSI TS 118 106], TIA, TSDSI [b-TSDSI STD T1.oneM2M TS-0006-2.0.1 V1.0.0], TTA [b-TTAT.MM-TS.0006 v2.0.1] and TTC [b-TTC TS-M2M-0006v2.0.1].

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.4500.1] Recommendation ITU-T Y.4500.1 (2018), *oneM2M – Functional architecture*.

[ITU-T Y.4500.4] Recommendation ITU-T Y.4500.4 (2018), *Service layer core protocol specification*.

[ITU-T Y.4500.11] Recommendation ITU-T Y.4500.11 (2018), *Common terminology*.

[BBF TR-069] TR-069, Issue 1, Amendment 6 (2018), *CPE WAN management protocol*.

[BBF TR-106] TR-106, Issue 1, Amendment 7 (2013), *Data model template for TR-069-enabled devices*.

[BBF TR-131] TR-131, Issue 1, Amendment 1 (2015), *ACS northbound interface requirements*.

[BBF TR-181] TR-181, Issue 2, Amendment 11 (2016), *Device data model for TR-069*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

For the purposes of this Recommendation, the terms and definitions given in oneM2M TS-0011 [ITU-T Y.4500.11] and the following apply.

3.1.1 application entity [ITU-T Y.4500.11]: represents an instantiation of application logic for end-to-end M2M solutions.

3.1.2 common services entity (CSE) [ITU-T Y. 4500.11]: Represents an instantiation of a set of common service functions of the M2M environments. Such service functions are exposed to other entities through reference points.

3.1.3 CPE proxier [BBF TR-069]: A CPE that is capable of proxying the communication between an ACS and a proxied device.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

For the purposes of this Recommendation, the abbreviations given in oneM2M TS-0011 [ITU-T Y.4500.11] and the following apply:

ACS Auto-Configuration Server

ADN Application-Dedicated Node

AE Application Entity

ASN Application Service Node

CMDH Communication Management and Delivery Handling

CPE Customer Premises Equipment

CWMP Common premises equipment Wide area network Management Protocol

DM Device Management

DU Deployment Unit

IN-CSE Infrastructure Node-Common Services Entity

LAN Local Area Network

MN Middle Node

MoCA Multimedia over Coax Alliance

OUI Organizationally Unique Identifier

PC Product Class

RPC Remote Procedure Call

SN Serial Number

UPA Universal Powerline Association

URI Uniform Resource Identifier

URL Uniform Resource Locator

USB Universal Serial Bus

UUID Universal Unique Identifier

XML Extensible Markup Language

Wi-Fi Wireless Fidelity

5 Conventions

The keywords "shall", "shall not", "should", "should not", "may", "need not" in this Recommendation are to be interpreted as follows.

Shall/shall not:

Requirements

1) effect on this Recommendation: This Recommendation needs to describe the required feature (i.e., specify a technical solution for the requirement);
2) effect on products: every implementation (M2M Solution that complies with this Recommendation) must support it;
3) effect on deployments: every deployment (M2M service based on this Recommendation) must use the standardized feature where applicable – otherwise interoperability problems with other services could arise, for example.

Should/should not

Recommendation

1) effect on this Recommendation: This Recommendation needs to describe a solution that allows the presence and the absence of the feature;
2) effect on products: an implementation may or may not support it; however, support is recommended;
3) effect on deployments: a deployment may or may not use it; however, usage is recommended.

May/need not:

Permission/Option

1) effect on this Recommendation: This Recommendation needs to describe a solution that allows the presence and the absence of the required feature;
2) effect on products: an implementation may or may not support it;
3) effect on deployments: A deployment may or may not use it.

6 Mapping of basic data types

[BBF TR-106] specifies the object structure supported by [BBF TR-069]-enabled devices and specifies the structural requirements for the data hierarchy. This clause includes the mapping attribute data types to [BBF TR-181] parameters that follow the conventions of section 3 of [BBF TR-106] and data types described in Table 4 of [BBF TR-106]. See Table 6-1.

Table 6-1 – Data type mapping

oneM2M data types	Mapping to data types in [BBF TR-106]	Conversion notes
xs:boolean	boolean	
xs:string	string	Mapping is constrained to the size of the string
xs:unsignedInt	unsignedInt	
xs:unsignedLong	unsignedLong	
xs:integer	long	Mapping is constrained to the size of the long data type
Xs:positiveInteger	unsignedLong	Mapping is constrained to a lower limit of 1 and the size of the unsignedLong data type
Xs:nonNegativeInteger	unsignedLong	Mapping is constrained the size of the unsignedLong data type
Comma-separated Lists	Comma-separated Lists	Data structure is represented by comma-separated list as described in section 3.2.3 of [BBF TR-106]

In some instances, the conversion of the contents between data types will cause an error to occur (e.g., xs:integer is too long). When an error occurs in the conversion of a data type, the response status code is STATUS_BAD_REQUEST.

7 Mapping of identifiers

7.0 Introduction

[BBF TR-069] specifies three types of device, known as customer premises equipment (CPE), that are capable of being managed from the perspective of the BBF TR-069 agent as follows.

- CPE that hosts the BBF TR-069 agent: section A.3.3.1 of [BBF TR-069] defines the required fields for a CPE to be identified. These fields include the organizationally unique identifier (OUI) and serial number (SN) of the CPE assigned by the CPE manufacturer. The manufacturer may assign a product class (PC) to the CPE. The format of the identifier is as follows: OUI-[PC-]SN.
- Virtual device: This type of device is addressed as a CPE. The virtual device has its own OUI-[PC-]SN as represented by the CPE proxier. The CPE proxier emulates a common premises equipment wide area network management protocol (CWMP) agent for each virtual device.
- Embedded device: This type of device is addressed as one or more objects within the data model of the CPE that hosts the BBF TR-069 agent.

7.1 Mapping of device identifiers to the node resource

Node resources are identified for each instance of an application-dedicated node (ADN), application service node (ASN) and middle node (MN), and are identified using the M2M node identifier (M2M-Node-ID) specified in oneM2M TS-0001 [ITU-T Y.4500.1].

CPE device identifiers shall map to the nodeID attribute of the <node> resource. CPE device identifiers are obtained from the contents of the following attributes:

- Device.DeviceInfo.ManufacturerOUI;
- Device.DeviceInfo.ProductClass;
- Device.DeviceInfo.SerialNumber.

Virtual device identifiers shall map to the nodeID attribute of the <node> resource. The virtual device identifiers are obtained from the CPE proxier using the contents of the attributes:

- Device.ManagementServer.VirtualDevice.{i}.ManufacturerOUI;
- Device.ManagementServer.VirtualDevice.{i}.ProductClass;
- Device.ManagementServer.VirtualDevice.{i}.SerialNumber.

Embedded device identifiers shall map to the nodeID attribute of the <node> resource. Embedded device identifiers are obtained using the containing CPE device or virtual device identifiers along with the contents of the attributes of the:

- Device.ManagementServer.EmbeddedDevice.{i}.ControllerID;
- Device.ManagementServer.EmbeddedDevice.{i}.ProxiedDeviceID.

7.2 Identifier of an object instance

The BBF TR-069 specification permits objects to have multiple object instances where each object instance is contained within the objectPath attribute of the resource within the context of the resource's objectId as defined in clause 1 of [BBF TR-069].

In order to allow the application entity (AE) or CSE originating the request that manipulates a resource to easily align the M2M service layer with the resource's external technology identifier, the value of the object instance "{i}" should be a part of the identifier of the resource in the M2M service layer where possible. For example, if the [areaNetwork] resource has an object instance identifier of "Device.X_oneM2M_org_CSE.1.M2MareaNetworkDevice.[foo]", then the M2M service layer resource should be identified using the object instance of the underlying technology (e.g., "/foo" for the resource areaNetwork).

8 Mapping of resources

8.0 Introduction

This clause contains all information on how to map management resources from oneM2M TS-0004 [ITU-T Y.4500.4] to managed objects and parameters as defined in the [BBF TR-181] data model or the remote procedure calls (RPCs) in [BBF TR-069].

8.1 General mapping assumptions

8.1.0 Introduction

[BBF TR-069] specifies a protocol for communication between CPE and an auto-configuration server (ACS). Any BBF TR-069-enabled device has to follow the data model as described in [BBF TR-106] and [BBF TR-181] as well as RPCs described in [BBF TR-069].

As [BBF TR-181] is the model on which the resources are mapped, all resources shall have the objects of the BBF TR-181 namespace (e.g., "urn:broadband-forum-org:tr-181-2-7-0").

8.1.1 Mapping of device identifiers

The device identifiers for CPEs are mapped to the resource types [deviceInfo].

CPE and virtual devices map their Device Identifiers (OUI-[PC-]SN) to the manufacturer, deviceType and deviceLabel attributes of the resource [deviceInfo].

For embedded devices, the ControllerID and ProxiedDeviceID parameters of the Device.ManagementServer.EmbeddedDevice.{i} object instance are mapped to the deviceLabel attribute of the Resource [deviceInfo] as a comma-separated list: "Device.ManagementServer.EmbeddedDevice.{i}.ControllerID, Device.ManagementServer.EmbeddedDevice.{i}.ProxiedDeviceID".

8.1.2 Mapping of embedded devices

The BBF TR-181 [BBF TR-181] specification does not provide a mechanism where embedded devices provide information related to the Device.DeviceInfo objects and sub-objects. Instead, [BBF TR-181] provides this information in a manner that is reliant on the underlying technology (e.g., ZigBee®, UpnP) of the embedded device.

As such the mapping of the [memory] and [battery] resources are implementation specific for each underlying technology and is outside the scope of this Recommendation.

8.2 Resource [deviceInfo]

The resource [deviceInfo] is a read-only resource that shall map to the Device.DeviceInfo object of [BBF TR-181] for CPE and virtual devices. See Tables 8.2-1 and 8.2-2.

The information shall be retrieved using the GetParameterValues RPC of [BBF TR-069].

NOTE – The SerialNumber, ModelNumber, ProductClass attributes for a virtual device are the same values as the Device.ManagementServer.VirtualDevice.{i} object in the CPE proxier.

Table 8.2-1 – Resource [deviceInfo] for CPE and virtual devices

Attribute name of [deviceInfo]	BBF TR-181 parameter
deviceLabel	Device.DeviceInfo.SerialNumber
manufacturer	Device.DeviceInfo.Manufacturer
model	Device.DeviceInfo.ModelNumber
deviceType	Device.DeviceInfo.ProductClass
fwVersion	Device.DeviceInfo.SoftwareVersion if the device supports only one software version. If the device supports multiple software versions this shall map to Device.DeviceInfo.AdditionalSoftwareVersion
swVersion	Device.DeviceInfo.SoftwareVersion
hwVersion	Device.DeviceInfo.HardwareVersion

Table 8.2-2 – Resource [deviceInfo] for embedded devices

Attribute name of [deviceInfo]	BBF TR-181 parameter
deviceLabel	Comma-separated list: "Device.ManagementServer.EmbeddedDevice.{i}.ControllerID, Device.ManagementServer.EmbeddedDevice.{i}.ProxiedDeviceID
manufacturer	No mapping available
model	No mapping available
deviceType	No mapping available
fwVersion	No mapping available
swVersion	No mapping available
hwVersion	No mapping available

8.3 Resource [memory]

The resource [memory] is a read-only resource that shall map to the Device.DeviceInfo.MemoryStatus object of [BBF TR-181] for CPE and virtual devices. See Table 8.3-1.

The information shall be retrieved using the GetParameterValues RPC of [BBF TR-069].

Attempts to modify the attributes of the memory Resource causes an error code "operation unsupported" to be returned.

Table 8.3-1 – Resource [memory]

Attribute name of [memory]	BBF TR-181 parameter
memAvailable	Device.DeviceInfo.MemoryStatus.Free
memTotal	Device.DeviceInfo.MemoryStatus.Total

8.4 Resource [battery]

The resource [battery] is a read-only resource that shall map to an instance of the Device.DeviceInfo.X_oneM2M_org_BatteryStatus.Battery.{i} object for CPE and virtual devices. See Table 8.4-1.

The information shall be retrieved using the GetParameterValues RPC of [BBF TR-069].

Table 8.4-1 – Resource [battery]

Attribute name of [battery]	BBF TR-181 parameter
batteryLevel	Device.DeviceInfo.X_oneM2M_org_BatteryStatus.Battery.{i}.Level
batteryStatus	Device.DeviceInfo.X_oneM2M_org_BatteryStatus.Battery.{i}.Status

8.5 Resource [areaNwkInfo]

The resource [areaNwkInfo] is a multi-instance resource where each instance of the resource shall map to an instance of the Device.X_oneM2M_org_CSE.{i}.M2MareaNetwork.{i} object. See Table 8.5-1.

As the resource [areaNwkInfo] is a multi-instance resource, the M2MareaNetwork object is a multi-object instance that can be created and deleted.

The M2MareaNetwork instance shall be created using the Add Object RPC of [BBF TR-069].

The M2MareaNetwork instance shall be deleted using the Delete Object RPC of [BBF TR-069].

The information of an M2MareaNetwork shall be retrieved using the GetParameterValues RPC of [BBF TR-069].

The information of an M2MareaNetwork shall be modified using the SetParameterValues RPC of [BBF TR-069].

Table 8.5-1 – Resource [areaNwkInfo]

Attribute name of [areaNwkInfo]	X_oneM2M_org parameter
areaNwkType	Device.X_oneM2M_org_CSE.{i}.M2MareaNetwork.{i}.Type
listOfDevices	Device.X_oneM2M_org_CSE.{i}.M2MareaNetwork.{i}.ListOfDevices

8.6 Resource [areaNwkDeviceInfo]

The resource [areaNwkDeviceInfo] is a multi-instance resource where each instance of the resource shall map to an instance of the Device.X_oneM2M_org_CSE.{i}.AreaNetworkDevice.{i} object. See Table 8.6-1.

As the resource [areaNwkDeviceInfo] is a multi-instance resource, the AreaNetworkDevice object is a multi-object instance that can be created and deleted.

Instances of the resource [areaNwkDeviceInfo] are referenced in the listOfDevices attribute of the associated resource [areaNwkInfo].

The M2MareaNetworkDevice instance shall be created using the Add Object RPC of [BBF TR-069].

The M2MareaNetworkDevice instance shall be deleted using the Delete Object RPC of [BBF TR-069].

The information of an M2MareaNetworkDevice shall be retrieved using the GetParameterValues RPC of [BBF TR-069].

The information of an M2MareaNetworkDevice shall be modified using the SetParameterValues RPC of [BBF TR-069].

Table 8.6-1 – Resource [areaNwkDeviceInfo]

Attribute name of [areaNwkDeviceInfo]	X_oneM2M_org parameter
devId	Device.X_oneM2M_org_CSE.{i}.M2MareaNetworkDevice.{i}.Host
devType	Device.X_oneM2M_org_CSE.{i}.M2MareaNetworkDevice.{i}.Type
areaNwkId	Reference to Device.X_oneM2M_org_CSE.{i}.M2MareaNetworkDevice.{i}.M2MareaNetwork
sleepInterval	Device.X_oneM2M_org_CSE.{i}.M2MareaNetworkDevice.{i}.SleepInterval
sleepDuration	Device.X_oneM2M_org_CSE.{i}.M2MareaNetworkDevice.{i}.SleepDuration
status	Device.X_oneM2M_org_CSE.{i}.M2MareaNetworkDevice.{i}.Status
listOfNeighbors	Device.X_oneM2M_org_CSE.{i}.M2MareaNetworkDevice.{i}.Neighbors

8.7 Resource [eventLog]

The resource [eventLog] is a multi-instance resource where each instance of the resource shall map to an instance of the Device.DeviceInfo.X_oneM2M_org_Diagnostics.EventLog.{i} object. See Table 8.7-1.

The EventLog instance shall be created using the Add Object RPC of [BBF TR-069].

The EventLog instance shall be deleted using the Delete Object RPC of [BBF TR-069].

The information of an EventLog instance shall be retrieved using the GetParameterValues RPC of [BBF TR-069].

The information of an EventLog instance shall be updated using the SetParameterValues RPC of [BBF TR-069].

Table 8.7-1 – Resource [eventLog]

Attribute name of [eventLog]	BBF TR-181 parameter
logTypeId	Device.DeviceInfo.X_oneM2M_org_Diagnostics.EventLog.{i}.Type
logData	Device.DeviceInfo.X_oneM2M_org_Diagnostics.EventLog.{i}.Data
logStatus	Device.DeviceInfo.X_oneM2M_org_Diagnostics.EventLog.{i}.Status
logStart	Set to "True" , the Device.DeviceInfo.X_oneM2M_org_Diagnostics.EventLog.{i}.Enable parameter is set to "True".
logStop	Set to "True" , the Device.DeviceInfo.X_oneM2M_org_Diagnostics.EventLog.{i}.Enable parameter is set to "False".

8.8 Resource [deviceCapability]

The resource [deviceCapability] represents a capability of device that can be administratively enabled or disabled. The lists of capabilities that are managed are defined in the enumeration of the capabilityName attribute. The BBF TR-181 data model defines a subset of capabilities listed in the deviceCapability enumeration. The supported device capabilities within [BBF TR-181] include:

- local area network (LAN) interfaces: universal serial bus (USB), wireless fidelity (Wi-Fi), HomePlug, multimedia over coax alliance (MoCA), Universal Powerline Association (UPA);
- hardware capabilities: SmartCardReader.

See Table 8.8-1.

The information shall be retrieved using the GetParameterValues RPC of [BBF TR-069].

The capabilities shall be enabled and disabled using the SetParameterValues RPC of [BBF TR-069].

Table 8.8-1 – Resource [capabilityInstance]

Attribute name of [capabilityInstance]	BBF TR-181 parameter
capabilityName	This attribute is fixed based on the value of the capabilityName attribute
attached	Returns "True"
capabilityActionStatus	Status is defined as: • Success if the SetParameterValues RPC indicates that the operation was successful • Failure if the response to the SetParameterValues RPCs indicates that the operation failed • In process if the SetParameterValues RPC is initiated but the response to the SetParameterValues RPC has not been received
currentState	USB: Device.USB.Interface.{i}.Enable Wi-Fi: Device.Wi-Fi.Radio.{i}.Enable HomePlug: Device.HomePlug.Interface.{i}.Enable MoCA: Device.MoCA.Interface.{i}.Enable UPA: Device.UPA.Interface.{i}.Enable SmartCardReader: Device.SmartCardReaders.SmartCardReader.{i}.Enable
enable	USB: Device.USB.Interface.{i}.Enable Wi-Fi: Device.Wi-Fi.Radio.{i}.Enable HomePlug: Device.HomePlug.Interface.{i}.Enable MoCA: Device.MoCA.Interface.{i}.Enable UPA: Device.UPA.Interface.{i}.Enable SmartCardReader: Device.SmartCardReaders.SmartCardReader.{i}.Enable
disable	Same parameter is used to disable a capability as the enable attribute

8.9 Resource [firmware]

The resource [firmware] represents a firmware instance and is not considered a BBF TR-069-managed entity within the device until the firmware resource's update attribute has been assigned a value of "True". When this occurs, the BBF TR-069 Download RPC shall be invoked. See Table 8.9-1.

NOTE – In many instances, the server from which the firmware is downloaded requires authentication in the form of username and password credentials. The CSE that executes firmware download shall maintain the mapping of the username and password of the download server needed to download the firmware outside the lifecycle of the specific firmware.

Table 8.9-1 – Resource [firmware]

Attribute name of [firmware]	RPC download arguments
Uniform resource locator (URL)	URL
update	When set to the value of "True" executes the Download operations with a FileType "1 Firmware Upgrade Image" is performed
	Username: Received from the CSE for the download server where the update is set to "True"
	Password: Received from the CSE for the download server where the update is set to "True"
	CommandKey: Automatically set by the CSE where the update is set to "True" in order to correlate the TransferComplete response
	FileSize: 0 (not used)
	TargetFileName: <empty> (not used)
	DelaySeconds: 0 (immediate)
	SuccessURL: <empty> (not used)
	FailureURL: <empty> (not used)

8.10 Resource [software]

The resource [software] is a multi-instance resource where each instance of the resource maps directly to an instance of Device.SoftwareModules.DeploymentUnit.{i} object for the deployment aspects (install, uninstall) of the resource [software]. The install and uninstall operation of the resource [software] is performed using a combination of the ChangeDUState and ChangeDUStateComplete RPCs. See Tables 8.10-1, 8.10-2 and 8.10-3.

Once a resource [software] has been installed, the resource shall be mapped to the associated Device.SoftwareModules.ExecutionUnit.{i} objects in order to activate and deactivate the associated execution unit.

The resource [software] version and name shall be retrieved using the GetParameterValues RPC of [BBF TR-069].

The activate and deactivate operations of the resource [software] shall be performed by manipulating the Device.SoftwareModules.ExecutionUnit.{i}.RequestedState parameter using the SetParameterValues RPC.

NOTE – The resource [software] provides support for only one execution unit per deployment unit (DU). If a DU is discovered by the M2M service layer that contains multiple execution units for a DU; only one execution unit is exposed. The selection of which execution unit is implementation specific.

Table 8.10-1 – Resource [software]

Attribute name of [software]	Description
version	Device.SoftwareModules.DeploymentUnit.{i}.Version
name	Device.SoftwareModules.DeploymentUnit.{i}.Name
URL	Device.SoftwareModules.DeploymentUnit.{i}.URL
install	Use the ChangeDUState:InstallOpStruct
installStatus	Status is defined as: • Success if the ChangeDUStateComplete RPC indicates that the operation was successful • Failure if the response to the ChangeDUState or ChangeDUStateComplete RPCs indicates that the operation failed • In process if the ChangeDUState RPC is initiated but the ChangeDUStateComplete RPC has not been received
activate	The action that activates software previously installed
deactivate	The action that deactivates software
activeStatus	Status is defined as: • Success if the SetParameterValues RPC indicates that the operation was successful • Failure if the response to the SetParameterValues RPCs indicates that the operation failed • In process if the SetParameterValues RPC is initiated but the response to the SetParameterValues RPC has not been received

Table 8.10-2 – RPC ChangeDUState:InstallOpStruct Arguments

RPC ChangeDUState:InstallOpStruct Argument
URL: URL of the Server that M2M Node uses to download the DU
Username: Username credential of Server that the CPE uses to download the DU - Supplied by the CSE
Password: Password credential of Server that the CPE uses to download the DU - Supplied by the CSE
Universal unique identifier (UUID): Supplied by the CSE and used to correlate the DU for the uninstall operation
ExecurtionEnvRef: <empty> not used

Table 8.10-3 – RPC ChangeDUState:UninstallOpStruct Arguments

RPC ChangeDUState:Uninstall OpStruct Argument
UUID: UUID of the DU that was installed - Maintained by the CSE
ExecutionEnvRef: <empty> not used

8.11 Resource [reboot]

The resource [reboot] maps to either the Reboot RPC or FactoryReset RPC of [BBF TR-069]. See Table 8.11-1.

When the reboot attribute of the resource [reboot] is set to "True", the CSE shall execute the Reboot RPC of [BBF TR-069].

When the factory reset attribute of resource [reboot] is set to "True", the CSE shall execute the FactoryReset RPC of [BBF TR-069].

See Table 8.11-2.

Table 8.11-1 – Resource [reboot]

Attribute name of [reboot]	Description
reboot	Executes the Reboot RPC
factoryReset	FactoryReset RPC

Table 8.11-2 – RPC reboot arguments

RPC reboot arguments
CommandKey: Automatically set by the CSE where the reboot is set to "True" in order to correlate the "M-Reboot" Event from the next Inform

8.12 Resource [cmdhPolicy]

8.12.0 Introduction

The resource [cmdhPolicy] represents a set of rules defining which communication management and delivery handling (CMDH) parameters will be used by default when a request issued by a local originator contains the **ec** (event category) parameter but not all other CMDH parameters, see clause D.12 of oneM2M TS-0001 [ITU-T Y.4500.1]. See Table 8.12-1.

The resource [cmdhPolicy] is a multi-instance resource where each instance of the resource shall map to an instance of the Device.X_oneM2M_org_CSE.{i}.CMDH.Policy.{i} object.

The Policy instance shall be created using the Add Object RPC of [BBF TR-069].

The Policy instance shall be deleted using the Delete Object RPC of [BBF TR-069].

The information of a Policy instance shall be retrieved using the GetParameterValues RPC of [BBF TR-069].

The information of a Policy instance shall be updated using the SetParameterValues RPC of [BBF TR-069].

Table 8.12-1 – Resource [cmdhPolicy]

Attribute name of [cmdhPolicy]	X_oneM2M_org parameter
name	Device.X_oneM2M_org_CSE.{i}.CMDH.Policy.{i}.Name
cmdhDefaults	Device.X_oneM2M_org_CSE.{i}.CMDH.Policy.{i}.DefaultRule
cmdhLimits	Device.X_oneM2M_org_CSE.{i}.CMDH.Policy.{i}.LimitRules
cmdhNetworkAccessRules	Device.X_oneM2M_org_CSE.{i}.CMDH.Policy.{i}.NetworkAccessECRules
cmdhBuffer	Device.X_oneM2M_org_CSE.{i}.CMDH.Policy.{i}.BufferRules

8.12.1 Resource [activeCmdhPolicy]

The resource [activeCmdhPolicy] provides a link to the currently active set of CMDH policies, see clause D.12.1 of oneM2M TS-0001 [ITU-T Y.4500.1]. See Table 8.12.1-1.

The resource [activeCmdhPolicy] is mapped to the Enable parameter of the Device.X_oneM2M_org_CSE.{i}.CMDH.Policy.{i} object.

The information of a Policy instance shall be updated using the SetParameterValues RPC of [BBF TR-069].

Table 8.12.1-1 – Resource [activeCmdhPolicy]

Attribute name of [activeCmdhPolicy]	X_oneM2M_org parameter
cmdhPolicy	Device.X_oneM2M_org_CSE.{i}.CMDH.Policy.{i}.Enable At most one Policy instance shall be enabled at a time. As such the Policy instance that has the Enable parameter with a value of "True" is the active CMDH policy

8.12.2 Resource [cmdhDefaults]

The resource [cmdhDefaults] defines default CMDH policy values, see clause D.12.2 of oneM2M TS-0001 [ITU-T Y.4500.1]. See Table 8.12.2-1.

The resource [cmdhDefaults] is a multi-instance resource where each instance of the resource shall map to an instance of the Device.X_oneM2M_org_CSE.{i}.CMDH.Default.{i} object.

The Default instance shall be created using the Add Object RPC of [BBF TR-069].

The Default instance shall be deleted using the Delete Object RPC of [BBF TR-069].

The information of a Default instance shall be retrieved using the GetParameterValues RPC of [BBF TR-069].

The information of a Default instance shall be updated using the SetParameterValues RPC of [BBF TR-069].

Table 8.12.2-1 – Resource [cmdhDefaults]

Attribute name of [cmdhDefaults]	X_oneM2M_org parameter
cmdhDefEcValue	Device.X_oneM2M_org_CSE.{i}.CMDH.Default.{i}.DefaultECRules
cmdhEcDefParamValues	Device.X_oneM2M_org_CSE.{i}.CMDH.Default.{i}.DefaultECParamRules

8.12.3 Resource [cmdhDefEcValue]

The resource [cmdhDefEcValue] represents a value for the **ec** (event category) parameter of an incoming request, see clause D.12.3 of oneM2M TS-0001 [ITU-T Y.4500.1]. See Table 8.12.3-1.The resource [cmdhDefEcValue] is a multi-instance resource where each instance of the resource shall map to an instance of the Device.X_oneM2M_org_CSE.{i}.CMDH.DefaultECRule.{i} object.

The DefaultECRule instance shall be created using the Add Object RPC of [BBF TR-069].

The DefaultECRule instance shall be deleted using the Delete Object RPC of [BBF TR-069].

The information of a DefaultECRule instance shall be retrieved using the GetParameterValues RPC of [BBF TR-069].

The information of a DefaultECRule instance shall be updated using the SetParameterValues RPC of [BBF TR-069].

Table 8.12.3-1 – Resource [cmdhDefEcValue]

Attribute name of [cmdhDefEcValue]	X_oneM2M_org parameter
order	Device.X_oneM2M_org_CSE.{i}.CMDH.DefaultECRule.{i}.Order
defEcValue	Device.X_oneM2M_org_CSE.{i}.CMDH.DefaultECRule.{i}.EventCategory
requestOrigin	Device.X_oneM2M_org_CSE.{i}.CMDH.DefaultECRule.{i}.RequestOrigin
requestContext	Device.X_oneM2M_org_CSE.{i}.CMDH.DefaultECRule.{i}.RequestContext
requestContextNotification	Device.X_oneM2M_org_CSE.{i}.CMDH.DefaultECRule.{i}.RequestContext NotificationEnable
requestCharacteristics	Device.X_oneM2M_org_CSE.{i}.CMDH.DefaultECRule.{i}.RequestCharact eristics

8.12.4 Resource [cmdhEcDefParamValues]

The resource [cmdhEcDefParamValues] represents a specific set of default values for the CMDH related parameters **rqet** (request expiration timestamp), **rset** (result expiration timestamp), **oet** (operational execution time), **rp** (response persistence) and **da** (delivery aggregation) that are applicable for a given **ec** (event category) if these parameters are not specified in the request, see clause D.12.4 of oneM2M TS-0001 [ITU-T Y.4500.1]. See Table 8.12.4-1.

The resource [cmdhEcDefParamValues] is a multi-instance resource where each instance of the resource shall map to an instance of the Device.X_oneM2M_org_CSE.{i}.CMDH.DefaultECParamRule.{i} object.

The DefaultECParamRule instance shall be created using the Add Object RPC of [BBF TR-069].

The DefaultECParamRule instance shall be deleted using the Delete Object RPC of [BBF TR-069].

The information of a DefaultECParamRule instance shall be retrieved using the GetParameterValues RPC of [BBF TR-069].

The information of a DefaultECParamRule instance shall be updated using the SetParameterValues RPC of [BBF TR-069].

Table 8.12.4-1 – Resource [cmdhEcDefParamValues]

Attribute name of [cmdhEcDefParamValues]	X_oneM2M_org parameter
applicableEventCategory	Device.X_oneM2M_org_CSE.{i}.CMDH.DefaultECParamRule.{i}.EventCategories
defaultRequestExpTime	Device.X_oneM2M_org_CSE.{i}.CMDH.DefaultECParamRule.{i}.RequestExpTime
defaultResultExpTime	Device.X_oneM2M_org_CSE.{i}.CMDH.DefaultECParamRule.{i}.ResultExpTime
defaultOpExecTime	Device.X_oneM2M_org_CSE.{i}.CMDH.DefaultECParamRule.{i}.OperationExecTime
defaultRespPersistence	Device.X_oneM2M_org_CSE.{i}.CMDH.DefaultECParamRule.{i}.ResponsePersistence
defaultDelAggregation	Device.X_oneM2M_org_CSE.{i}.CMDH.DefaultECParamRule.{i}.DeliveryAggregation

8.12.5 Resource [cmdhLimits]

The resource [cmdhLimits] represents limits for CMDH related parameter values, see clause D.12.5 of oneM2M TS-0001 [ITU-T Y.4500.1]. See Table 8.12.5-1.

The resource [cmdhLimits] is a multi-instance resource where each instance of the resource shall map to an instance of the Device.X_oneM2M_org_CSE.{i}.CMDH.Limit.{i} object.

The Limit instance shall be created using the Add Object RPC of [BBF TR-069].

The Limit instance shall be deleted using the Delete Object RPC of [BBF TR-069].

The information of a Limit instance shall be retrieved using the GetParameterValues RPC of [BBF TR-069].

The information of a Limit instance shall be updated using the SetParameterValues RPC of [BBF TR-069].

Table 8.12.5-1 – Resource [cmdhLimits]

Attribute name of [cmdhLimits]	X_oneM2M_org parameter
order	Device.X_oneM2M_org_CSE.{i}.CMDH.Limit.{i}.Order
requestOrigin	Device.X_oneM2M_org_CSE.{i}.CMDH.Limit.{i}.RequestOrigin
requestContext	Device.X_oneM2M_org_CSE.{i}.CMDH.Limit.{i}.RequestContext
requestContextNotification	Device.X_oneM2M_org_CSE.{i}.CMDH.Limit.{i}.RequestContextNotificationEnable
requestCharacteristics	Device.X_oneM2M_org_CSE.{i}.CMDH.Limit.{i}.RequestCharacteristics
limitsEventCategory	Device.X_oneM2M_org_CSE.{i}.CMDH.Limit.{i}.EventCategories
limitsRequestExpTime	Device.X_oneM2M_org_CSE.{i}.CMDH.Limit.{i}.RequestExpTime
limitsResultExpTime	Device.X_oneM2M_org_CSE.{i}.CMDH.Limit.{i}.ResultExpTime
limitsOpExecTime	Device.X_oneM2M_org_CSE.{i}.CMDH.Limit.{i}.OperationExecTime
limitsRespPersistence	Device.X_oneM2M_org_CSE.{i}.CMDH.Limit.{i}.ResponsePersistence
limitsDelAggregation	Device.X_oneM2M_org_CSE.{i}.CMDH.Limit.{i}.DeliveryAggregation

8.12.6 Resource [cmdhNetworkAccessRules]

The resource [cmdhNetworkAccessRules] defines the usage of underlying networks for forwarding information to other CSEs during processing of CMDH-related requests in a CSE, see clause D.12.6 of oneM2M TS-0001 [ITU-T Y.4500.1]. See Table 8.12.6-1.

The resource [cmdhNetworkAccessRules] is a multi-instance resource where each instance of the resource shall map to an instance of the Device.X_oneM2M_org_CSE.{i}.CMDH.NetworkAccessECRule.{i} object.

The NetworkAccessECRule instance shall be created using the Add Object RPC of [BBF TR-069].

The NetworkAccessECRule instance shall be deleted using the Delete Object RPC of [BBF TR-069].

The information of a NetworkAccessECRule instance shall be retrieved using the GetParameterValues RPC of [BBF TR-069].

The information of a NetworkAccessECRule instance shall be updated using the SetParameterValues RPC of [BBF TR-069].

Table 8.12.6-1 – Resource [cmdhNetworkAccessRules]

Attribute name of [cmdhNetworkAccessRules]	X_oneM2M_org parameter
applicableEventCategories	Device.X_oneM2M_org_CSE.{i}.CMDH.NetworkAccessECRule.{i}.EventCategories
cmdhNwAccessRule	Device.X_oneM2M_org_CSE.{i}.CMDH.NetworkAccessECRule.{i}.NetworkAccessRules

8.12.7 Resource [cmdhNwAccessRule]

The resource [cmdhNwAccessRule] defines limits in usage of specific underlying networks for forwarding information to other CSEs during processing of CMDH-related requests, see clause D.12.7 of oneM2M TS-0001 [ITU-T Y.4500.1]. See Table 8.12.7-1

The resource [cmdhNwAccessRule] is a multi-instance resource where each instance of the resource shall map to an instance of the Device.X_oneM2M_org_CSE.{i}.CMDH.NetworkAccessECRule.{i} object.

The NetworkAccessRule instance shall be created using the Add Object RPC of [BBF TR-069].

The NetworkAccessRule instance shall be deleted using the Delete Object RPC of [BBF TR-069].

The information of a NetworkAccessRule instance shall be retrieved using the GetParameterValues RPC of [BBF TR-069].

The information of a NetworkAccessRule instance shall be updated using the SetParameterValues RPC of [BBF TR-069].

Table 8.12.7-1 – Resource [cmdhNwAccessRule]

Attribute name of [cmdhNwAccessRule]	X_oneM2M_org parameter
targetNetwork	Device.X_oneM2M_org_CSE.{i}.CMDH.NetworkAccessRule.{i}.TargetNetworks
minReqVolume	Device.X_oneM2M_org_CSE.{i}.CMDH.NetworkAccessRule.{i}.MinimumReqVolume
backOffParameters	Device.X_oneM2M_org_CSE.{i}.CMDH.NetworkAccessRule.{i}.BackoffTime
	Device.X_oneM2M_org_CSE.{i}.CMDH.NetworkAccessRule.{i}.BackoffTimeIncrement
	Device.X_oneM2M_org_CSE.{i}.CMDH.NetworkAccessRule.{i}.MaximumBackoffTime
otherConditions	Device.X_oneM2M_org_CSE.{i}.CMDH.NetworkAccessRule.{i}.OtherConditions
allowedSchedule	Device.X_oneM2M_org_CSE.{i}.CMDH.NetworkAccessRule.{i}.AllowedSchedule

8.12.8 Resource [cmdhBuffer]

The resource [cmdhBuffer] represents limits in usage of buffers for temporarily storing information that needs to be forwarded to other CSEs during processing of CMDH-related requests in a CSE, see clause D.12.8 of oneM2M TS-0001 [ITU-T Y.4500.1]. See Table 8.12.8-1.

The resource [cmdhBuffer] is a multi-instance resource where each instance of the resource shall map to an instance of the Device.X_oneM2M_org_CSE.{i}.CMDH.Buffer.{i} object.

The Buffer instance shall be created using the Add Object RPC of [BBF TR-069].

The Buffer instance shall be deleted using the Delete Object RPC of [BBF TR-069].

The information of a Buffer instance shall be retrieved using the GetParameterValues RPC of [BBF TR-069].

The information of a Buffer instance shall be updated using the SetParameterValues RPC of [BBF TR-069].

Table 8.12.8-1 – Resource [cmdhBuffer]

Attribute name of [cmdhBuffer]	X_oneM2M_org parameter
applicableEventCategory	Device.X_oneM2M_org_CSE.{i}.CMDH.Buffer.{i}.EventCategories
maxBufferSize	Device.X_oneM2M_org_CSE.{i}.CMDH.Buffer.{i}.MaximumBufferSize
storagePriority	Device.X_oneM2M_org_CSE.{i}.CMDH.Buffer.{i}.StoragePriority

8.13 Resource type <mgmtCmd>

Each mgmtCmd Resource shall map to [BBF TR-069] RPC commands based on the value of cmdType. Accordingly, execReqArgs shall contain arguments related to the corresponding BBF TR-069 RPCs. See Table 8.13-1. Details of the corresponding procedure mapping are described in clause 9.2.

Table 8.13-1 – Resource type <mgmtCmd>

Attribute cmdType of mgmtCmd	Attribute execReqArgs of mgmtCmd
cmdType = RESET	Shall include all arguments related to BBF FactoryReset RPC
cmdType = REBOOT	Shall include all arguments related to BBF Reboot RPC
cmdType = UPLOAD	Shall include all arguments related to BBF Reboot RPC
cmdType = DOWNLOAD	Shall contain all arguments related to BBF Reboot RPC
cmdType = SOFTWAREINSTALL	Shall contain all arguments related to BBF ChangeDUState RPC which shall contain "InstallOpStruct" structure
cmdType = SOFTWAREUNINSTALL	Shall contain all arguments related to BBF ChangeDUState RPC which shall contain "UninstallOpStruct" structure

8.14 Resource type <execInstance>

The <execInstance> resource from oneM2M TS-0004 [ITU-T Y.4500.4] shall map to BBF CancelTransfer RPC commands when it is disabled or cancelled using an update operation or deleted using a delete operation. The details are described in clause 9.2.

9 Mapping of procedures for management

9.0 Introduction

This clause contains all information on how to map management resource primitives from oneM2M TS-0004 [ITU-T Y.4500.4] to the RPCs in [BBF TR-069].

9.1 Resource type <mgmtObj> primitive mappings

9.1.0 Introduction

This clause contains all information on how to map resource type <mgmtObj> primitives from oneM2M TS-0004 [ITU-T Y.4500.4] to the RPCs in [BBF TR-069].

9.1.1 Alias-based addressing mechanism

In order to utilize the alias-based addressing mechanism, the mechanism has to be supported by the ACS and CPE in order to map the M2M service layer identifier for the resource instance to the CPE object instance. If the alias-based addressing mechanism feature is not supported by either the ACS or CPE, the CSE has to retain the mapping of these M2M Resource instance identifiers.

9.1.2 Create primitive mapping

9.1.2.0 Introduction

The Create Request and Response primitives shall map to the AddObject RPC. The AddObject RPC is defined in [BBF TR-069] as a synchronous RPC and returns a successful response or one of the fault codes listed in Table 9.1.2.0-1.

Table 9.1.2.0-1 – AddObject fault code mapping

Fault code	Description	Response Status Code
9001	Request denied (no reason specified)	STATUS_BAD_REQUEST
9002	Internal error	STATUS_BAD_REQUEST
9003	Invalid arguments	STATUS_BAD_REQUEST
9004	Resources exceeded (when used in association with SetParameterValues, this cannot be used to indicate Parameters in error)	STATUS_BAD_REQUEST
9005	Invalid Parameter name (associated with Set/GetParameterValues, GetParameterNames, Set/GetParameterAttributes, AddObject, and DeleteObject)	STATUS_NOT_IMPLEMENTED

9.1.2.1 M2M service layer resource instance identifier mapping

When the resource is a multi-instance resource, the AddObject RPC should utilize the alias-based addressing mechanism as defined in section 3.6.1 of [BBF TR-069] in order to use the resource instance value of the uniform resource identifier (URI).

9.1.3 Delete primitive mapping

9.1.3.1 Delete primitive mapping for deletion of object instances

The Delete Request and Response primitives that result in the deletion of a resource shall map to the DeleteObject RPC. The DeleteObject RPC is defined in [BBF TR-069] as a synchronous RPC and returns a successful response or one of the fault codes listed in Table 9.1.3.1-1.

Table 9.1.3.1-1 – DeleteObject fault code mapping

Fault code	Description	Response status code
9001	Request denied (no reason specified)	STATUS_BAD_REQUEST
9002	Internal error	STATUS_BAD_REQUEST
9003	Invalid arguments	STATUS_BAD_REQUEST
9005	Invalid Parameter name (associated with Set/GetParameterValues, GetParameterNames, Set/GetParameterAttributes, AddObject, and DeleteObject)	STATUS_NOT_IMPLEMENTED

9.1.3.2 Delete primitive mapping for software un-install operation

The Delete Request and Response primitives that result in a software un-install operation (e.g., resource [software]) shall use the ChangeDUState mechanism defined in [BBF TR-069]. The ChangeDUState mechanism is an asynchronous command that consists of the synchronous ChangeDUState RPC for the un-installation request and the asynchronous ChangeDUStateComplete RPC. The ChangeDUState RPC returns a successful response or one of the fault codes listed in Table 9.1.3.2-1. A successful response means that the CPE has accepted the ChangeDUState RPC.

Table 9.1.3.2-1 – ChangeDUState Fault Code Mapping

Fault code	Description	Response Status Code
9000	Method not supported	STATUS_BAD_REQUEST
9001	Request denied (no reason specified)	STATUS_BAD_REQUEST
9002	Internal error	STATUS_BAD_REQUEST
9004	Resources exceeded (when used in association with SetParameterValues, this cannot be used to indicate Parameters in error)	STATUS_BAD_REQUEST

Once the CPE has attempted to change the state of the DU, the CPE reports the result of the state change operation using the ChangeDUStateComplete RPC. The ChangeDUStateComplete RPC indicates a successful operation or one of the fault codes listed in Table 9.1.3.2-2.

Table 9.1.3.2-2 – ChangeDUStateComplete fault code mapping

Fault code	Description	Response Status Code
9001	Request denied (no reason specified)	STATUS_BAD_REQUEST
9003	Invalid arguments	STATUS_BAD_REQUEST
9012	File transfer server authentication failure (associated with Upload, Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST
9013	Unsupported protocol for file transfer (associated with Upload, Download, ScheduleDownload, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST
9015	File transfer failure: unable to contact file server (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST
9016	File transfer failure: unable to access file (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST
9017	File transfer failure: unable to complete download (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST
9018	File transfer failure: file corrupted or otherwise unusable (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST
9022	Invalid UUID Format (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install, Update, and Uninstall)	STATUS_BAD_REQUEST
9023	Unknown Execution Environment (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install only)	STATUS_BAD_REQUEST

Table 9.1.3.2-2 – ChangeDUStateComplete fault code mapping

Fault code	Description	Response Status Code
9024	Disabled Execution Environment (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install, Update, and Uninstall)	STATUS_BAD_REQUEST
9025	Deployment Unit to Execution Environment Mismatch (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install and Update)	STATUS_BAD_REQUEST
9026	Duplicate Deployment Unit (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install only)	STATUS_BAD_REQUEST
9027	System Resources Exceeded (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install and Update)	STATUS_BAD_REQUEST
9028	Unknown Deployment Unit (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Update and Uninstall)	STATUS_BAD_REQUEST
9029	Invalid Deployment Unit State (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install, Update and Uninstall)	STATUS_BAD_REQUEST
9030	Invalid Deployment Unit Update - Downgrade not permitted (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Update only)	STATUS_BAD_REQUEST
9031	Invalid Deployment Unit Update - Version not specified (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Update only.	STATUS_BAD_REQUEST
9032	Invalid Deployment Unit Update - Version already exists (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Update only)	STATUS_BAD_REQUEST

9.1.4 Update primitive mapping

9.1.4.1 Update primitive mapping for parameter modifications

The Update Request and Response primitives that modify the value of resource attributes shall map to the SetParameterValues RPC. The SetParametersValue RPC is defined in [BBF TR-069] as a synchronous RPC and returns a successful response or one of the fault codes listed in Table 9.1.4.1-1.

Table 9.1.4.1-1 – SetParameterValues fault code mapping

Fault code	Description	Response Status Code
9001	Request denied (no reason specified)	STATUS_BAD_REQUEST
9002	Internal error	STATUS_BAD_REQUEST
9003	Invalid arguments	STATUS_BAD_REQUEST
9004	Resources exceeded (when used in association with SetParameterValues, this cannot be used to indicate Parameters in error)	STATUS_BAD_REQUEST
9005	Invalid Parameter name (associated with Set/GetParameterValues, GetParameterNames, Set/GetParameterAttributes, AddObject, and DeleteObject)	STATUS_NOT-IMPLEMENTED
9006	Invalid Parameter type (associated with SetParameterValues)	STATUS_BAD_REQUEST
9007	Invalid Parameter value (associated with SetParameterValues)	STATUS_BAD_REQUEST
9008	Attempt to set a non-writable Parameter (associated with SetParameterValues)	STATUS_BAD_REQUEST

9.1.4.2 Update primitive mapping for upload file transfer operations

The Update Request and Response primitives that result in an upload file transfer operation (e.g., logStop attribute of the resource [eventLog]) shall use the upload mechanism defined in [BBF TR-069]. The upload mechanism is an asynchronous command that consists of the synchronous upload RPC for the upload and the asynchronous TransferComplete RPC. The Upload RPC returns a successful response or one of the fault codes listed in Table 9.1.4.2-1. A successful response means that the CPE has accepted the Upload RPC.

Table 9.1.4.2-1 – Upload fault code mapping

Fault code	Description	Response Status Code
9000	Method not supported	STATUS_BAD_REQUEST
9001	Request denied (no reason specified)	STATUS_BAD_REQUEST
9002	Internal error	STATUS_BAD_REQUEST
9003	Invalid arguments	STATUS_BAD_REQUEST
9004	Resources exceeded (when used in association with SetParameterValues, this cannot be used to indicate Parameters in error)	STATUS_BAD_REQUEST
9011	Upload failure (associated with Upload, TransferComplete or AutonomousTransferComplete methods)	STATUS_BAD_REQUEST
9012	File transfer server authentication failure (associated with Upload, Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST
9013	Unsupported protocol for file transfer (associated with Upload, Download, ScheduleDownload, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST

Once the CPE has attempted to upload the file, the CPE reports the result of the upload operation using the TransferComplete RPC. The TransferComplete RPC indicates a successful operation or one of the fault codes listed in Table 9.1.4.2-2.

Table 9.1.4.2-2 – TransferComplete fault code mapping

Fault code	Description	Response Status Code
9001	Request denied (no reason specified)	STATUS_BAD_REQUEST
9002	Internal error	STATUS_BAD_REQUEST
9010	File transfer failure (associated with Download, ScheduleDownload, TransferComplete or AutonomousTransferComplete methods)	STATUS_BAD_REQUEST
9011	Upload failure (associated with Upload, TransferComplete or AutonomousTransferComplete methods)	STATUS_BAD_REQUEST
9012	File transfer server authentication failure (associated with Upload, Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST
9014	File transfer failure: unable to join multicast group (associated with Download, TransferComplete or AutonomousTransferComplete methods)	STATUS_BAD_REQUEST
9015	File transfer failure: unable to contact file server (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST
9016	File transfer failure: unable to access file (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST
9017	File transfer failure: unable to complete download (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST
9018	File transfer failure: file corrupted or otherwise unusable (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST
9019	File transfer failure: file authentication failure (associated with Download, TransferComplete or AutonomousTransferComplete methods)	STATUS_BAD_REQUEST
9020	File transfer failure: unable to complete download within specified time windows (associated with TransferComplete method)	STATUS_BAD_REQUEST

9.1.4.3 Update primitive mapping for download file transfer operations

The Update Request and Response primitives that result in a download file transfer operation (e.g., update attribute of resource [firmware]) shall use the download mechanism defined in [BBF TR-069]. The download mechanism is an asynchronous command that consists of the synchronous Download RPC for the download and the asynchronous TransferComplete RPC. The Download RPC returns a successful response or one of the fault codes listed in Table 9.1.4.3-1. A successful response means that the CPE has accepted the Download RPC.

Table 9.1.4.3-1 – Download fault code mapping

Fault code	Description	Response Status Code
9000	Method not supported	STATUS_BAD_REQUEST
9001	Request denied (no reason specified)	STATUS_BAD_REQUEST
9002	Internal error	STATUS_BAD_REQUEST
9003	Invalid arguments	STATUS_BAD_REQUEST
9004	Resources exceeded (when used in association with SetParameterValues, this cannot be used to indicate Parameters in error)	STATUS_BAD_REQUEST
9010	File transfer failure (associated with Download, ScheduleDownload, TransferComplete or AutonomousTransferComplete methods)	STATUS_BAD_REQUEST
9012	File transfer server authentication failure (associated with Upload, Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST
9013	Unsupported protocol for file transfer (associated with Upload, Download, ScheduleDownload, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST

Once the CPE has attempted to download the file, the CPE reports the result of the download operation using the TransferComplete RPC. The TransferComplete RPC indicates a successful operation or one of the fault codes listed in Table 9.1.4.3-2.

Table 9.1.4.3-2 – TransferComplete fault code mapping

Fault code	Description	Response Status Code
9001	Request denied (no reason specified)	STATUS_BAD_REQUEST
9002	Internal error	STATUS_BAD_REQUEST
9010	File transfer failure (associated with Download, ScheduleDownload, TransferComplete or AutonomousTransferComplete methods)	STATUS_BAD_REQUEST
9011	Upload failure (associated with Upload, TransferComplete or AutonomousTransferComplete methods)	STATUS_BAD_REQUEST
9012	File transfer server authentication failure (associated with Upload, Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST
9014	File transfer failure: unable to join multicast group (associated with Download, TransferComplete or AutonomousTransferComplete methods)	STATUS_BAD_REQUEST
9015	File transfer failure: unable to contact file server (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST
9016	File transfer failure: unable to access file (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST
9017	File transfer failure: unable to complete download (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST

Table 9.1.4.3-2 – TransferComplete fault code mapping

Fault code	Description	Response Status Code
9018	File transfer failure: file corrupted or otherwise unusable (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST
9019	File transfer failure: file authentication failure (associated with Download, TransferComplete or AutonomousTransferComplete methods)	STATUS_BAD_REQUEST
9020	File transfer failure: unable to complete download within specified time windows (associated with TransferComplete method)	STATUS_BAD_REQUEST

9.1.4.4 Update primitive mapping for reboot operation

The Update Request and Response primitives that result in a reboot operation (e.g., reboot attribute of resource [reboot]) shall use the Reboot RPC defined in [BBF TR-069]. The Reboot RPC is asynchronous command. The Reboot RPC returns a successful response or one of the fault codes listed in Table 9.1.4.4-1.

Table 9.1.4.4-1 – Reboot fault code mapping

Fault code	Description	Response Status Code
9001	Request denied (no reason specified)	STATUS_BAD_REQUEST
9002	Internal error	STATUS_BAD_REQUEST
9003	Invalid arguments	STATUS_BAD_REQUEST

9.1.4.5 Update primitive mapping for factory reset operation

The Update Request and Response primitives that result in a factory reset operation (e.g., factoryReset attribute of resource [reboot]) shall use the FactoryReset RPC defined in [BBF TR-069]. The FactoryReset RPC is an asynchronous command. The FactoryReset RPC returns a successful response or one of the fault codes listed in Table 9.1.4.5-1.

Table 9.1.4.5-1 – FactoryReset fault code mapping

Fault code	Description	Response Status Code
9000	Method not supported	STATUS_BAD_REQUEST
9001	Request denied (no reason specified)	STATUS_BAD_REQUEST
9002	Internal error	STATUS_BAD_REQUEST
9003	Invalid arguments	STATUS_BAD_REQUEST

9.1.4.6 Update primitive mapping for software install operation

The Update Request and Response primitives that result in a software installation operation (e.g., install attribute of resource [software]) shall use the ChangeDUState mechanism defined in [BBF TR-069]. The ChangeDUState mechanism is an asynchronous command that consists of the synchronous ChangeDUState RPC for the download and the asynchronous ChangeDUStateComplete RPC. The ChangeDUState RPC returns a successful response or one of the fault codes listed in Table 9.1.4.6-1. A successful response means that the CPE has accepted the ChangeDUState RPC.

Table 9.1.4.6-1 – ChangeDUState fault code mapping

Fault code	Description	Response Status Code
9000	Method not supported	STATUS_BAD_REQUEST
9001	Request denied (no reason specified)	STATUS_BAD_REQUEST
9002	Internal error	STATUS_BAD_REQUEST
9004	Resources exceeded (when used in association with SetParameterValues, this cannot be used to indicate Parameters in error)	STATUS_BAD_REQUEST

Once the CPE has attempted to change the state of the DU, the CPE reports the result of the state change operation using the ChangeDUStateComplete RPC. The ChangeDUStateComplete RPC indicates a successful operation or one of the fault codes listed in Table 9.1.4.6-2.

Table 9.1.4.6-2 – ChangeDUStateComplete fault code mapping

Fault code	Description	Response Status Code
9001	Request denied (no reason specified)	STATUS_BAD_REQUEST
9003	Invalid arguments	STATUS_BAD_REQUEST
9012	File transfer server authentication failure (associated with Upload, Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST
9013	Unsupported protocol for file transfer (associated with Upload, Download, ScheduleDownload, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST
9015	File transfer failure: unable to contact file server (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST
9016	File transfer failure: unable to access file (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST
9017	File transfer failure: unable to complete download (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST
9018	File transfer failure: file corrupted or otherwise unusable (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_BAD_REQUEST
9022	Invalid UUID Format (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install, Update, and Uninstall)	STATUS_BAD_REQUEST
9023	Unknown Execution Environment (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install only)	STATUS_BAD_REQUEST
9024	Disabled Execution Environment (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install, Update, and Uninstall)	STATUS_BAD_REQUEST
9025	Deployment Unit to Execution Environment Mismatch (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install and Update)	STATUS_BAD_REQUEST

Table 9.1.4.6-2 – ChangeDUStateComplete fault code mapping

Fault code	Description	Response Status Code
9026	Duplicate Deployment Unit (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install only)	STATUS_BAD_REQUEST
9027	System Resources Exceeded (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install and Update)	STATUS_BAD_REQUEST
9028	Unknown Deployment Unit (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Update and Uninstall)	STATUS_BAD_REQUEST
9029	Invalid Deployment Unit State (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install, Update and Uninstall)	STATUS_BAD_REQUEST
9030	Invalid Deployment Unit Update - Downgrade not permitted (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Update only)	STATUS_BAD_REQUEST
9031	Invalid Deployment Unit Update - Version not specified (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Update only)	STATUS_BAD_REQUEST
9032	Invalid Deployment Unit Update - Version already exists (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Update only)	STATUS_BAD_REQUEST

9.1.5 Retrieve primitive mapping

The Retrieve Request and Response primitives shall map to the GetParameterValues RPC. The GetParametersValue RPC is defined in [BBF TR-069] as a synchronous RPC and returns a successful response or one of the fault codes listed in Table 9.1.5-1.

Table 9.1.5-1 – GetParameterValues fault code mapping

Fault code	Description	Response Status Code
9001	Request denied (no reason specified)	STATUS_BAD_REQUEST
9002	Internal error	STATUS_BAD_REQUEST
9003	Invalid arguments	STATUS_BAD_REQUEST
9004	Resources exceeded (when used in association with SetParameterValues, this cannot used to indicate Parameters in error)	STATUS_BAD_REQUEST
9005	Invalid Parameter name (associated with Set/GetParameterValues, GetParameterNames, Set/GetParameterAttributes, AddObject, and DeleteObject)	STATUS_BAD_REQUEST

9.1.6 Notify primitive mapping

9.1.6.0 Introduction

The NotifyRequest and Response primitives permit notifications to the AE or CSEs that have subscribed to a resource.

While [BBF TR-069] has the capability to notify the subscribed ACS when a parameter of an object has been modified, [BBF TR-069] does not have the capability for an ACS to be notified if any parameter within the object has been modified, unless the ACS individually subscribes to all the parameters of the object.

As such, the procedure for mapping the Notify Request and Response primitives for [BBF TR-069] is not possible unless the CSE subscribes to receive notification to all the parameters of an object that are mapped to the resource's attributes.

NOTE – In many implementations, subscribing to all the parameters of an object that are mapped to the resource can cause performance issues in the CPE as well as the CSE. As such, using the attribute-based subscription capabilities of [BBF TR-069] for subscription of resources should be avoided wherever possible.

9.1.6.1 Procedure for subscribed resource attributes

When a <subscription> Resource for a <mgmtObj> resource is created, deleted or updated the CSE shall map to the SetParameterAttributes RPC in the following manner:

- [BBF TR-069] provides the capability to subscribe to changes of a specific attribute through the use of the SetParameterAttributes RPC using the "Active" value for the Notification parameter;
- [BBF TR-069] provides the capability to unsubscribe to changes of a specific attribute through the use of the SetParameterAttributes RPC using the "None" value for the Notification parameter.

The SetParametersAttributes RPC is defined in [BBF TR-069] as a synchronous RPC and returns a successful response or one of the fault codes listed in Table 9.1.6.1-1.

Table 9.1.6.1-1 – SetParameterAttributes fault code mapping

Fault code	Description	Response Status Code
9000	Method not supported	STATUS_BAD_REQUEST
9001	Request denied (no reason specified)	STATUS_BAD_REQUEST
9002	Internal error	STATUS_BAD_REQUEST
9003	Invalid arguments	STATUS_BAD_REQUEST
9004	Resources exceeded (when used in association with SetParameterValues, this cannot be used to indicate Parameters in error)	STATUS_BAD_REQUEST
9010	File transfer failure (associated with Download, ScheduleDownload, TransferComplete or AutonomousTransferComplete methods)	STATUS_BAD_REQUEST

9.1.6.2 Notification primitive mapping

Notify Request and Response primitives shall map to the BBF TR-069 notification mechanism. CPEs produce notifications for subscribed attributes using the Inform method of [BBF TR-069]; the Inform method has an argument Event, which has, as one of the EventCodes, a value "4 VALUE CHANGE", indicating that a subscribed parameter's value has changed. The parameter(s) that have changed are included in the ParameterList argument of the Inform method.

The ParameterList argument is a list of name-value pairs; the name is parameter name and shall be mapped to the objectPath attribute of the Resource, while the value is the most recent value of the parameter.

NOTE – BBF TR-069 CPEs do not report value changes of parameters that were modified by the ACS.

9.2 <mgmtCmd> and <execInstance> resource primitive mappings

9.2.1 Update (Execute) primitive for the <mgmtCmd> resource

9.2.1.0 Introduction

When the Update Request primitive for <mgmtCmd> resource addresses the execEnable attribute of the <mgmtCmd> resource, it effectively triggers an Execute <mgmtCmd> procedure.

The Hosting CSE performs command conversion of its <execInstance> subresources. The mapping between the <execInstance> attributes and the BBF TR-069 RPC procedures triggered is based on the value of the cmdType attribute of the <mgmtCmd> resource defined in Table 9.2.1.0-1. The CPE acceptance of the corresponding RPC procedures is indicated by returning a successful Response primitive to the initial Update Request.

The fault codes that may be returned by the CPE to the Hosting CSE are mapped on to execStatus codes and stored in the corresponding <execInstance> attributes, and are detailed in clauses 9.2.1.1 to 9.2.1.7.

Table 9.2.1.0-1 – Mapping of Execute <mgmtCmd> primitives to a BBF TR-069 RPC

cmdType value	BBF TR-069 RPCs
"DOWNLOAD"	Download RPC (see clause 9.2.1.1) and TransferComplete RPC (clause 9.2.1.3)
"UPLOAD"	Upload RPC (clause 9.2.1.2) and TransferComplete RPC (clause 9.2.1.3)
"SOFTWAREINSTALL"	ChangeDUState RPC (clause 9.2.1.4) and ChangeDUStateComplete RPC (clause 9.2.1.5)
"SOFTWAREUNINSTALL"	ChangeDUState RPC (clause 9.2.1.4) and ChangeDUStateComplete RPC (clause 9.2.1.5)
"REBOOT"	Reboot RPC (clause 9.2.1.6)
"RESET"	Factory reset RPC (clause 9.2.1.7)

9.2.1.1 Execute file download

The download file transfer operation may use the download mechanism defined in [BBF TR-069]. The download mechanism is an asynchronous command that returns a successful response or one of the fault codes mapped on to execStatus values as listedd in Table 9.2.1.1-1. A successful response to the Update primitive triggering the Execute procedure means that the CPE has accepted the Download RPC.

Table 9.2.1.1-1 – Download fault code mapping

Fault code	Description	execStatus Code
9000	Method not supported	STATUS_REQUEST_UNSUPPORTED
9001	Request denied (no reason specified)	STATUS_REQUEST_DENIED
9002	Internal error	STATUS_INTERNAL_ERROR
9003	Invalid arguments	STATUS_INVALID_ARGUMENTS
9004	Resources exceeded (when used in association with SetParameterValues, this cannot be used to indicate Parameters in error)	STATUS_RESOURCES_EXCEEDED
9010	File transfer failure (associated with Download, ScheduleDownload, TransferComplete or AutonomousTransferComplete methods)	STATUS_FILE_TRANSFER_FAILED

Table 9.2.1.1-1 – Download fault code mapping

Fault code	Description	execStatus Code
9012	File transfer server authentication failure (associated with Upload, Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods, not associated with Scheduled Download method)	STATUS_FILE_TRANSFER_SERVER_AUTHENTICATION_FAILURE
9013	Unsupported protocol for file transfer (associated with Upload, Download, ScheduleDownload, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_UNSUPPORTED_PROTOCOL

9.2.1.2 Execute file upload operations

The upload file transfer operation shall use the Upload mechanism defined in [BBF TR-069]. The Upload mechanism is an asynchronous command that consists of the synchronous Upload RPC for the Upload and the asynchronous TransferComplete RPC. The Upload RPC returns a successful response or one of the fault codes mapped on to execStatus values as listed in Table 9.2.1.2-1. A successful response to the Update primitive triggering the execute procedure means that the CPE has accepted the Upload RPC in Table 9.2.1.2-1.

Table 9.2.1.2-1 – Upload fault code mapping

Fault code	Description	execStatus Code
9000	Method not supported	STATUS_REQUEST_UNSUPPORTED
9001	Request denied (no reason specified)	STATUS_REQUEST DENIED
9002	Internal error	STATUS_INTERNAL_ERROR
9003	Invalid arguments	STATUS_INVALID_ARGUMENTS
9004	Resources exceeded (when used in association with SetParameterValues, this cannot be used to indicate Parameters in error)	STATUS_RESOURCES_EXCEEDED
9011	Upload failure (associated with Upload, TransferComplete or AutonomousTransferComplete methods)	STATUS_UPLOAD_FAILED
9012	File transfer server authentication failure (associated with Upload, Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_FILE_TRANSFER_SERVER_AUTHENTICATION_FAILURE
9013	Unsupported protocol for file transfer (associated with Upload, Download, ScheduleDownload, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_UNSUPPORTED_PROTOCOL

9.2.1.3 Report results using TransferComplete RPC

After a file download or upload has been attempted, the result of the operation is reported using the TransferComplete RPC. The TransferComplete RPC indicates a successful operation or one of the fault codes mapped on to execStatus values as listed in Table 9.2.1.3-2.

Table 9.2.1.3-2 – TransferComplete fault code mapping

Fault code	Description	execStatus Code
9001	Request denied (no reason specified)	STATUS_REQUEST DENIED
9002	Internal error	STATUS_INTERNAL_ERROR
9010	File transfer failure (associated with Download, ScheduleDownload, TransferComplete or AutonomousTransferComplete methods)	STATUS_FILE_TRANSFER_FAILED
9011	Upload failure (associated with Upload, TransferComplete or AutonomousTransferComplete methods)	STATUS_UPLOAD_FAILED
9012	File transfer server authentication failure (associated with Upload, Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_FILE_TRANSFER_SERVER_AUTHENTICATION_FAILURE
9014	File transfer failure: unable to join multicast group (associated with Download, TransferComplete or AutonomousTransferComplete methods)	STATUS_FILE_TRANSFER_FAILED_MULTICAST_GROUP_UNABLE_JOIN
9015	File transfer failure: unable to contact file server (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_FILE_TRANSFER_FAILED_SERVER_CONTACT_FAILED
9016	File transfer failure: unable to access file (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_FILE_TRANSFER_FAILED_FILE_ACCESS_FAILED
9017	File transfer failure: unable to complete download (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_FILE_TRANSFER_FAILED_DOWNLOAD_INCOMPLETE
9018	File transfer failure: file corrupted or otherwise unusable (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_FILE_TRANSFER_FAILED_FILE_CORRUPTED
9019	File transfer failure: file authentication failure (associated with Download, TransferComplete or AutonomousTransferComplete methods)	STATUS_FILE_TRANSFER_FILE_AUTHENTICATION_FAILURE
9020	File transfer failure: unable to complete download within specified time windows (associated with TransferComplete method)	STATUS_FILE_TRANSFER_WINDOW_EXCEEDED

9.2.1.4 Execute software operations with ChangeDUState RPC

The software installation and uninstall operations shall use the ChangeDUState mechanism defined in [BBF TR-069]. The ChangeDUState mechanism is an asynchronous command that consists of the synchronous ChangeDUState RPC and returns a successful response or one of the fault codes mapped on to execStatus values as listed in Table 9.2.1.4-1. A successful response to the Update primitive triggering the Execute procedure means that the CPE has accepted the ChangeDUState RPC.

Table 9.2.1.4-1 – ChangeDUState fault code mapping

Fault code	Description	execStatus Code
9000	Method not supported	STATUS_REQUEST_UNSUPPORTED
9001	Request denied (no reason specified)	STATUS_REQUEST DENIED
9002	Internal error	STATUS_INTERNAL_ERROR
9004	Resources exceeded (when used in association with SetParameterValues, this cannot be used to indicate Parameters in error)	STATUS_RESOURCES_EXCEEDED

9.2.1.5 Report Results with ChangeDUStateComplete RPC

After software installation and uninstall operations using a ChangeDUState mechanism as defined in [BBF TR-069], the result of the state change operation is retrieved using the ChangeDUStateComplete RPC. The ChangeDUStateComplete RPC indicates a successful operation or one of the fault codes mapped on to execStatus values as listed in Table 9.2.1.5-1.

Table 9.2.1.5-1 – ChangeDUStateComplete fault code mapping

Fault code	Description	execStatus Code
9001	Request denied (no reason specified)	STATUS_REQUEST_DENIED
9003	Invalid arguments	STATUS_INVALID_ARGUMENTS
9012	File transfer server authentication failure (associated with Upload, Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_FILE_TRANSFER_SERVER_AUTHENTICATION_FAILURE
9013	Unsupported protocol for file transfer (associated with Upload, Download, ScheduleDownload, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_UNSUPPORTED_PROTOCOL
9015	File transfer failure: unable to contact file server (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_FILE_TRANSFER_FAILED_SERVER_CONTACT_FAILED
9016	File transfer failure: unable to access file (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_FILE_TRANSFER_FAILED_FILE_ACCESS_FAILED

Table 9.2.1.5-1 – ChangeDUStateComplete fault code mapping

Fault code	Description	execStatus Code
9017	File transfer failure: unable to complete download (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_FILE_TRANSFER_FAILED_DOWNLOAD_INCOMPLETE
9018	File transfer failure: file corrupted or otherwise unusable (associated with Download, TransferComplete, AutonomousTransferComplete, DUStateChangeComplete, or AutonomousDUStateChangeComplete methods)	STATUS_FILE_TRANSFER_FAILED_FILE_CORRUPTED
9022	Invalid UUID Format (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install, Update, and Uninstall)	STATUS_INVALID_UUID_FORMAT
9023	Unknown Execution Environment (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install only)	STATUS_UNKNOWN_EXECUTION_ENVIRONMENT
9024	Disabled Execution Environment (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install, Update, and Uninstall)	STATUS_DISABLED_EXECUTION_ENVIRONMENT
9025	Deployment Unit to Execution Environment Mismatch (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install and Update)	STATUS_EXECUTION_ENVIRONMENT_MISMATCH
9026	Duplicate Deployment Unit (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install only)	STATUS_DUPLICATE_DEPLOYMENT_UNIT
9027	System Resources Exceeded (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install and Update)	STATUS_SYSTEM_RESOURCES_EXCEEDED
9028	Unknown Deployment Unit (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Update and Uninstall)	STATUS_UNKNOWN_DEPLOYMENT_UNIT
9029	Invalid Deployment Unit State (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Install, Update and Uninstall)	STATUS_INVALID_DEPLOYMENT_UNIT_STATE
9030	Invalid Deployment Unit Update - Downgrade not permitted (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Update only)	STATUS_INVALID_DEPLOYMENT_UNIT_UPDATE_DOWNGRADE_DISALLOWED
9031	Invalid Deployment Unit Update - Version not specified (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Update only)	STATUS_INVALID_DEPLOYMENT_UNIT_UPDATE_UPGRADE_DISALLOWED

Table 9.2.1.5-1 – ChangeDUStateComplete fault code mapping

Fault code	Description	execStatus Code
9032	Invalid Deployment Unit Update - Version already exists (associated with DUStateChangeComplete or AutonomousDUStateChangeComplete methods: Update only)	STATUS_INVALID_DEPLOYMENT_UNIT_UPDATE_VERSION_EXISTS

9.2.1.6 Execute reboot operation

The reboot operation shall use the Reboot RPC defined in [BBF TR-069]. The Reboot RPC is a synchronous command. A successful response to the Update primitive triggering the execute procedure means that the CPE has accepted the Reboot RPC. The Reboot RPC returns a successful response or one of the fault codes mapped on to execStatus values as listed in Table 9.2.1.6-1.

Table 9.2.1.6-1 – Reboot fault code mapping

Fault code	Description	execStatus Code
9001	Request denied (no reason specified)	STATUS_REQUEST_DENIED
9002	Internal error	STATUS_INTERNAL_ERROR
9003	Invalid arguments	STATUS_INVALID_ARGUMENTS

9.2.1.7 Execute factory reset operation

The factory reset operation shall use the FactoryReset RPC defined in [BBF TR-069]. The FactoryReset RPC is a synchronous command. A successful response to the Update primitive triggering the Execute procedure means that the CPE has accepted the FactoryReset RPC. The FactoryReset RPC returns a successful response or one of the fault codes mapped on to execStatus values as listed in Table 9.2.1.7-1.

Table 9.2.1.7-1 – FactoryReset fault code mapping

Fault code	Description	execStatus Code
9000	Method not supported	STATUS_REQUEST_UNSUPPORTED
9001	Request denied (no reason specified)	STATUS_REQUEST_DENIED
9002	Internal error	STATUS_INTERNAL_ERROR
9003	Invalid arguments	STATUS_INVALID_ARGUMENTS

9.2.2 Delete <mgmtCmd> resource primitive mapping

The Delete Request primitive for the <mgmtCmd> resource may initiate BBF TR-069 RPC commands for the corresponding <execInstance> subresources as follows:

- If there are no <execInstance> subresources with RUNNING execStatus, a successful response to the Delete primitive is returned and the <mgmtCmd> resource is deleted without triggering any BBF TR-069 RPCs.
- If there are <execInstance> subresources with RUNNING execStatus that resulted in cancellable BBF TR-069 RPCs (e.g., File Upload and File Download RPCs), a BBF TR-069 CancelTransfer RPC shall be initiated for each cancellable operation. Upon completion of all the cancellation operations, if any fault codes are returned by the CPE, an unsuccessful Response to the Delete primitive with status code "Delete mgmtCmd- execInstance

cancellation error" is returned, and the <mgmtCmd> resource is not deleted. The execStatus attribute of each specific <execInstance> is set to CANCELLED for successful RPCs or is determined from the RPC fault codes as listed in Table 9.2.2-1. If all cancellation operations are successful on the managed entity, a successful Response to the Delete primitive is returned and the <mgmtCmd> resource is deleted.

- If there is at least one <execInstance> subresource with RUNNING execStatus that resulted in non-cancellable BBF TR-069 RPCs (e.g., RPCs other than File Upload and File Download RPCs), the execStatus attribute of the specific <execInstance> is changed to STATUS_NON_CANCELLABLE. An unsuccessful Response to the Delete primitive with status code "Delete mgmtCmd- execInstance cancellation error" is returned and the <mgmtCmd> resource is not deleted.

Table 9.2.2-1 – CancelTransfer fault code mapping for Delete <mgmtCmd>

Fault code	Description	Response Status Code
9000	Method not supported	STATUS_REQUEST_UNSUPPORTED
9001	Request denied (no reason specified)	STATUS_REQUEST DENIED
9021	Cancelation of file transfer not permitted in current transfer state	STATUS_CANCELLATION_DENIED

9.2.3 Update (Cancel) <execInstance> primitive mapping

When the Update Request primitive for an <execInstance> subresource addresses the execDisable attribute of the <execInstance > subresource, it effectively triggers a Cancel <execInstance> resource procedure.

The hosting CSE determines whether the <execInstance> resource has a RUNNING execStatus and whether the resulting BBF TR-069 RPCs are cancellable. Currently, only the BBF TR-069 File Upload and File Download RPCs are cancellable using the BBF TR-069 CancelTransfer RPC.

- If the addressed <execInstance> subresource has an execStatus other than RUNNING, an unsuccessful Response to the Update primitive is returned with status code "Cancel execInstance – already complete".
- If the addressed <execInstance> subresources has RUNNING execStatus and resulted in cancellable BBF TR-069 RPCs (e.g., File Upload and File Download RPCs), a BBF TR-069 CancelTransfer RPC shall be initiated. For a successful CancelTransfer RPC, the execStatus attribute of the specific <execInstance> is set to CANCELLED and a successful Response is sent to the Update primitive. For an unsuccessful CancelTransfer RPC, the execStatus attribute is determined from the RPC fault codes as listed in Table 9.2.3-1 and an unsuccessful Response is sent to the Update primitive with status code "Cancel execInstance – cancellation error".
- If the addressed <execInstance> subresource has RUNNING execStatus and resulted in non-cancellable BBF TR-069 RPCs (e.g., RPCs other than File Upload and File Download RPCs), the execStatus attribute of the specific <execInstance> is changed to STATUS_NON_CANCELLABLE. An unsuccessful Response is sent to the Update primitive with status code "Cancel execInstance – not cancellable".

Table 9.2.3-1 – CancelTransfer fault code mapping for update (Cancel) <execInstance>

Fault code	Description	execStatus Code
9000	Method not supported	STATUS_REQUEST_UNSUPPORTED
9001	Request denied (no reason specified)	STATUS_REQUEST DENIED
9021	Cancelation of file transfer not permitted in current transfer state	STATUS_REQUEST_UNSUPPORTED

9.2.4 Delete <execInstance> primitive mapping

The Delete Request primitive for an <execInstance> subresource may initiate BBF TR-069 RPC commands for the corresponding <execInstance> subresources as follows.

- If the addressed <execInstance> subresource has an execStatus other than RUNNING, a successful Response to the Delete primitive is returned and the <execInstance> subresource is deleted without triggering any BBF TR-069 RPCs.
- If the addressed <execInstance> subresource has RUNNING execStatus and resulted in cancellable BBF TR-069 RPCs (e.g., File Upload and File Download RPCs), a BBF TR-069 CancelTransfer RPC shall be initiated. For a successful CancelTransfer RPC, a successful response is sent to the Delete primitive and the <execInstance> subresource is deleted. For an unsuccessful CancelTransfer RPC, the execStatus attribute is determined from the RPC fault codes as listed in Table 9.2.4-1 and an unsuccessful Response is sent to the Delete primitive with status code "Delete execInstance - cancellation failed".
- If the addressed <execInstance> subresource has RUNNING execStatus and resulted in non-cancellable BBF TR-069 RPCs (e.g., RPCs other than File Upload and File Download RPCs), the execStatus attribute is set to STATUS_NOT_CANCELLABLE and an unsuccessful Response is sent to the Update primitive with status code "Delete execInstance – not cancellable".

Table 9.2.4-1 – CancelTransfer fault code mapping for Delete <execInstance>

Fault code	Description	execStatus Code
9000	Method not supported	STATUS_REQUEST_UNSUPPORTED
9001	Request denied (no reason specified)	STATUS_REQUEST DENIED
9021	Cancelation of file transfer not permitted in current transfer state	STATUS_CANCELLATION_DENIED

10 Server interactions

10.0 Introduction

This clause specifies how the infrastructure node-common services entity (IN-CSE) interacts with an ACS in order to manage the Resources described in this Recommendation. The IN-CSE interaction with an ACS includes:

- establishment of the communication session between the IN-CSE and ACS;
- processing of requests and notifications between the IN-CSE and the ACS;
- discovery.

NOTE – The Broadband Forum (BBF) has not defined a protocol specification for the northbound interface of an ACS. As such, this Recommendation only describes the expectations of this interface in the form of requirements on the ACS.

10.1 Communication session establishment

10.1.1 IN-CSE to ACS communication session establishment

When the IN-CSE detects that it has to delegate an interaction with a device resource to an ACS, the IN-CSE establishes a communication session with the ACS. The establishment of a communication session between the IN-CSE and ACS provides security dimensions for access control, authentication, non-repudiation, data confidentiality, communication security, data integrity and privacy in accordance with architectural requirement A7 of [BBF TR-131].

The IN-CSE may establish multiple sessions with an ACS based on the security model utilized between the IN-CSE and the ACS.

10.1.2 ACS to IN-CSE communication session establishment

When the ACS detects a change to resources it manages in which the IN-CSE has expressed interest, the ACS requests the IN-CSE to establish a session if a session does not exist for the resource being managed. The establishment of a communication session between the IN-CSE and ACS provides security dimensions for access control, authentication, non-repudiation, data confidentiality, communication security, data integrity and privacy in accordance with architectural requirement A7 of [BBF TR-131].

The ACS may establish multiple sessions with an IN-CSE based on the security model utilized between the IN-CSE and the ACS.

While a session between the ACS and IN-CSE is not established, the ACS retains any notifications or changes in the resources based on an Event retention policy (i.e., time, number of events).

When an ACS to IN-CSE interaction is required and a session does not exist, the ACS requests a session based on a session initiation policy [i.e., periodic contact establishment (schedule) upon event detection with timeframe window] to be initiated.

10.1.3 ACS and IN-CSE communication session requirements

When establishing a session from the ACS to the IN-CSE:

- if a session does not exist between the IN-CSE and ACS, the ACS shall retain any notifications or changes in the resources based on an Event retention policy (i.e., time, number of events);
- when an ACS to IN-CSE interaction is required and a session does not exist, the ACS shall be capable of initiating a session based on a session initiation policy [i.e., periodic contact establishment (schedule) upon event detection with timeframe window].

10.2 Processing of requests and responses

10.2.1 Request and notification formatting

Request and notification mechanisms between the IN-CSE and the device management (DM) server format the extensible markup language (XML) schema of the CPE methods defined in [BBF TR-069] as an ACS would format the CPE methods that it would pass to the CPE. The IN-CSE would then also process the CPE methods as defined in [BBF TR-069]. Likewise the ACS would send notifications in the format of the XML schema of the CPE for sending events using the Inform RPC.

10.2.2 ACS request processing requirements

When receiving requests from the IN-CSE, the ACS shall be capable of defining mechanisms to support triggering of immediate operations to device. If the device is not available, the ACS returns an appropriate error code.

The ACS shall provide capability for the IN-CSE to indicate request policies to include: retry policy, request time out.

10.2.3 ACS notification processing requirements

When sending notifications to the IN-CSE:

- the ACS shall be capable of providing a mechanism for the IN-CSE to subscribe to events;
- the ACS shall be capable of providing a list of events for which the IN-CSE can subscribe;
- the ACS shall be capable of providing a mechanism for the IN-CSE to unsubscribe from events;
- the ACS shall be capable of providing an event delivery mechanism;
- the ACS shall be capable of providing the capability for the IN-CSE to request event filters including: event code; specific parameters changing value; device; any combination of the previous criteria;
- the IN-CSE shall be capable of subscribing to be notified of changes to resources it manages;
- the ACS shall be capable of notifying the IN-CSE of changes to resources to which the client has subscribed.

10.3 Discovery and synchronization of resources

For devices under management, the IN-CSE may discover resources of interest (metadata and values) within a device using the ACS.

For resources of interest, the IN-CSE may also express an interest to be notified of a resource if a resource is changed (added, deleted, updated).

The IN-CSE shall be capable of discovering and subscribing to changes of resources in order to synchronize the IN-CSE with resources of interest of the ACS.

10.4 Access management

10.4.0 Introduction

Once a request has obtained an access decision from the IN-CSE to allow the request, the IN-CSE shall select the appropriate ACS along with elements the ACS would need to implement access management within the ACS. These elements include the identity of the subject (oneM2M Originator) of the request which is needed in scenarios where the original issuer of the request needs to be known - this could be done by correlating principals (e.g., roles, accounts) used by the IN-CSE and ACS.

10.4.1 Access management requirements

- The ACS shall be capable of providing a mechanism for the IN-CSE to discover the access management elements used to authorize and authenticate access to resources controlled by the ACS.
- The IN-CSE shall be capable of correlating access management elements provided by the ACS to access management elements used by the IN-CSE.
- The IN-CSE shall be capable of providing secured storage of access management elements within the IN-CSE.

11 New management technology specific resources

[BBF TR-181] provides a list of management objects that have been standardized by the BBF and where possible, clause 7 provides a mapping of the resources to standardized management objects. This clause provides the oneM2M vendor specific extensions to the BBF TR-181 data model as specified in the ts-0006-1-2-0.xml [b-oneM2M.XML].

Annex A

oneM2M specification update and maintenance control procedure

(This annex forms an integral part of this Recommendation.)

The provisions of Annex L in [ITU-T Y.4500.1] relating to the oneM2M specification update and maintenance control procedure shall apply to this Recommendation.

Bibliography

[b-ATIS.oneM2M.TS0006V201] ATIS.oneM2M.TS0006V201-2016, *Management enablement (BBF).*

[b-CCSA M2M-TS-0006-V2.0.1] CCSA M2M-TS-0006-V2.0.1 (2016), *Management enablement (BBF).*

[b-ETSI TS 118 106] ETSI TS 118 106 V2.0.1 (2016), *oneM2M; Management enablement (BBF) – (oneM2M TS-0006 version 2.0.1 Release 2)*

[b-oneM2M.XML] oneM2M XML Schemas. http://www.onem2m.org/technical/xml-schemas

[b-TSDSI STD T1.oneM2M TS-0006-2.0.1] TSDSI STD T1.oneM2M TS-0006-2.0.1 V1.0.0 (2017), *Management enablement (BBF).*

[b-TTAT.MM-TS.0006 v2.0.1] TTAT.MM-TS.0006 v2.0.1 (2017), *oneM2M – Management enablement (BBF).*

[b-TTC TS-M2M-0006v2.0.1] TTC TS-M2M-0006v2.0.1 (2016), *oneM2M technical specification –- Management enablement (BBF).*

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	**Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities**
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2018