ONEM2M TECHNICAL SPECIFICATION

Document Number	oneM2M-TS-0004-V-0.6.1
Document Name:	oneM2M Service Layer Protocol Core Specification
Date:	2014-Aug-01
Abstract:	The present document specifies the communication protocol(s) for oneM2M compliant Systems, M2M Applications, and/or other M2MSystems.The present document also specifies the common data formats, interfaces and message sequences to support reference points(s) defined by oneM2M.

About oneM2M

The purpose and goal of oneM2M is to develop technical specifications which address the need for a common M2M Service Layer that can be readily embedded within various hardware and software, and relied upon to connect the myriad of devices in the field with M2M application servers worldwide.

More information about oneM2M may be found at: http//www.oneM2M.org

Notice of Disclaimer & Limitation of Liability

The information provided in this document is directed solely to professionals who have the appropriate degree of experience to understand and interpret its contents in accordance with generally accepted engineering or other professional standards and applicable regulations. No recommendation as to products or vendors is made or should be implied.

NO REPRESENTATION OR WARRANTY IS MADE THAT THE INFORMATION IS TECHNICALLY ACCURATE OR SUFFICIENT OR CONFORMS TO ANY STATUTE, GOVERNMENTAL RULE OR REGULATION, AND FURTHER, NO REPRESENTATION OR WARRANTY IS MADE OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR AGAINST INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS. NO oneM2M PARTNER TYPE 1 SHALL BE LIABLE, BEYOND THE AMOUNT OF ANY SUM RECEIVED IN PAYMENT BY THAT PARTNER FOR THIS DOCUMENT, WITH RESPECT TO ANY CLAIM, AND IN NO EVENT SHALL oneM2M BE LIABLE FOR LOST PROFITS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES. oneM2M EXPRESSLY ADVISES ANY AND ALL USE OF OR RELIANCE UPON THIS INFORMATION PROVIDED IN THIS DOCUMENT IS AT THE RISK OF THE USER.

Contents

1 Scope

The present document specifies the communication protocol(s) for oneM2M compliant Systems, M2M Applications, and/or other M2MSystems.

The present document also specifies the common data formats, interfaces and message sequences to support reference points(s) defined by oneM2M.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

2.1. Normative references

The following referenced documents are necessary for the application of the present document.

[1] IETF RFC 5139: "Revised Civic Location Format for Presence Information Data Format Location Object (PIDF-LO)".

[2] IETF RFC 3986: "Uniform Resource Identifier (URI): Generic Syntax".

[3] W3C XMLSchemaP2: "W3C Recommendation (2004), XML Schema Part 2:Datatypes Second Edition.".

[4] oneM2M TS-0005 Management Enablement (OMA)[5]oneM2M TS-0006 Management Enablement (BBF)

[6] oneM2M TS-0001 "Functional Architecture". TBD.

[7] oneM2M TS-0003 Security Solutions

[8] IEEE 754-2008: IEEE. IEEE Standard for Floating-Point Arithmetic. 29 August 2008. http://ieeexplore.ieee.org/servlet/opac?punumber=4610933

[9] IETF RFC 3548: "The Base16, Base32, and Base64 Data Encodings". 2003.

[10] IETF RFC 2045: "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies". 1996.

[11] IETF RFC 3987:" Internationalized Resource Identifiers (IRIs)" . January 2005.

[12] IETF BCP 47: "Best Current Practices 47". Concatenation of RFC 4646:" Tags for Identifying Languages"(2006) and RFC 4647: "Matching of Language Tags"(2006).

[13] IETF RFC 3588: "Diameter Base Protocol". September 2003.

[14] IETF RFC 6733: "Diameter Base Protocol". October 2012.

[15] 3GPP TS 23.682: "Architecture enhancements to facilitate communications with packet data networks and applications" Release 11.

[16] 3GPP TS 29.368: "Tsp interface protocol between the MTC Interworking Function (MTC-IWF) and Service Capability Server (SCS)" Release 11.

コメントの追加 **[AvT1]:** Empty reference. Mentioned in the document in table 6.3.2.1-1. Please update.

[17] 3GPP TS 23.003: "Numbering, addressing and identification".

[18] IETF RFC 4282: "The Network Access Identifier".

2.2. Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] oneM2M Drafting Rules.

NOTE: Available at http://member.onem2m.org/Static_pages/Others/Rules_Pages/oneM2M-Drafting-Rules-V1_0.doc.

[i.2] Fielding, Roy Thomas (2000): "Architectural Styles and the Design of Network-based Software Architectures", Doctoral dissertation, University of California, Irvine.

[i.3] OMA-TS-REST-NetAPI_TerminalLocation-V1_0-20130924-A: "RESTful Network API for Terminal Location", Version 1.0.

[i.4] "RESTful Network API for Notification Channel", Open Mobile Alliance™, OMA-TS-REST_NetAPI_NotificationChannel-V1_0.

[i.5] OMA-TS-MLP-V3_4-20130226-C: "Mobile Location Protocol", Version 3.4.

コメントの追加 **[AvT2]:** These references are not mentioned in the document. Move to the Bibliography or delete them.

3 Definitions, symbols, abbreviations and acronyms

Delete from the above heading the word(s) which is/are not applicable.

3.1. Definitions

Clause numbering depends on applicability.

- ***A definition shall not take the form of, or contain, a requirement.***
- ***The form of a definition shall be such that it can replace the term in context. Additional information shall be given only in the form of examples or notes (see below).***
- ***The terms and definitions shall be presented in alphabetical order.***

For the purposes of the present document, the [following] terms and definitions [given in ... and the following] apply:

Definition format

<defined term>: <definition>

If a definition is taken from an external source, use the format below where [N] *identifies the external document which must be listed in Section 2 References.*

<defined term>[N]: <definition>

example 1: text used to clarify abstract rules by applying them literally

NOTE: This may contain additional information.

3.2. Symbols

Clause numbering depends on applicability.

For the purposes of the present document, the [following] symbols [given in ... and the following] apply:

Symbol format

<symbol>	<Explanation>
<2nd symbol>	<2nd Explanation>
<3rd symbol>	<3rd Explanation>

3.3. Abbreviations

For the purposes of the present document, the abbreviations given in oneM2M-TS-0011-Definitions and Acronyms-V0.6.0 and the following apply:

3GPP	3rd Generation Partnership Project
AE	application entity
API	application programming interface
AVP	attribute value pair
BBF	Broadband Forum
BCP	best current practices
CMDH	communication management and delivery handling
CoAP	Constrained Application Protocol
CRUD	The operations CREATE. RETRIEVE, UPDATE and DELETE
CRUD+N	The operations CREATE. RETRIEVE, UPDATE, DELETE and NOTIFY
CSE	common services entity
CSEBase	A <CSEBase> resource shall represent a CSE.
CSE-ID	Identifier of the CSE
DAA	device action answer
DAR	device-action-request
DNR	device notification request
HTTP	Hypertext Transfer Protocol
ID	identifier
IETF	Internet Engineering Task Force
IN-CSE	CSE which resides in the Infrastructure Node
IRI	internationalized resource identifier reference
JSON	JavaScript Object Notation
M2M	machine to machine
Mcc	Communication flows between two Common Services Entities (CSEs) cross the Mcc reference point.
Mca	Communication flows between an Application Entity (AE) and a Common Services Entity (CSE) cross the Mca reference point.
MQTT	
MSISDN	mobile subscriber integrated services digital network-number
MTC-IWF	achine type communications - interworking function
OMA	Open Mobile Alliance
RFC	request for comment
RSC	response status codes
SCS-Identifier	services capability server identifier
TS	technical specification
URI	universal resource identifier
XML	extensible markup language
XSD	XML schema definition

3.4. Acronyms

Acronyms should be ordered alphabetically.

Clause numbering depends on applicability.

For the purposes of the present document, the [following] abbreviations [given in ... and the following] apply:

Acronym format

<ACRONYM1> <Explanation>
<ACRONYM2> <Explanation>
<ACRONYM3> <Explanation>

4 Conventions

The key words "Shall", "Shall not", "May", "Need not", "Should", "Should not" in the present document are to be interpreted as described in the oneM2M Drafting Rules [i.1].

5 Protocol Design Principles and Requirements

The following clauses contain the design principles and requirements for the oneM2M protocol.

Editor's Note: The following sub-clauses are intended to provide design principle and specify aspects of protocol requirements, including but not limited to scalability, performance, common message format, reliability, security, extensibility, robustness, resilience, efficiency, message minimisation, etc. Requirements derived from the Requirements TS and inferred from the evolving Architecture TS should be captured here and expressed in terms of Protocol Requirements. Beyond these, additional Protocol requirements are expected. More contributions are requested

5.1. Introduction

The oneM2M architecture is resource-based, (see oneM2M TS-0001 [6]). The functionality of the system is exposed by means of APIs over the reference points specified in [6]. Operations upon resources hosted by a CSE are carried over an established channel that constitutes the communication on the reference points Mca and Mcc.

Each resource operation comprises a pair of primitives: Request and Response.

In order to provide a well-defined interface for the reference points in the architecture [6], the following aspects need to be provided:

- the collection of primitives carried over a specific reference point; and
- the description and applicability of security methods in relation to the underlying protocols and reference points involved.

The current document provides:

- data type definitions;
- primitive definitions; and
- XML definitions and schema.

NOTE: The actual binding of the interface to a specific protocol is not part of the present document, but is specified in a separate TS.

In accordance with the oneM2M architecture, each reference point is applicable to a wide range of underlying network technologies and transport protocols. oneM2M will only define a set of bindings for specific underlying network

technologies and transport protocols, these bindings are not limiting the applicability of the interfaces when used in other underlying networks and transport protocols. However the behaviour of the interface needs to be respected in accordance to the present document and the architecture (see [6]).

5.1.1. Interfaces to the Underlying Networks

The CSEs access the network service functions provided by the underlying networks such as 3GPP, 3GPP2 and the M2M Area Networks via Mcn reference point. The following services are provided by the underlying networks:

- Device Triggering (see Annex B)
- Location Request (see Annex G)
- Device managment(see clause 7.2.2.4)

5.2. API Design Guidelines

The following are the guidelines for designing APIs:

1) API shall follow the principle of RESTful architecture, as described in [i.2].
2) APIs shall define how to address resources and how to manipulate resources, in accordance with oneM2M TS-0001 [6]; the resource is identified by a Universal Resource Identifier (URI), [2].
3) Resource has a representation (see [i.2]) that shall be transferred and manipulated with the verbs. These verbs are identified as operations in [6]: CREATE. RETRIEVE, UPDATE, DELETE and NOTIFY.
4) All primitives shall be defined as well as the way that those primitives are sent. The functionality of the primitives shall be compliant to the resource type specific procedure as specified in [6], clause 10.2.
5) API shall provide the format and syntax of the operation primitives for all resources defined in [6]. In case that for a particular protocol binding an operation cannot be supported it has to be clearly stated in the specific protocol binding technical specification.
6) Primitives shall include attributes in accordance with [6] for a specific resource.
7) Primitive shall be self-descriptive and contain all the information needed for the receiver of the primitives to handle the primitives.
8) Primitive should be idempotent operations which means no matter how many times the primitive is sent, the result doesn't change, in accordance to [i.2].
9) API shall indicate which features are supported and not supported over the reference points specified in [6].
10) Primitives shall be mapped on the transport layer protocols.

5.3. Primitives

5.3.1. Introduction

Primitives are service layer messages transmitted over the Mca and Mcc reference points. The Create, Update, Retrieve, Delete and Notify operation is mapped to one or more primitives. The primitive is then further mapped to transport layer protocols such as HTTP, CoAP or MQTT for the transmission. The primitive is independent from the transport protocols.

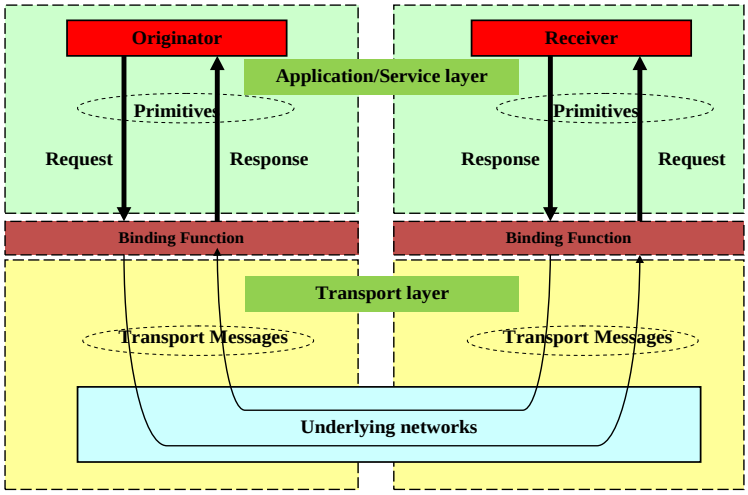


Figure 5.3.1-1: Primitive overview

A single primitive in the application/service layer may be mapped to zero or more transport messages in the transport layer.

The Originators shall send requests to Receivers through primitives. The Originator and Receiver may be AE or CSE. The CRUD request primitive addresses a resource residing in a CSE. The Notify request primitive may address an AE or CSE.

Each CRUD+N operation shall consist of one request and one response primitive.

Communication between an originator and receiver in the Application/Service layer shall be performed with the above primitives.

Communication between an originator and receiver in the Transport layer shall be performed using underlying network.

5.3.2. Primitives modelling

Primitives shall be modelled as follows.

A primitive shall be a data structure that describes with appropriate attributes a specific procedure request or answer in both originator and receiver entities.

A primitive shall consist of:

- control part: contains attributes required for the processing of the request or response; and
- optional content part: user data. This is the representation of the resource or the value of the attribute in partial addressing case.

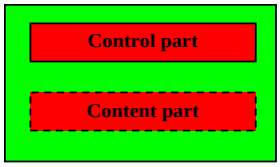


Figure 5.3.2-1: Primitives modeling

5.3.3. Primitive principles

Primitives shall be atomic. When two concurrent request primitives affect the same resource, the execution of one primitive shall finish completely before the second one starts.

When creating or updating the resource, the representation of the resource shall be contained in the Content part of the primitive. Based on the representation of the resource, the Hosting CSE of the addressed resource can create or update the entire resource without the need for further information.

The handling of primitives shall be idempotent. This means no matter how many times the same primitive is targeted to the same resource, the resource does not change after the first handling of the primitive.

5.4. Design Principles

5.4.1. Scalability

When considering scalability as a requirement in the design of oneM2M protocols, one or several of the following mechanisms could be used:

- Wherever possible, ensure direct addressability to the CSEs hosting target resources, to minimize network hops.
- Asynchrony in terms of data processing, with the ultimate objective of minimizing the number of discarded packets.
- Caching mechanisms that allow all the received packets to be processed.
- Efficient load distribution to avoid bottlenecks and data loss.
- Data compression and/or aggregation, in order to reduce the amount of data sent through the network.

5.4.2. Extensibility

The design of extensible oneM2M protocols will consider and mitigate the risk of unintended consequences, such as interoperability issues, operational problems, or security vulnerabilities.

The oneM2M protocols will be designed to allow continued development and to facilitate changes by means of standardized extensions.

The impact of the extensibility on the existing oneM2M protocol functions will be minimized.

As an example, extensibility can be related to one or more of the following aspects:

- Deal withHandling a different number of devices,
- Add, remove or modify oneM2M protocol functionality,
- New oneM2M protocol routines,

- New data types.

The design of extensible oneM2M protocols will consider and mitigate the risk of unintended consequences, such as interoperability issues, operational problems, or security vulnerabilities.

5.4.3. Efficiency

Energy Efficiency:

- As energy consumption directly affects the overall system performance, oneM2M protocols should consider energy efficiency, especially in resource constrained environments with battery-powered oneM2M devices.
- Energy efficient oneM2M protocols will aim at reducing the overall energy consumption while maintaining the performance required by the oneM2M Applications.

6 oneM2M Protocols/API Overview

6.1. Introduction

This technical specification describes message formats and procedures to communicate with oneM2M compliant M2M Platform System.

The present document describes:

- Data representation for communication protocol messages.
- Normal and exceptional procedure.
- Status codes.
- Guidelines for drafting APIs.

6.2. M2M Identifiers

This clause describes all identifiers from [6] and the data types used to represent them.

Table 6.2-1: M2M Identifiers

Identifier	Data Type	Description
M2M-SP-ID	m2m:id (see clause 6.3.2.1)	A globally unique ID as specified in [6]
App-ID	xs:string (see clause 6.3.2.1)	The identifier is specified in [6]
AE-ID	m2m:id (see clause 6.3.2.1)	A globally unique ID as specified in [6]
CSE-ID	m2m:id (see clause 6.3.2.1)	A globally unique ID as specified in [6]
M2M-Node-ID	TBD (see clause 6.3.2.1)	A globally unique ID as specified in [6]
M2M-Sub-ID	m2m:id (see clause 6.3.2.1)	A globally unique ID as specified in [6]
M2M-Request-ID	TBD (see clause 6.3.2.1)	A unique ID as specified in [6]
M2M-Ext-ID	M2m:externalId (see clause 6.3.2.3)	The identifier is specified in [6]
UNetwork-ID	m2m:id (see clause 6.3.2.1)	A unique ID as specified in [6]
Trigger-Recipient-ID	xs:unsignedInt	The identifier is specified in [6]
M2M-Serv-ID	TBD	The identifier is specified in [6]

Editor's Note: data type for M2M-Node-ID, M2M-Sub-ID, and M2M-Serv-ID is TBD

6.3. Common Data Types

For wide acceptance by industrial markets, the present document describes structured and non-structured data for oneM2M Protocol using XML Schema Language (aka XSD) [3].

Note that the actual data format is depends on chosen Protocol Bindings. Each protocol binding specification will specify the mapping rule between XSD version of data definitions and its native data format.

Any Data Types and XML elements defined for use in oneM2M protocols shall use the namespace:

- http://www.onem2m.org/xml/protocols.

The present document, and any XML or XML Schema Documents produced by oneM2M shall use the prefix m2m: to refer to that namespace.

6.3.1. Simple Data Types incorporated from XML Schema

Following 'built-in data types' are incorporated from XML Schema definition [3].

Note that name space identifier for 'http://www.w3.org/2001/XMLSchema' shall be described as 'xs:' in the present document.

Table 6.3.2.1-1: Simple Data Types incorporated from XML Schema

Data Type	Description	Notes
xs:string	The string datatype represents character strings in XML	
xs:boolean	boolean represents the values of two-valued logic.	
xs:decimal	decimal represents a subset of the real numbers, which can be represented by decimal numerals. The value space· of decimal is the set of numbers that can be obtained by dividing an integer by a non-negative power of ten, i.e. expressible as i / 10 n where i and n are integers and n ≥ 0. Precision is not reflected in this value space; the number 2.0 is not distinct from the number 2.00. The order relation on decimal is the order relation on real numbers, restricted to this subset.	
xs:float	The float datatype is patterned after the IEEE single-precision 32-bit floating point datatype IEEE 754-2008 [8]. Its value space is a subset of the rational numbers. Floating point numbers are often used to approximate arbitrary real numbers.	
xs:double	The double datatype is patterned after the IEEE double-precision 64-bit floating point datatype IEEE 754-2008 [8]. Each floating point datatype has a value space that is a subset of the rational numbers. Floating point numbers are often used to approximate arbitrary real numbers.	
xs:duration	duration is a datatype that represents durations of time.	
xs:dateTime	dateTime represents instants of time, optionally marked with a particular time zone offset. Values representing the same instant but having different time zone offsets are equal but not identical.	
xs:time	time represents instants of time that recur at the same point in each calendar day, or that occur in some arbitrary calendar day.	
xs:date	date represents top-open intervals of exactly one day in length on the timelines of dateTime, beginning on the beginning moment of each day, up to but not including the beginning moment of the next day). For non-timezoned values, the top-open intervals disjointly cover the non-timezoned timeline, one per day. For timezoned values, the intervals begin at every minute and therefore overlap.	
xs:hexBinary	hexBinary represents arbitrary hex-encoded binary data.	
xs:base64Binary	base64Binary represents arbitrary Base64-encoded binary data. For base64Binary data the entire binary stream is encoded using the Base64 Encoding defined in RFC 3548 [9], which is derived from the encoding described in RFC 2045 [10].	
xs:anyURI	anyURI represents an Internationalized Resource Identifier Reference (IRI). An anyURI value can be absolute or relative, and may have an optional fragment identifier (i.e. it may be an IRI Reference). This type should be used when the value fulfills the role of an IRI, as defined in RFC 3987 [11] or its successor(s) in the IETF Standards Track.	
xs:normalizedString	normalizedString represents white space normalized strings. The ·value space· of normalizedString is the set of strings that do not contain the carriage return (#xD), line feed (#xA) nor tab (#x9) characters. The lexical space· of normalizedString is the set of strings that do not contain the carriage return (#xD), line feed (#xA) nor tab (#x9) characters. The base type·of normalizedString is string.	
xs:token	token represents tokenized strings. The ·value space· of token is the set of strings that do not contain the carriage return (#xD), line feed (#xA) nor tab (#x9) characters, that have no leading or trailing spaces (#x20) and that have no internal sequences of two or more spaces. The lexical space· of token is the set of strings that do not contain	

Data Type	Description	Notes
	the carriage return (#xD), line feed (#xA) nor tab (#x9) characters, that have no leading or trailing spaces (#x20) and that have no internal sequences of two or more spaces. The base type·of token is normalizedString.	
xs:language	language represents formal natural language identifiers, as defined by BCP 47[12].	
xs:integer	integer is derived from decimal by fixing the value of fractionDigits·to be 0 and disallowing the trailing decimal point. This results in the standard mathematical concept of the integer numbers. The ·value space· of integer is the infinite set {...,-2,-1,0,1,2,...}. The·base type of integer is decimal.	
xs:nonNegativeInteger	nonNegativeInteger has a lexical representation consisting of an optional sign followed by a non-empty finite-length sequence of decimal digits (#x30-#x39). If the sign is omitted, the positive sign ('+') is assumed. If the sign is present, it must be "+" except for lexical forms denoting zero, which may be preceded by a positive ('+') or a negative ('-') sign. For example: 1, 0, 12678967543233, +100000.	
xs:positiveInteger	positiveInteger is·derived·from nonNegativeInteger by setting the value of minInclusive·to be 1. This results in the standard mathematical concept of the positive integer numbers. The ·value space· of positiveInteger is the infinite set {1,2,...}. The base type·of positiveInteger is nonNegativeInteger.	
xs:unsignedLong	unsignedLong is derived· from nonNegativeInteger by setting the value of ·maxInclusive· to be 18446744073709551615. The base type·of unsignedLong is nonNegativeInteger.	
xs:unsignedInt	unsignedInt is ·derived·from unsignedLong by setting the value of·maxInclusive·to be 4294967295. The base type·of unsignedInt is unsignedLong.	
xs:unsignedShort	unsignedShort is ·derived· from unsignedInt by setting the value of maxInclusive· to be 65535. The·base type·of unsignedShort is unsignedInt.	
xs:dateTimeStamp	The dateTimeStamp datatype is derived·from dateTime by giving the value required to its explicitTimezone facet. The result is that all values of dateTimeStamp are required to have explicit time zone offsets and the datatype is totally ordered.	

Editor's Note: How to choose the local time zone at timestamp is FFS.

6.3.2. Additional Data Types used in oneM2M

This clause defines the data types that are defined by oneM2M for use in the oneM2M API. These data types are used in request/response parameters and attributes of resource types.

6.3.2.1. Simple Data Types

Table 6.3.2.1-1 describes oneM2M-specific simple data type definitions.

Table 6.3.2.1-1: oneM2M Simple Data Types

XSD type name	Type Name	Examples	Description
m2m:id	Generic ID	(TBD)	Used to represent generic IDs generated and used within oneM2M
m2m:nodeId	Node ID	(TBD)	Used for Node IDs. The constraints on this type are different from those on Generic IDs
m2m:requestId	Request ID	(TBD)	Used for Request IDs. This type includes the ID of the target CSE as well as a part that varies for each ID
m2m:acpType	ACP Type	(TBD)	Used to represent an AccessControlPolicy identifier. This can be either a URI or an opaque token
m2m:networkaccessIdentifier	Network Access Identifier	user@realm	The networkaccssIdentifier is a standard way of identifying users who request access to a network as specified at IETF RFC 4282 [18].
m2m:mgmtLink	Management Link	mynode/cmdhPolicyInstance1	m2m:id restricted to allowed types of instances of resource type <mgmtObj>
m2m:listOfM2MID	List of M2M identifiers		xs:list of elements of data type m2m:id
m2m:listOfEventCat	List of (applicable) Event Categories	(default bestEffort latest 1 10)	xs:list of elements of data type m2m:eventCat
m2m:listOfMinMax	List of Time Limits	(10 2560)	xs:list of two xs:long values defining min and max limits of time intervals in units of milliseconds (value -1 representing infinite time)
m2m:backOffParameters	List of Backoff Parameters	(100 100 2000)	Ordered sequence of 3 values of data type xs:nonNegativeInteger representing backoffTime, backoffTimeIncrement, maximumBackoffTime (in units of milliseconds)
m2m:listOfBoolean	List of allowed values in the boolean value space domain	“True“, "False“, or "True False“	Used to define the value space of the attribute “limitsDelAggregation” by imposing restrictions on the value space of xs:boolean

6.3.2.2. Enumerated Data Types

6.3.2.2.1. Introduction

The oneM2M Enumeration Types are defined as extention from ‘enumeration type’ which is defined in XML Schema definition [3]. The oneM2M Enumeration Types are based on <xs:integer>, and the numeric values are interpreted as specified in clause 6.3.2.2. Table 6.3.2.2.1-1 shows the example of Enumeration Type Definition for m2m:enumFooType.

Table 6.3.2.2.1-1: Example of oneM2M Enumeration Type Definition

Value	Interpretation	Note
1	Interpretation-1	
2	Interpretation-2	
3	Interpretation-3	
See Cluase x.x.x “title of cluase”		

The oneM2M Enumeration Type definition shall be implemented as part of CDT-enumeration-v1_0_0-<<date of publication>>.xsd. Figure 6.3.2.1-1 shows the example of XSD representation of 'm2m:enumFooType'.

```
<xs:simpleType name="enumFooType">
 <xs:restriction base="xs:integer">
  <xs:enumeration value="1"/>
```

Figure 6.3.2.2.1-1: Example of XSD version of oneM2M Enumeration Type

6.3.2.2.2. Enumeration Type Definitions

6.3.2.2.2.1. m2m:resourceType

Table 6.3.2.2.2.1-1: Interpretation of resourceType

Value	Interpretation	Note
1	accessControlPolicy	
2	AE	
3	container	
4	contentInstance	
5	CSEBase	
6	delivery	
7	eventConfig	
8	execInstance	
9	fanOutPoint	
10	Group	
11	localPolicy	
12	m2mServiceSubscription	
13	mgmtCmd	
14	mgmtObj	
15	Node	
16	nodeInfo	
17	pollingChannel	
18	remoteCSE	
19	Request	
20	Schedule	
21	statsCollect	
22	statsConfig	
23	Subscription	
See Clause 6.4.1 "Request message parameter data types"		

6.3.2.2.2.2. m2m:commanList

Used for supportedResourceType attribute of <CSEBase> resource.

Table 6.3.2.2.2.2-1: Interpretation of commaList

Value	Interpretation	Note
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
See Cluase 7.3.2 “Resource Type container”		

6.3.2.2.2.3. m2m:cseTypeID

Used for cseType attribute of <CSEBase> resource.

Table 6.3.2.2.2.3-1: Interpretation of cseTypeID

Value	Interpretation	Note
1	IN-CSE	
2	MN-CSE	
3	AEN-CSE	
See Cluase 7.3.3 “Resource Type remoteCSE”		

6.3.2.2.2.4. m2m:locationSource

Used for locationSource attribute of <locationPolicy> resource.

Table 6.3.2.2.2.4-1: Interpretation of locationSource

Value	Interpretation	Note
1	Network-based	
2	Device-based	
3	Sharing-based	
See Clauase 7.3.9 “Resource Type locationPolicy”		

6.3.2.2.2.5. m2m:eventCatType

Used for ec parameter in request and eventCat attribute of <delivery> resource and cmdh policy resource types.

Table 6.3.2.2.2.5-1: Interpretation of eventCatType

Value	Interpretation	Note
1	Default	
2	immediate	
3	bestEffort	
4	Latest	
See Clauase 7.3.10 “Resource Type delivery” and Annex D.12 “Resource cmdhPolicy”		

6.3.2.2.2.6. m2m:operation

Used for op parameter in request and operation attribute in <request> resource.

Table 6.3.2.2.2.6-1: Interpretation of operation

Value	Interpretation	Note
1	Create	
2	Retrieve	
3	Update	
4	Delete	
5	Notify	
See Clause 6.4.1 “Request message parameter data types”		

6.3.2.2.2.7. m2m:responseType

Used for rt parameter in request and operation attribute in <request> resource

Table 6.3.2.2.2.7-1: Interpretation of responseType

Value	Interpretation	Note
1	nonBlockingRequestSynch	
2	nonBlockingRequestAsynch	
3	blockingRequest	
See Clause 6.4.1 “Request message parameter data types”		

6.3.2.2.2.8. m2m:resultContent

Used for rc parameter in request.

Table 6.3.2.2.2.8-1: Interpretation of resultContent

Value	Interpretation	Note
0	Nothing	
1	attributes	
2	childResources	
3	cttribute+childResources	
4	originalResources	
See Clause 6.4.1 “Request message parameter data types”		

6.3.2.2.2.9. m2m:discResType

Table 6.3.2.2.2.9-1: Interpretation of discResType

Value	Interpretation	Note
1	hierarchical	
2	Non-hierarchical	
3	cseID+resourceID	
See Cluase 6.4.1 “Request message parameter data types”		

6.3.2.2.2.10. m2m:statusCode

Table 6.3.2.2.2.10-1: Interpretation of statusCode

Value	Interpretation	Note
1		
2		
3		
See Clause 6.6.3 " Current Response Status Codes"		

6.3.2.2.2.11. m2m:requestStatus

Used for requestStatus attribute in <request> resource.

Table 6.3.2.2.2.11-1: Interpretation of requestStatus

Value	Interpretation	Note
1		
2		
3		
See Clause 7.3.11"Resource Type request"		

6.3.2.2.2.12. m2m:operationResult

Used for operationResult attribute in <request> resource

Table 6.3.2.2.2.12-1: Interpretation of operationResult

Value	Interpretation	Note
1		
2		
3		
See Clause 7.3.11"Resource Type request"		

6.3.2.2.2.13. m2m:memberType

Used for memberType attribute in <member> resource.

Table 6.3.2.2.2.13-1: Interpretation of memberType

Value	Interpretation	Note
1	accessControlPolicy	
2	AE	
3	container	
4	contentInstance	
5	CSEBase	
6	delivery	
7	eventConfig	
8	execInstance	
9	fanOutPoint	
10	Group	
11	localPolicy	
12	m2mServiceSubscription	
13	mgmtCmd	
14	mgmtObj	
15	Node	
16	nodeInfo	
17	pollingChannel	
18	remoteCSE	
19	Request	
20	Schedule	
21	statsCollect	
22	statsConfig	
23	Subscription	
24	Mixed	
See Clause 7.3.12 “Resoruce Type group”		

6.3.2.2.2.14. m2m:consistencyStrategy

Used for consistencyStrategy attribute in <group> resource.

Table 6.3.2.2.2.14-1: Interpretation of consistencyStrategy

Value	Interpretation	Note
1		
2		
3		
See Clause 7.3.12 “Resoruce Type group”		

6.3.2.2.2.15. m2m:cmdType

Used for cmdType attribute in <mgmtCmd> resource.

Table 6.3.2.2.2.15-1: Interpretation of cmdType

Value	Interpretation	Note
1	RESET	
2	REBOOT	
3	UPLOAD	
4	DOWNLOAD	
5	SOFTWAREINSTALL	
6	SOFTWAREUNINSTALL	
See Clause 7.3.15 “Resource Type mgmtCmd”		

6.3.2.2.2.16. m2m:execReqArgsListType

Used for execReqArgsListType attribute in <mgmtCmd> and <execInstance resource.

Table 6.3.2.2.2.16-1: Interpretation of execReqArgsListType

Value	Interpretation	Note
1		
2		
3		
4		
5		
6		
See Clause 7.3.15 “Resource Type mgmtCmd” .		

6.3.2.2.2.17. m2m:execModeType

Used for execModeType attribute in <mgmtCmd> and <execInstance> resource.

Table 6.3.2.2.2.17-1: Interpretation of execModetType

Value	Interpretation	Note
1	IMMEDIATEONCE	
2	IMMEDIATEREPEAT	
3	RANDOMONCE	
4	RANDOMREPEAT	
See Clause 7.3.15 “Resource Type mgmtCmd” and Clause 7.3.16 “Resource Type execInstance”		

6.3.2.2.2.18. m2m:execStateType

Used for execStatusType attribute in <execInstance> resource.

Table 6.3.2.2.2.18-1: Interpretation of execStateType

Value	Interpretation	Note
1	INITIATED	
2	STARTED	
3	FINISHED	
4	CANCELLING	
5	CANCELLED	
See Clause 7.3.16 “Resource Type execInstance”		

6.3.2.2.2.19. m2m:pendingNotification

This is used for pendingNotification attribute in <subscription> resource.

Table 6.3.2.2.2.19-1: Interpretation of pendingNotification

Value	Interpretation	Note
1	sendLatest	
2	sendAllPending	
See Clause 7.3.7 “Resource Type subscription”		

6.3.2.2.2.20. m2m:notificationContentType

Table 6.3.2.2.2.20-1: Interpretation of notificationContentType

Value	Interpretation	Note
1	modifiedAttributes	
2	wholeResource	
3	referenceOnly	
See Clause 7.3.26 “Definition of Notification”		

6.3.2.2.2.21. m2m:resourceStatus

This is used for eventNotificationCriteria.

Table 6.3.2.2.2.21-1: Interpretation of resourceStatus

Value	Interpretation	Note
1	childCreated	
2	childDeleted	
3	updated	
4	deleted	
See Clause 7.3.26 “Definition of Notification”		

6.3.2.2.2.22. m2m:operationMonitor

This is used for eventNotificationCriteria.

Table 6.3.2.2.2.22-1: Interpretation of operationMonitor

Value	Interpretation	Note
1	Create	
2	Retrieve	
3	Update	
4	Delete	
See Clause 7.3.26 “Definition of Notification”		

6.3.2.2.2.23. m2m:attribute

Table 6.3.2.2.2.23-1: Interpretation of attribute

Value	Interpretation	Note
1		
2		
3		
4		

6.3.2.2.2.24. m2m:status

This is used for [software], [firmware] resource.

Table 6.3.2.2.2.24-1: Interpretation of status

Value	Interpretation	Note
1	Successful	
2	Failure	
3	In-Process	
See Clause D.2, D.3 firmware and software management		

6.3.2.2.2.25. m2m:batteryStatus

This is used for [battery] resource.

Table 6.3.2.2.2.25-1: Interpretation of batteryStatus

Value	Interpretation	Note
1	NORMAL	The battery is operating normally and not on power.
2	CHARGING	The battery is currently charging.
3	CHARGING-COMPLETE	The battery is fully charged and still on power.
4	DAMAGED	The battery has some problem.
5	LOW-BATTERY	The battery is low on charge.
6	NOT-INSTALLED	The battery is not installed.
7	UNKNOWN	The battery information is not available.
See Clause D.7 battery management		

6.3.2.2.2.26. m2m:mgmtDefinition

This is used for mgmtObj resource.

Table 6.3.2.2.2.26-1: Interpretation of mgmtDefinition

Value	Interpretation	Note
1	firmware	
2	software	
3	memory	
4	areaNwkInfo	
5	areaNwkDeviceInfo	
6	battery	
7	deviceInfo	
8	deviceCapability	
9	reboot	
10	eventLog	
11	cmdhPolicy	
12	activeCmdhPolicy	
13	cmdhDefaults	
14	cmdhDefEcValue	
15	cmdhEcDefParamValues	
16	cmdhLimits	
17	cmdhNetworkAccessRules	
18	cmdhNwAccessRules	
19	cmdhBuffer	
See Clause 7.3.13 mgmtObj		

6.3.2.2.2.27. m2m:logTypeId

Used for the *logTypeId* attribute of [eventLog] Management Resource.

Table 6.3.2.2.2.27-1: Interpretation of logTypeId

Value	Interpretation	Note
1	system	
2	security	
3	event	
4	trace	
5	panic	

6.3.2.2.2.28. m2m:logStatus

Used for the *logStatus* attribute of [eventLog] Management Resource.

Table 6.3.2.2.2.28-1: Interpretation of logStatus

Value	Interpretation	Note
1	Started	the logging activity is started
2	Stopped	the logging activity is stopped
3	Unknown	the current status of the logging activity is unknown.
4	NotPresent	the log data is not present and the logData attribute shall be ignored.
5	Error	error conditions for the logging activities, and the logging is stopped.

6.3.2.3. Complex Data Types

Table 6.3.2.3-1describes oneM2M-specific complex data type definitions.

Table 6.3.2.3-1: oneM2M Complex Data Types

XSD type name	Element	Element Data Type	Description
m2m:filterCriteria	createdBefore	xs:dateTime	Used for fc (filter criteria) parameter in a request.
	createdAfter	xs:dateTime	
	modifiedSince	xs:dateTime	
	unmodifiedSince	xs:dateTime	
	stateTagSmaller	xs:positiveInteger	
	stateTagBigger	xs:nonNegativeInteger	
	expireBefore	xs:dateTime	
	expireAfter	xs:dateTime	
	labels	xs:token	
	resourceType	m2m:resourceType	
	sizeAbove	xs:nonNegativeInteger	
	sizeBelow	xs:positiveInteger	
	contentType	m2m:typeOfContent	
	limit	xs:positiveInteger	
	attribute	m2m:attribute	
	filterUsage	m2m:filterUsage	
m2m:eventCat	eventCatType	m2m:eventCatType	Used for ec parameter in request and eventCat attribute of <delivery> resource.
	eventCatNo	xs:nonNegativeInteger	
m2m:deliveryMetaData	TBD		Used for deliveryMetaData attribute in <delivery> resource.
m2m:aggregatedRequest	TBD		Used for aggregatedRequest attribute in <delivery> resource.
m2m:metaInformation	TBD		Used for metaInformation attribute in <request> resource.
m2m:content	TBD		Used for cn parameter in request/response and content attribute in <request> resource.
m2m:batchNotify	number	xs:nonNegativeInteger	Used for batchNotify attribute in <subscription> resource.
	duration	xs:duration	
m2m:eventNotificationCriteria	createdBefore	xs:dateTime	Used for eventNotificationCriteria of a <subscription> resource..
	createdAfter	xs:dateTime	
	modifiedSince	xs:dateTime	
	unmodifiedSince	xs:dateTime	
	stateTagSmaller	xs:positiveInteger	
	stateTagBigger	xs:nonNegativeInteger	
	expireBefore	xs:dateTime	
	expireAfter	xs:dateTime	
	sizeAbove	xs:nonNegativeInteger	
	sizeBelow	xs:positiveInteger	
	resourceStatus	m2m:resourceStatus	
	operationMonitor	m2m:operation	
	Attribute	m2m:attribute	
m2m:aggregatedNotification	singleNotification	list of m2m:singleNotification	
m2m:singleNotification	notificationEvent	list of m2m:notificationEvent	
	verificationRequest	xs:boolean	
	subscriptionDeletion	xs:boolean	
	subscriptionReference	xs:anyURI	
	creator	m2m:id	
	notificationForwardingURI	xs:anyURI	
m2m:notificationEvent	all resource specific attributes of subscribable resource types.	data type of each attribute (This is defined in the sub-clauses of clause 7.3)	used to indicate the date and time when the event occurs
	resourceStatus	m2m:resourceStatus	
	operationMonitor	m2m:operationMonitor	

m2m:operationMonitor	operation	m2m:operation	
	originator	m2m:id	
m2m: externalID	accessId	m2m:networkaccessIdentifier	The identifier of the node for the underlying network provider. In 3GPP case, the accessId is mapped to External Identifier as specified in TS 23.003 [17].
	MSISDN	xs: string	The identifier of the node as specified in TS 23.003 [17],
m2m:actionStatus	action	xs:anyURI	Reference to the action (represented by a resource attribute) being performed
	status	m2m:status	Indicates the status of the operation is successful, failure or in process. See Table 6.3.2.2 1

NOTE: List of subscribable resource types are <accessControlPolicy>, <CSEBase>, <remoteCSE>, <AE>, <container>, <contentInstance>, <schedule>, <locationPolicy>, <delivery>, <request>, <group>, <mgmtObj>, <mgmtCmd>, <execInstance>, <node>, <m2mServiceSubscription>, <nodeInfo>, <statsConfig>, <eventConfig> and <statsCollect>

6.3.3. Resource common attributes

Resource common attributes are specified in oneM2M TS-0001 [6]. The type and values shall be supported according to the description given in table 6.3.3-1.

Table 6.3.3-1: Resource Common Attributes

Attribute Name	Data Type	Default	Value restrictions	Notes
accessControlPolicyIDs	m2m:acpType	Policy is inherited from the parent resource		
parentID	m2m:nhURI	Not applicable	Must be a non-hierarchical URI	
creationTime	xs:dateTime	Not applicable		
expirationTime	xs:dateTime	Absence of the attribute means that the resource has no explicit expirationTime		
lastModifiedTime	xs:dateTime	Not applicable		
resourceType	m2m:resourceType	In the absence of this attribute, the resource type can be inferred from the root element		
labels	list of xs:token	Absence of this attribute means there are no labels		
stateTag	xs:nonNegativeInteger	Not applicable		
link	xs:anyURI	Not applicable		
announceTo	list of xs:anyURI	Not applicable		
announcedAttribute	list of xs:token	Not applicable		

6.4. Message parameter data types

This clause specifies the message parameter data types for messages, which cross the Mca and Mcc reference points.

Editor's Note: This is per ARC TS, what about Mcc'?

6.4.1. Request message parameter data types

The data Types of request message parameters are specified in this clause.

Detailed request message parameter descriptions and usage can be found in clause 8.1 of the Architecture TS [6].

Table 6.4.1-1: Data Types for Request message parameters

Request message parameter	Short Name	Data Type	Default?	Comment
Content	cn	m2m:		
Delivery Aggregation	da	xs:boolean	False	If da is not included in the Request and the defaultDelAggregation attribute in the cmdhEcDefParamValues resource is not present, then this default value is used.
Discovery Result Type	dr	m2m:discResType	"hierachicalURI"	
Event Category	ec	m2m:eventCatType	N/A	
Filter Criteria	fc	m2m:filterCriteria		
From	fr	xs:anyURI	N/A	
Group Request Identifier	gid	xs:string	N/A	
Name	nm	xs:string	N/A	
Operation	op	m2m:char	N/A	
Operational Execution Timestamp	oet	m2m:timestamp		
Originating Timestamp	ort	m2m:timestamp		
Request Expiration Timestamp	ret	m2m:timestamp		"Result ExpirationTimestamp" shall be later than "Request Expiration Timestamp"
Request Identifier	ri	xs:string	N/A	
Resource Type	rqt	m2m:resourceType	N/A	
Response Type	rst	m2m:responseType	N/A	
Result Content	rc	m2m:resultContent	"attribute"	
Result Expiration Timestamp	ret	m2m:timestamp		
Result Persistence	rp	xs:duration	"0"	
Role	ro	xs:string	N/A	Editor's Note: Missing procedure in ARC TS
To	to	xs:anyURI	N/A	See ARC TS 9.3.1

Editor's Note: This list is from ARC TS 0.8.0 and may need updating. Should defaults be included?

Table 6.4.1-2: Data Types for *filterCriteria* parameter

Condition tag	Data Type	Default?	
createdBefore	xs:dateTime		
createdAfter	xs:dateTime		
modifiedSince	xs:dateTime		
unmodifiedSince	xs:dateTime		
stateTagSmaller	xs:integer		(Same type as common stateTag attribute)
stateTagBigger	xs:integer		(Same type as common stateTag attribute)
expireBefore	xs:dateTime		
expireAfter	xs:dateTime		
Labels	m2m:labels		
resourceType	m2m:resourceType		
sizeAbove	xs:int		(Same type as contentSize attribute of the <contentInstance> resource)
sizeBelow	xs:int		(Same type as contentSize attribute of the <contentInstance> resource)
contentType	xs:string		
Limit			
attribute	list of xs:string		
filterUsage	xs:string		
NOTE1:			

6.4.2. Response message parameter data types

The data types of response message parameters are specified in this clause.

Detailed response message parameter descriptions and usage can be found in clause 8.1 of the Architecture TS [6].

Table 6.4.2-1: Data Types for Response message parameters

Request message parameter	Short Name	Data Type		Comment
Content	cn	m2m:content		
From	fr	m2m:id		
Originating Timestamp	ort	m2m:timestamp		
Request Identifier	ri	xs:string		
Result Expiration Timestamp	rset	m2m:timestamp		
To	to	xs:anyURI		See ARC TS 9.3.1

Editor's Note: This list is from ARC TS 0.8.0 (less **status codes**) and may need updating.

6.4.3. resourceData type

6.4.3.1. 6.4.4.1 Description

This type defines data structure used as MIME-Type 'application/vnd.onem2m-resource-data +xml'. It is included in CDT-mimeTypes-v1_0_0-<date>.xsd.

6.4.3.2. Reference

See Annex F

6.4.3.3. Usage Example

```
<?xml version="1.0"?>
<m2m:resourceData xmlns:m2m="http://www.onem2m.org/xml/protocols">
 <m2m:resourceTypeId>containerType</m2m:resourceTypeId>
 <m2m:container>
  <m2m:name>container1</m2m:name>
 </m2m:container>
</m2m:resourceData>
```

6.5. Resource Data Types

Each oneM2M Resource Data Type is defined using XML Schema (XSD), and supplied as a separate XSD document. This XML Schema defines the attributes of the Resource in accordance with oneM2M TS-0001 [6]. It represents an entire resource, in other words if a requestor retrieves an entire resource in XML format, the XML that is returned shall be valid with respect to the schema for that resource. Note that the payload of a Create or Update request primitive does not necessarily have to be valid according to the schema, as this payload is not required to contain values for all the resource attributes. In particular a resource might contain mandatory read-only primitives, and these would not appear in a Create or Update request.

Each Resource Type will be defined in a separate XSD file. The name of that file should be prefixed with 'CDT-' and followed by the resource type name, version of the Core Protocol TS, and date of update.

Table 6.5-1: oneM2M Structured Data Types

Data Type ID	File Name	Where defined	Notes
Simple Types	CDT-simpleType-v1_0-<date>.xsd	6.3.1	
CommonTypes	CDT-commonTypes-v1_0_0-<date>.xsd		

6.5.1. regularResourceType

6.5.1.1. Description

This type definition includes the common attributes used by the majority of M2M resources. It is included in CDT-commonTypes-v1_0_0-<date>.xsd.

6.5.1.2. Reference

See clause 9.6.1 of [6].

6.5.1.3. Usage Example

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.w3.org/2001/XMLSchema"
   targetNamespace=http://www.onem2m.org/xml/protocols
   xmlns:m2m=http://www.onem2m.org/xml/protocols
   elementFormDefault="unqualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:include schemaLocation="common_types-v1_0_0-20140409.xsd" />

   <xs:element name="container">
      <xs:complexType>
         <xs:complexContent>
           <!--   Inherit Common Attributes from regularResourceType     -->
            <xs:extension base="m2m:regularResourceType">
```

```
            <xs:sequence>
                <!--          Resource Specific Attributes           -->
                <xs:element name="maxNrOfInstances" type="xs:nonNegativeInteger" minOccurs="0" />
                <xs:element name="maxByteSize" type="xs:nonNegativeInteger" minOccurs="0" />
                <xs:element name="maxInstanceAge" type="xs:nonNegativeInteger"   minOccurs="0" />
                <xs:element name="currentNrOfInstances" type="xs:nonNegativeInteger" />
                <xs:element name="currentByteSize" type="xs:nonNegativeInteger" />
                <xs:element name="latest" type="xs:anyURI"  minOccurs="0" />
                <xs:element name="locationID" type="xs:anyURI" minOccurs="0" />
                <xs:element name="ontologyRef" type="xs:anyURI"  minOccurs="0" />
                <!--        Child Resources                                -->
                <xs:element name="childResource" type="m2m:childResourceType"  minOccurs="0"
                   maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
   </xs:element>
</xs:schema>
```

6.5.2. announceableResourceType

6.5.2.1. Description

This type definition includes the common attributes used by M2M resource types that are capable of being announced. In addition to the attributes of a regularResource, It includes (as optional) the common attributes that are used by the announcement mechanism.

6.5.2.2. Reference

See clause 9.6.1 of [6].

6.5.2.3. Usage Example

<Text>

6.5.3. subordinateResourceType

6.5.3.1. Description

This type definition includes the common attributes used by M2M resource types that are subordinate children over other resource types (for example the Instance child of a container). It excludes attributes like expirationTime, which are not defined for such resources.

6.5.3.2. Reference

See clause 9.6.1 of [6].

6.5.3.3. Usage Example

<Text>

Editor's Note: This is example of description for each Common Data Type, and remove this sub clause when actual entries are added.

6.6. Response Status Codes

6.6.1. Introduction

The present clause specifies the assignment of oneM2M Response Status Codes (RSC), which is returned as content of Response primitive when the Response type is other than Successful Response.

6.6.2. RSC Framework Overview

Unsuccessful RSCs are categorised as one of three classes:

- Originator Error:
  The 4xxx class of status code is intended for cases in which the Request primitive sent from Originator has some problem.
- Receiver Error:
  The 5xxx class of status code is intended for cases in which the Receiver could not process the Request by some reason.
- External System Error:
- The 6xxx class of status code is intended for cases in which the Receiver could not process the Request due to error on calling external system thru Mcn reference point.

The term "native" is used to indicate that the RSC source originated as a result of a resource operation validation procedure before protocol mapping is applied.

Table 6.6.2-1 RSC Framework Overview

Response Class	RSC Source	Numeric Code Range
Un-successful	Current • native oneM2M codes • mapped security codes • mapped device management codes • mapped transport codes • mapped interworking codes • mapped vendor extensions	2000 - bbbb
	Reserved for future • native oneM2M codes • mapped security codes • mapped device management codes • mapped transport codes • mapped Interworking codes • mapped vendor extensions	bbbb+1 - 7999
Acknowledgements	Current • native oneM2M codes • mapped security codes • mapped device management codes • mapped transport codes • mapped interworking codes • mapped vendor extensions	8000 - cccc
	Reserved for future • native oneM2M codes • mapped security codes • mapped device management codes • mapped transport codes • mapped interworking codes • mapped vendor extensions	cccc+1 - 9999

Editor's Note: Other terms rather than "current" and "reserved for future" may be preferred e.g. first release and subsequent release.

In Table X.3-1, 0<= aaaa <= 1999, 2000 <= bbbb <= 7999, 8000 <= cccc <= 9999

6.6.3. Current Response Status Codes

The tables in the following clauses specify the RSCs for oneM2M releases. Each RSC includes: a response status description, a numeric code, arguments, and applicable APIs (indicated by reference points). The response status description shall be unique within the set of response status descriptions for a release. Response status descriptions may be re-used among RSCs associated with different releases.

6.6.3.1. Successful Responses

Successful Response can be interpreted from native status code of protocol in use.

6.6.3.2. Un-successful Response Codes

Table 6.6.3.2-1 specify the RSCs for un-successful responses for each release.

Table 6.6.3.2-1: RSCs for Un-successful Responses for Release 1

Description	Numeric Code	StatusDescription	Applicability			
			Mca	Mcc	Mcn	Mcc'
Location info not authorized	6001				X	
Unsupported resource	5001	Resource URI	X	X		
Unsupported attribute	5002	Attribute URI	X	X		
Target not reachable	5003	Resource URI	X	X		
Cannot forward, other reason TBD	5004	Resource URI	X	X		
No privilege	5005	Resource URI	X	X		
Already exists	5006	Resource URI	X	X		
Create error - missing mandatory parameter	4007	Missing parameter	X	X		
Retrieve error - does not exist	5009	Target URI	X	X		
Update error - unacceptable contents	4012	Attribute URI	X	X		
Create delivery - not able to take on responsibility	5015		X	X		
group request identifier exists	6002		X	X		
External object not reachable	6003	mgmtObj URI	X	X		
Create mgmtObj - memory shortage	6004	CSEBase URI	X	X		
External object not found	6005	MgmtObj URI	X	X		
Cancel execInstance - not cancellable	6006	execInstance URI	X	X		
Cancel execInstance - already complete	6007	execInstance URI	X	X		
Delete execInstance - not cancellable	6008	execInstance URI	X	X		
Delete execInstance - already complete	6009	execInstance URI	X	X		
Retrieve CSEBase - format error	5035	CSEBase URI	X	X		
CMDH rules -non compliant	5036					
Target is not subscribable	5037					
Cannot initiate subscription verification	5038					
Subscription verification failed – Originator ID	5039	Originator ID				
Subscription verification failed - creator		creator				
Max number of member exceeded	6010					
Member type inconsistent	6011					
Management session cannot be established	6022		X			
Management session establishment timeout	6003		X			
Create mgmtCmd – invalid cmdType	6003		X	X		
Create mgmtCmd – invalid arguments	6004		X	X		
Create mgmtCmd – insufficient arguments	6005		X	X		
MgmtCmd – conversion error	6006		X	X		
Delete mgmtCmd- execInstance cancellation error	6007		X	X		
Cancel execInstance – cancellation error	6008		X			
Delete execInstance – cancellation failed	6009		X			

Editor's Note: The contents of this table are exemplary and are expected to be revised. In particular ARC discussions regarding UPDATE and attributes may impact argument1. Some of the exemplary RSCs may be equivalent and can be collapsed to fewer RSCs.

Editor's Note: Since more RSCs may be added until the first release, the Numeric Codes will be finalized just before the first release.

6.6.3.3. Acknowledgement Responses

Table 6.6.3.3-1 specify the RSCs for acknowledgement responses for each release.

Table 6.6.3.3-1: RSCs for Acknowledgement Responses for Release 1

Description	Numeric Code	StatusDescription	Applicability			
			Mca	Mcc	Mcn	Mcc'
	1001					
	1002					
	1003					
	1004					
	1005					
	1006					

Editor's Note: Since more RSCs may be added until the first release, the Numeric Codes will be finalized just before the first release.

6.7. MIME Types for oneM2M protocols

The present sub-clause explains the use of oneM2M specific MIME-Type which shall be used in one of protocol bindings.

Those MIME-Types are registered with the prefix 'application/vnd.onem2m-' on IANA registry.

Editor's Note: actual request of assignment in IANA is FFS.

6.7.1. application/vnd.onem2m-resource-data+xml

This data type is used for XML-based data to represent the oneM2M resource.

Top Level Element	Child Element Name	Element Type	Note
resourceData	resourceTypeId	xs:string	
	(Choice of defined m2m resource Types)		Resource type which is included.

6.7.2. application/vnd.onem2m-response-status+xml

This data type is used for XML-based data m2m:responseStatus to represent the response status.

See clause 6.3.2.3 for XSD definition of m2m:responseStatus .

6.7.3. application/vnd.onem2m-partial-updates+xml

This data type is used for XML-based data m2m:partialUpdates to represent the instruction to perform partial update.

See clause 6.3.2.3 for XSD definition of m2m:partialUpdates

7 oneM2M Resource Types

7.1. Introduction

<Text>

7.2. Prerequisites

7.2.1. Primitive format and procedure outlines

7.2.1.1. Primitive format

7.2.1.1.1. Request primitive format

Table 7.2.1.1-1 summarizes the primitive parametersattributes for Request primitive, showing any differences as applied to C, R, U, D or N operations. "M" indicates mandatory, "O" indicates optional, "NP" indicates not present, "C" indicates conditional.

Table 7.2.1.1-1: Request Primitive Parameters

Primitive Parameter	CREATE		RETRIEVE		UPDATE		DELETE		NOTIFY	
	M/O		M/O		M/O		M/O		M/O	
	Mca	Mcc	Mca	Mcc	Mca	Mcc	Mca	Mcc	Mca	Mcc
primitiveType (=REQUEST)	M	M	M	M	M	M	M	M	M	M
Operation	M	M	M	M	M	M	M	M	M	M
To	M	M	M	M	M	M	M	M	M	M
From	M	M	M	M	M	M	M	M	M	M
Request Identifier	M	M	M	M	M	M	M	M	M	M
Resource Type	M	M	NP	NP	NP	NP	NP	NP	NP	NP
Name	C	C	NP	NP	NP	NP	NP	NP	NP	NP
Content	C	C	C	C	C	C	NP	NP	M	M
Original Timestamp	O	O	O	O	O	O	O	O	O	O
Request Expiration Timestamp	C	C	C	C	C	C	C	C	C	C
Result Expiration Time	C	C	C	C	C	C	C	C	C	C
Operation Execution Time	C	C	C	C	C	C	C	C	C	C
Response Type	C	C	C	C	C	C	C	C	C	C
Result Persistence	C	C	C	C	C	C	C	C	NP	NP
Result Content	C	C	C	C	C	C	C	C	NP	NP
Event Category	C	C	C	C	C	C	C	C	C	C
Delivery Aggregation	C	C	C	C	C	C	C	C	C	C
Group Request Identifier	C	C	C	C	C	C	C	C	C	C
Filter Criteria	NP	NP	C	C	NP	NP	NP	NP	NP	NP
Discovery Result Type	NP	NP	C	C	NP	NP	NP	NP	NP	NP

Editor's Note: This is an initial list of primitive parameters. Parameter name, mandatory/optional or description for each parameter is FFS.

7.2.1.1.2. Response primitive format

Table 7.2.1.1-2 summarizes the primitive parameters for Response primitive, showing any differences as applied to C, R, U, D or N operations and unsuccessful operations. "M" indicates mandatory, "O" indicates optional, "NP" indicates not present, "C" indicates conditional.

Table 7.2.1.1-2: Response Primitive Parameters (1/2)

Primitive parameter	Ack		Successful CREATE		Successful RETRIEVE		Successful UPDATE	
	M/O		M/O		M/O		M/O	
	Mca	Mcc	Mca	Mcc	Mca	Mcc	Mca	Mcc
primitiveType (=RESPONSE)	M	M	M	M	M	M	M	M
Response Code	M	M	M	M	M	M	M	M
Request Identifier	M	M	M	M	M	M	M	M
Content	NP	NP	C	C	M	M	C	C
To	C	C	C	C	C	C	C	C
From	C	C	C	C	C	C	C	C
Originating Timestamp	O	O	O	O	O	O	O	O
Result Expiration Timestamp	C	C	C	C	C	C	C	C
Pending Requests	NP	NP	NP	NP	C	C	NP	NP

Table 7.2.1.1-3 : Response Primitive Parameters (2/2)

Primitive parameter	Successful DELETE		Successful NOTIFY		Unsuccessful	
	M/O		M/O		M/O	
	M_{ca}	M_{cc}	M_{ca}	M_{cc}	M_{ca}	M_{cc}
primitiveType (=RESPONSE)	M	M	M	M	M	M
Response Status Code	M	M	M	M	M	M
Request Identifier	M	M	M	M	M	M
Content	C	C	NP	NP	C	C
To	C	C	C	C	C	C
From	C	C	C	C	C	C
Originating Timestamp	O	O	O	O	O	O
Result Expiration Timestamp	C	C	NP	NP	C	C

Editor's Note: This is an initial list of primitive parameters. Parameter name, mandatory/optional or description for each parameter is FFS.

7.2.1.2. Description of Generic Procedures

7.2.1.2.1. Generic Resource Request Procedure for Originator

A generic resource Request procedure shall be comprised of the following actions. Additional actions specific to individual procedures are listed in the respective sections by referencing these actions and providing additional steps.The Originator shall execute the following steps in order:

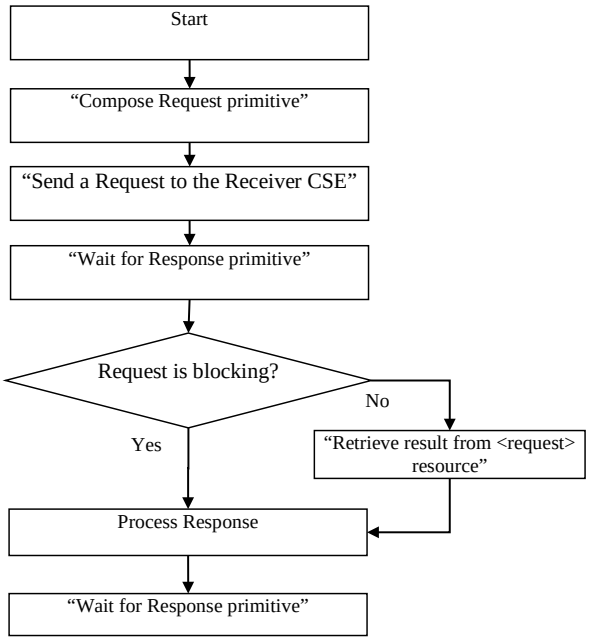


Figure 7.2.1.2.1-1: Generic procedure of Originator

Orig-1.0 "Compose Request primitive": Please refer to clause 7.2.2 for details.

Orig-2.0 "Send a Request to the receiver CSE": Please refer to clause 7.2.2 for details.

Orig-3.0 "Wait for Response primitive": Please refer to clause 7.2.2 for details. If the Originator is using blocking communication, this step should be the last step of the Originator.

Orig-4.0 "Request is blocking?": This step shall be operated after getting the Response primitive from step Oring-3.0 "Wait for Response primitive". In this step, the Originator checks whether the request was blocking. If the request was blocking, it goes to step Orig-6.0 "Process Response" (TRUE branch). If the request was non-blocking, it goes to step Orig-5.0 "Retrieve result from the <request> resource" (FALSE branch)..

Orig-5.0 "Retrieve result from the <request> resource": See clause 7.2.2.1.4 for details.

Orig-6.0 "Process Response": the Originator processes the response.

7.2.1.2.2. Generic Request Procedure for Receiver

The Receiver shall execute the following steps in order. In case of error in any of the steps below, the Receiver shall execute "Create an unsuccessful Response" (refer to clause 7.2.2.3.12 for details) and then "Send Response primitive" (refer to clause 7.2.2.2.2 for details). The corresponding Response Code shall be included in the Response primitive.

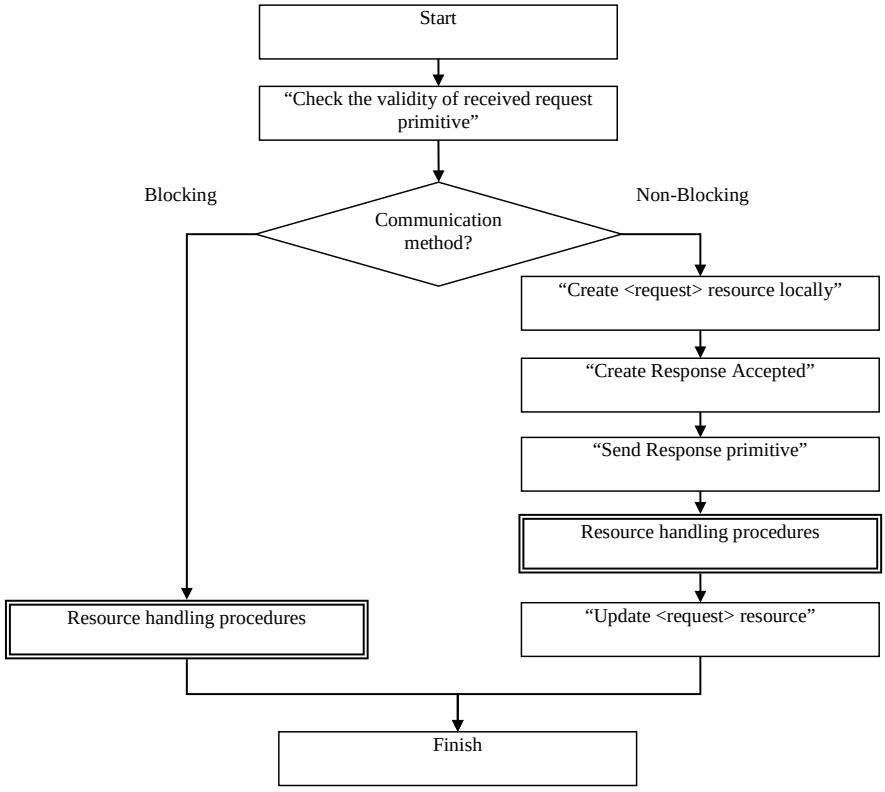


Figure 7.2.1.2.2-1: Generic procedure of Receiver

Recv-1.0 "Check the validity of received request primitive": See clause 7.2.2 for details.

Recv-2.0 "Communication method?": The Receiver CSE checks whether a received request is blocking or non-blocking by using ***rt*** parameter (see detail inclause 8.1.2 in TS-0001 [6]). If the request is blocking, it goes to step Recv-6.0 "Resource handling procedure" (Blocking branch). If the request is non-blocking, it goes to step Recv-3.0 "Create <request> resource locally" (Non-blocking branch).

Recv-3.0 "Create <request> resource locally": Please refer to clause 7.2.2.2.4 for details.

Recv-4.0 "Create Response Accepted": Please refer to clause 7.2.2.2.2 for details.

Recv-5.0 "Send Response Primitive": Please refer to clause7.2.2.2.2 for details.

Recv-6.0 "Resource handling procedure": Please refer to Figure 7.2.1.2.2-1 for details.

Recv-7.0 "Update <request> resource": Please refer to clause 7.2.2.2.7 for details. This step is only valid when the request is non-blocking.

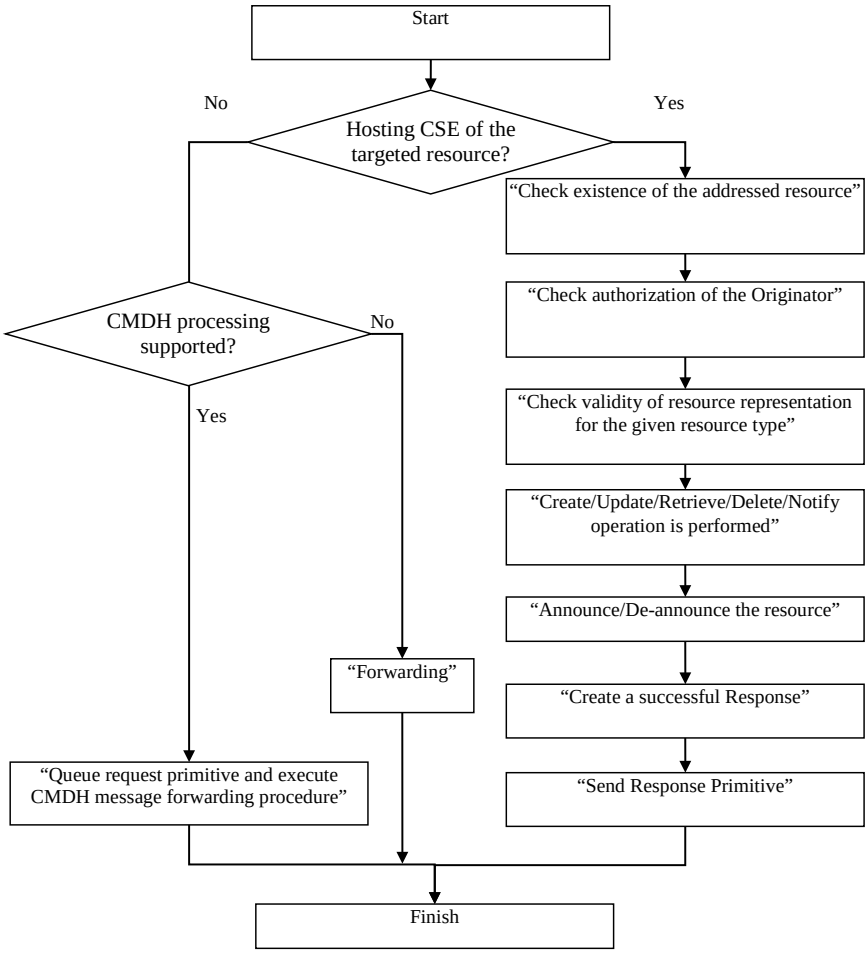


Figure 7.2.1.2.2-2: Resource handling procedure

The above figure describes the generic procedure to resource handling procedures.

Recv-6.1 "Hosting CSE of the targeted resource?": The step checks if the receiver is a Transit CSE or the Hosting CSE of the received Request by examining the to parameter of the Request primitive. If the receiver hosts the resource that the address in the to parameter represents, the receiver is the Hosting CSE (goes to Recv-6.2"Check existence of the addressed resource", Yes branch). Otherwise, the receiver is the Transit CSE (goes to Recv-6.9 "Queue request primitive and execute CMDH message forwarding procedure", No branch).

Recv-6.2 "Check existence of the addressed resource": Please refer to clause 7.2.2.3.1 for details.

Recv-6.3 "Check authorization of the Originator" : Please refer to clause (TBD) for details.

Recv-6.4 "Check validity of resource representation for the given resource type" : Please refer to clause 7.2.2.3.2 and clause 7.2.2.3.3 for details. Notify is not applicable for this step.

Recv-6.5 “Create/Update/Retrieve/Delete/Notify operation is performed” : The step represents five common operations which are “Create the resource”, “Retrieve the resource”, “Update the resource”, “Delete the resource” and “Notify re-targeting”. The procedure specified in the clause 7.2.2.3.4 Create the resource, 7.2.2.3.6 Update the resource, 7.2.2.3.6 Retrieve the resource, 7.2.2.3.7 Delete the resource or 7.2.2.3.1 Notify re-targeting is performed for the Create, Update, Retrieve, Delete, or Notify operation respectively.

Recv-6.6 “Announce/De-announce the resource”: The step represents two common operations which are “Announce the resource” and “De-announce the resource”. Please refer to clause 7.2.2.3.9 and clause for details. Notify is not applicable for this step.

Recv-6.7 “Create a successful Response”: Please refer to clause 7.2.2.3.11 for details.

Recv-6.8 “Send Response Primitive”: Please refer to clause 7.2.2.2.2 for details. If the Receiver is Hosting CSE, after this step, the procedure is terminated.

Recv-6.9 If CMDH processing is supported, carry out “Queue request primitive and execute CMDH message forwarding procedure”: The Receiver CSE shall queue the received request primitive and execute the “CMDH message forwarding procedure”. Please refer to Annex H.2.4. for details. If CMDH processing is not supported, carry out message forwarding as defined in clause 7.2.2.2.8.

Editor’s Note: These procedures and steps are FFS. Steps are pointer to the common operation section in TS-0004

7.2.2. Common operations

コメントの追加 **[SF3]:** PRO-2014-0261R01 intends to replace all following sub-clauses ?

7.2.2.1. Originator Actions

7.2.2.1.1. Compose Request primitive

The originator shall compose a Request message that shall be mapped to a specific protocol.

The Request shall include the "fr" and "to" to indicate the identifier of the originator of the request and the targeted receiver of the request.

The Request shall include the other attributes in case needed depend on the resource the request is addressing.

When including a resource representation in the request indication for create and update, the originator shall take into account the validation rules as specified in "Check validity for resource representation for create" and "Check validity for resource representation for update" respectively.

EXAMPLE: Any attributes marked with NP shall not be present in the resource representation for the corresponding request indication.

7.2.2.1.2. Send a Request to the Receiver CSE

The originator shall determine the receiver CSE.

The receiver of the Request shall be the registrar CSE of the originator in case the originator is not IN-CSE.

If the originator is the IN-CSE, the receiver of the Request shall be the CSE whose identifier is the prefix of the "to" attribute of the Request.

If this results in no matching CSE, then the request is rejected with a STATUS_NOT_FOUND.

If results in multiple CSEs, the request is rejected with a status STATUS_INTERNAL_SERVER_ERROR, since this shall never happen.

7.2.2.1.3. Wait for Response primitive

The originator shall wait for the Response primitive from the receiver that corresponds to the Request primitive that was sent by the originator. Correlation between the Request and the corresponding Response is handled by the transport layer.

If no Response primitive is received within a certain time, specified by server policy and/or by the underlying transport technology, this shall be handled as if a Response primitive with a statusCode STATUS_REQUEST_TIMEOUT was received.

7.2.2.1.4. Retrieve the <request> resource

When the Originator needs to retrieve information about an associated previously issued non-blocking request, the Originator shall request to RETRIEVE the attributes of the <request> resource. The Originator shall compose Request primitive with the following paramters and send the Request to the Receiver CSE. See clause 7.2.2.1.1 and 7.2.2.1.2.

NOTE: The Originator may include optional parameters described in clause 8.1.2 of [6].

Table 7.2.2.1.4-1: Request parameter settings

Parameter Name	Value
primitiveType	REQUEST
Operation (op)	Retrieve
To (to)	This shall be set to the URI of the <request> resource received in the response (acknowledgement) to the previously issued non-blocking request.
From (fr)	Id of the Originator
Request Identifier (ri)	The identifier of this request message.
Content (cn)	Optionally includes the name of attributes that needs to be retrieved.

7.2.2.2. Reciever CSE actions

7.2.2.2.1. Check the validity of received request primitive

The validity checking of the message carrying the received request primitive is specified by the protocol mapping TSes (HTTP binding, CoAP binding and MQTT binding). The received resource representation (e.g. in plain XML, binary XML or JSON) shall be validated against the provided schema definitions.

If a received request needs to be forwarded to another CSE and if CMDH processing is supported, then in addition the “CMDH message validation procedure” defined in Annex H.2.3. shall be carried out.

If the message is not valid, the request shall be rejected with a "STATUS_BAD_REQUEST".

7.2.2.2.2. Create Response Accepted

Editor’s note: In case of non-blocking communication, the receiver of the request is not be able to return the result in a short time. Instead of holding the connection, the receiver decides to first acknowledge the Request. In order to do this, the receiver needs to create a Response just to inform the Originator, Request accepted. The detailed procedure is to be contributed.

7.2.2.2.3. Send Response primitive

A Response primitive shall be sent back to the originator.

7.2.2.2.4. Create <request> resource locally

Creation of a <request> resource can only be done on a Receiver CSE implicitly. When the Receiver CSE receives a request for targeting any other resource type or requesting a notification in non-blocking mode , i.e. the rt parameter of the request is set to either ‘nonBlockingRequestSynch’ or ‘nonBlockingRequestAsynch’, and if the Receiver CSE supports the <request> resource type as indicated by the ‘supportedResourceType’ attribute of the <CSEBase> resource, the Receiver CSE shall create an instance of <request> resource based on the following steps. If the Receiver CSE does not support the <request> resource type, the request shall be rejected with a “nonBlockingRequest not supported”. The Receiver CSE of a non-blocking request is the Hosting CSE for the <request> resource that shall be associated with the non-blocking request.

1) Assign a value to the common attributes of <request> resource according to the following table:

Table 7.2.2.2.4-1: Common attributes settings for <request> resource

Attribute Name	Value
resourceType	request
resourceID	Hosting CSE shall assign a value to this attribute.
expirationTime	The value of the expirationTime shall be chosen dependent on the ***rqet***, ***rset***, ***oet*** and ***rp*** parameters associated with the original request. If the value consistent with the ***rqet***, ***rset***, ***oet*** and ***rp*** parameters is too long, the Hosting CSE shall reject the request.
parentID	The parent resource of a <request> resource shall be the <CSEBase> resource of the Hosting CSE.
creationTime	Date/time of creation of this resource.
lastModifiedTime	Date/time which is equal to the creationTime.
accessControlPolicyIDs	Populate with one ID of an <accessControlPolicy> that contains the following: In the 'privileges' attribute • Allow RUD operations to the Hosting CSE • Allow RD operations to the Originator, i.e. the value of the parameter ***fr*** in the associated non-blocking request In the 'selfPrivileges' attribute 1. Allow U operations the parent <accessControlPolicy> resource to the Originator, i.e. the value of the parameter ***fr*** in the associated non-blocking request
labels	Originator ID
stateTag	0

Editor's Note: Use case of giving selfPrivileges to the Originator is not clear, This needs to be clarified in WG2.

2) Assign a value to the resource-specific attributes of <request> resource accordinf to the following table:

Table 7.2.2.2.4-2: Resource-specific attributes settings for <request> resource

Attribute Name	Value
operation	The value of the parameter ***op*** in the associated non-blocking request.
target	The value of the parameter ***to*** in the associated non-blocking request.
originator	The value of the parameter ***fr*** in the associated non-blocking request.
requestID	The value of the parameter ***ri*** in the associated non-blocking request.
metaInformation	The content of this attribute is set to information in optional parameters described in clause 8.1.2 of [6] given in the associated non-blocking request.
content	The value of the parameter ***cn***, if any, in the associated non-blocking request.
requestStatus	LOCALLY_ACCEPTED
operationResult	Empty

Editor's Note: Possible values of the requestStatus need to be defined in the data type section.

7.2.2.2.5. Create a Successful Response (Acknoledgement)

The Receiver CSE shall create a Response primitive. The Receiver CSE shall include the following parameters in the Response primitive.

Table 7.2.2.2.5-1: Response parameter settings

Parameter Name	Value
primitiveType	RESPONSE
Operation (***op***)	The value of the parameter ***to*** in the associated non-blocking request.
statusResponseCode	"Locally accepted" – 800x
requestIdentifier (***ri***)	The value of the parameter ***ri*** in the associated non-blocking request.
originatingTimestamp (***ot***)	Timestamp when this message was built
Content (***cn***)	Reference to the <request> of the associated non-blocking request

7.2.2.2.6. Send Response primitive (Acknowledgement)

A Response primitive shall be sent back to the originator.

7.2.2.2.7. Update <request> resource

Changes in the attributes of a <request> resource shall be done by the Hosting CSE implicitly due to changes of the status (requestStatus) of the associated non-blocking request or due to the reception of an operation result (operationResult) in response to the associated non-blocking request. The Receiver CSE shall update attributes of an instance of <request> resource based on the following steps.

1) Update a value to the common attributes of <request> resource according to the following table:

Table 7.2.2.2.7-1: Common attributes settings for <request> resource

Attribute Name	Value
lastModifiedTime	Date/time of the last modification.
stateTag	This value is incremented on every modification.

2) Update a value to the resource-specific attributes of <request> resource according to the following table:

Table 7.2.2.2.7-2: Resource-specific attributes settings for <request> resource

Attribute Name	Value
requestStatus	Hosting CSE shall set this value to the appropriate status of the associated non-blocking request.
operationResult	Hosting CSE shall set this value to the result of the originally requested operation – if any – in line with the ***rc*** parameter in the associated non-blocking request.

7.2.2.2.8. Forwarding

If the "to" attribute in the request does not start with the CSEBase URI of the receiver, the receiver CSE shall forward the request or shall serve the request locally (see below).

If the "to" attribute in the request starts with the CSEBase URI of the receiver, then the receiver CSE shall handle the request locally.

Acting as an originator the CSE shall perform the following procedures:

1) "Send a Request to the receiver CSE".

2) "wait for Response primitive".

When the Response is received the receiver CSE shall:

1) Primitive specific procedure: Forward the Response to the original CSE.

7.2.2.3. Hosting CSE actions

7.2.2.3.1. Check existence of the addressed resource

The hosting CSE shall check if the resource addressed by the "to" attribute exists in the repository. If the resource does not exist, the hosting CSE shall reject the request with a "STATUS_NOT_FOUND".

7.2.2.3.2. Check validity of resource representation for CREATE

The handling below shall apply t to each attribute in the resource for CREATE request primitives and the handling depends on the "presence in CREATE request" column of the resource table. If the request is rejected based on the rules below, then the other attributes do not have to be checked.

If no resource representation is present in the CREATE request, then the request is rejected with a STATUS_BAD_REQUEST statusCode.

The id attribute has special handling. If the id-attribute is present in the CREATE request, the hosting CSE shall check if a resource with the same id already exists in the addressed collection. If such a resource exists and the response column is marked as M, then the hosting CSE shall reject the request with a "STATUS_CONFLICT".

If the expirationTime attribute is present in the resource representation, but its value indicates a time in the past, then the request shall be rejected with a STATUS_BAD_REQUEST.

N/A attribute

Indicates that the attribute is not supported in the CREATE request.

The resource indicated as N/A shall be created when the parent resource was created, or the resource shall be permanently exist in oneM2M System.

M attribute

If the attribute is present in the resource representation in the CREATE request, the hosting CSE shall check if the value is acceptable according to internal policies.

If the provided value is not accepted and the response column is marked M then the hosting CSE shall reject the request with a "STATUS_BAD_REQUEST".

If the attribute is not present in the resource representation in the CREATE request the hosting CSE shall reject the request with a "STATUS_BAD_REQUEST".

O attribute

If the attribute is present in the resource representation in the CREATE request, the hosting CSE shall check if the value is acceptable according to internal policies.

If the provided value is not accepted and the response column is marked M or O then the hosting CSE shall reject the request with a "STATUS_NOT_IMPLEMENTED".

NP attribute

If the attribute is present in the resource representation in the CREATE request, the hosting CSE shall reject the request with a "STATUS_BAD_REQUEST".

7.2.2.3.3. Check validity of resource representation for UPDATE

The handling below shall apply to each attribute in the resource for UPDATE request primitives and the handling depends on the "presence in UPDATE request" column of the resource table. If the request is rejected based on the rules below, then the other attributes do not have to be checked.

If the expirationTime attribute is present in the resource representation, but its value indicates a time in the past, then the request shall be rejected with a STATUS_BAD_REQUEST.

N/A attribute

Indicates that the attribute is not supported in the UPDATE requestThe resource indicated as N/A shall not allow any modification on the resource. If Originator requested to perform UPDATE operation on the resource, the request shall be rejected with a STATUS_METHOD_NOT_ALLOWED.

M attribute

If the attribute is present in the resource representation in the UPDATE request, the hosting CSE shall check if the value is acceptable according to internal policies.

If the provided value is not accepted and the response column is marked M, the hosting CSE shall reject the request with a "STATUS_BAD_REQUEST".

If the attribute is not present in the resource representation in the UPDATE request, the hosting CSE shall reject the request with a "STATUS_BAD_REQUEST".

O attribute

If the attribute is present in the resource representation in the UPDATE request, the hosting CSE shall check if the value is acceptable according to internal policies.

If the provided value is not accepted and the response column is marked M or O then the hosting CSE shall reject the request with a "STATUS_NOT_IMPLEMENTED" statusCode.

NP attribute

If the attribute is present in the resource representation in the UPDATE request, the hosting CSE shall reject the request with a "STATUS_BAD_REQUEST" unless the value provided for the attribute exactly matches the value in the current resource representation stored in the hosting CSE. In addition, the lastModifiedTime attribute shall always be accepted (but ignored) by the hosting CSE, no matter what value was provided in the request.

7.2.2.3.4. Create the resource

A new resource shall be created and correlated to the addressed and existing parent resource. This shall modify the resource representation of the addressed parent resource, specifically, if the parent resource has a lastModifiedTime attribute this shall be set to same value as the creationTime attribute of the created resource. The following rules shall be applied.

If the created resource type has an id attribute, then the URI of the created resource shall be the URI of its parent resource with the URI-encoded id appended. (e.g. http://CSEbase.operator.org/myAppID, for an application resource with id "myAppID" created in the parent resource http://CSEbase.operator.org).

If a resource with the same ID already exists in the addressed collection, the hosting CSE shall provide a new id that is unique within the collection.

If expirationTime attribute is present in the resource representation of the to be created resource and the expirationTime is set to a non-negative time, then an expiration timer shall be started by the hosting CSE. At timer expiration the related resource is deleted by "Delete the addressed resource".

For setting the attributes in the resource representation the following rules shall apply in CREATE request primitives:

M attribute

If the provided value is acceptable, the server shall use the provided value in the resource representation of the created resource.

O attribute

If a value is provided and accepted, then the server shall use the provided value in the resource representation of the created resource.

If the attribute is not provided or accepted, but the attribute is marked M in the response, the hosting CSE shall assign default value or assign value based on local policy.

If the attribute is not present in the resource representation in the CREATE request and the response column is marked O then the hosting CSE shall create the resource without the attribute.

NP attribute

If the attribute is not present in the resource representation in the CREATE request, and the response column is marked M, then the hosting CSE shall create the resource with the default value.

7.2.2.3.5. Retrieve the resource

An existing and accessible resource is addressed. The content of its attributes and references to its child resources shall be read in form of resource representation.

When the resource is read to provide a response to RETRIEVE request primitives, then the following rules shall be applied:

M attribute

The attribute is always present in the resource representation.

O attribute

The attribute is present in the resource representation if some conditions occur.

NP attribute

The attribute is never present in the resource representation.

7.2.2.3.6. Update the resource

An existing and accessible resource is addressed. For attributes that are not included in the **content** parameter, the hosting CSE shall not change such attributes. For attributes provided in the **content** parameter, The content of its attributes shall be updated with the following rules shall be applied:

If the *announceTo* attribute or *announcedAttribute* attribute of the resource is requested to update, the hosting CSE shall update the attribute as described in the "announce the resource or attribute" and "de-announce the resource or attribute" as specified in the clause 7.2.2.3.9 and clause 7.2.2.3.10, respectively.

M attribute

If the provided value is accepted, the server shall use the provided value in the resource representation of the updated resource.

O attribute

If a value is provided and the value is accepted, the server shall use the provided value in the resource representation of the updated resource.

If the attribute is not provided or accepted, but the attribute is marked M in the response, the hosting CSE shall assign default value or assign value based on local policy.

If this attribute is provided in the content parameter and does not exist in the target resource, the hosting CSE shall create such attribute with the provided value.

If this attribute is set to NULL in the content parameter and exists in the target resource, the hosting CSE shall delete such attribute if the deletion of the attribute is allowed by the local policy.

NP attribute

If the attribute is not present in the resource representation in the UPDATE request and the response column is marked M then the hosting CSE shall not update the attribute value. There is only one exception to this rule and that is the lastModifiedTime attribute. The hosting CSE shall set the lastModifiedTime to the current time whenever an update primitive is received.

If the attribute is present in the resource representation in the UPDATE request the presented value shall be ignored, i.e. the hosting CSE shall never update its resource representation based on the presence of an NP attribute value in an update.

If the expirationTime attribute is present and modified by the procedure and it is set to a non-negative time, then an expiration timer shall be re-started by the hosting CSE. At timer expiration the related resource is deleted by "Delete the addressed resource".

7.2.2.3.7. Delete the resource

An existing and accessible resource is addressed. The resource with all its attributes shall be deleted. Any expiration timer shall be stopped. This same procedure shall then be invoked (recursively) for each child resource of the deleted resource in case the child resource is only linked to the deleted resource.

The parent resource of the addressed resource shall be updated to remove the reference to the deleted resource. If the parent resource has a lastModificationTime attribute then this attribute shall be set to the time of the deletion.

If the resource is announced, the CSE shall try to de-announce the resource correspondingly.

7.2.2.3.8. Notify re-targeting

When the Hosting CSE receives a Notify request primitive targeting (i.e., to parameter) its <AE> resource, the Hosting CSE re-targets the primitive to the AE if the <AE> resource does not have any <pollingChannel> resource as a child.

1) Get pointOfAccess attribute value of the corresponding <AE> resource. If there is no available pointOfAccess address then the Hosting CSE shall send the Notify response primitive with "Cannot forward, target not reachable" error code.

2) Forward the Notify request primitive to the first address retrieved from pointOfAccess value

3) If the forwarding is failed due to "target not found", iterate 2) with the next address.

4) If the Hosting CSE cannot forward it in the end, then it send the Notify response primitive with "Cannot forward, target not reachable" error code.

7.2.2.3.9. Announce the resource or attribute

If CREATE request that contains an ***announceTo*** attribute is received,

- Compose the CREATE Request primitive as follows:
  - Link is set to the URI of the original resource.
  - If accessControlPolicyIDs of the original resource is not present, accessControlPolicyIDs is set to the same value with the parent resource or from the local policy of the original resource.
  - Attributes marked with MA and attributes marked with OA that are included in the announcedAttribute attribute. Such attributes shall be present in the original resource and set to same value as the original resource.
- Send a CREATE Request to the the CSE(s) represented by exact URI(s) or CSE-ID(s) in the announceTo of the request.
- Wait for Response primitive
- Add the URI of successfully announced resource to the announceTo attribute of the resource
- Include updated announceTo attribute in the content parameter in the Response to the received CREATE Request.

If UPDATE request that adds the URI or CSE-ID into the ***announceTo*** attribute is received,

- Compose the CREATE Request primitive as follows:
  - Link is set to the URI of the original resource.
  - If accessControlPolicyIDs of the original resource is not present, accessControlPolicyIDs is set to the same value with the parent resource or from the local policy of the original resource.
  - Attributes marked with MA and attributes marked with OA that are included in the announcedAttribute attribute. Such attributes shall be present in the original resource and set to same value as the original resource.
- Send a CREATE Request to the CSE(s) represented by exact URI(s) or CSE-ID(s) in the announceTo of the request, which is not included in the announceTo attribute of the original resource.
- Wait for Response primitive
- Add the URI of successfully announced resource to the announceTo attribute of the resource
- Include updated announceTo attribute in the content parameter in the Response to the received UPDATE Request.

If UPDATE request that adds the attribute name into the ***announcedAttribute*** attribute is received,

- Compose the UPDATE Request. The UPDATE Request shall provide the attribute name for the attribute to be announced, and the initial value for the attribute in the content parameter. The initial value shall be the same with the value from the original resource. The attribute that will be announced shall be marked as OA.
- Send UPDATE Requests to all announced resources listed in the announceTo attribute.
- Wait for Response primitive.
- Add the attribute name of the successfully announced attribute to the announcedAttribute attribute.
- Include updated announcedAttribute attribute in the content parameter in the Response to the received UPDATE Request.

If an attribute(s) specified as MA (See [6]) or an attribute(s) included in the announcedAttribute attribute is updated:

- Compose an UPDATE Request primitive by including the updated attribute(s) with its associated updated value.
- Send the UPDATE Request to all CSE(s) represented by the URI(s) in the announceTo attribute of the original resource.

If an attribute(s) specified as MA (See [6]) or an attribute(s) included in the announcedAttribute attribute is deleted:

- Compose an UPDATE Request primitive by including the updated attribute(s) with its value set to NULL.
- Send the UPDATE Request to all CSE(s) represented by the URI(s) in the announceTo attribute of the original resource.

7.2.2.3.10. De-announce the resource or attribute

If UPDATE Request that deletes the URI from the ***announceTo*** attribute is received:

- Compose the DELETE Request primitive.
- Send a DELETE Request to the CSE(s) represented by URI(s) in the announceTo attribute of the resource, which is not included in the announceTo of the request. The to parameter in the DELETE Request shall be set to the URI for the announced resource that will be deleted.
- Wait for Response primitive.
- Remove the URI of successfully de-announced resource from the announceTo attribute of the resource.
- Include updated announceTo attribute in the Content parameter in the Response to the UPDATE Request of the original resource.

If DELETE Request is received:

- Compose the DELETE Request primitive.
- Send DELETE Requests to all announced resources addressed by the URI(s) in the announceTo attribute of the resource.
- Wait for Response primitive.

If UPDATE request that deletes the attribute name from the *announcedAttribute* attribute is received:

- Compose the UPDATE Request primitive. The to parameter in the UPDATE Request shall be set to the URI for the announced resource. The UPDATE Request shall set the attribute to NULL that will be de-announced (i.e. to be deleted) in the content parameter. The attribute that will be de-announced shall be marked as OA.
- Send UPDATE Requests to all announced resources listed in the announceTo attribute of the original resource.
- Wait for Response primitive.

- Delete the attribute name of the successfully de-announced attribute from the announcedAttribute attribute.
- Include updated announcedAttribute attribute in the content parameter in the Response to the received UPDATE Request.

7.2.2.3.11. Create a successful Response

The receiver shall create a successful Response primitive with a statusCode indicating "STATUS_CREATED". The response shall include the representation of the addressed resource, if the hosting CSE modified any of the provided attributes as provided in the Request. The Response shall also include the URI of a created resource.

7.2.2.3.12. Create an unsuccessful Response

The receiver shall create an unsuccessful Response primitive with a statusCode indicating the detected error condition.

NOTE: Possible error codes and its error handling is described in resource specific procedure.

7.2.2.3.13. Resource Discovery Procedure

A resource discovery is used to discover resources in a CSE. A Resource discovery request is done by sending RETRIEVE request with *filterUsage*, one of the *filterCriteria* parameters, configured as "discovery" and the request may include other *filterCriteria* parameters as well. A resource discovery request procedure shall be comprised of the following actions.

Originator:

The Originator shall follow the steps from Orig-1.0 to Orig-6.0 specified in clause 7.2.1.2.1 Generic Resource Request Procedure for Originator.

In addition to Orig-1.0, the following steps shall be performed.

The *to* parameter in the RETRIEVE Request indicates the root of where the discovery begins.

The RETRIEVE Request shall include *filterUsage* parameter in fc.

The RETRIEVE Request may include other parameters of *filterCriteria*.

Receiver:

The Receiver shall follow the steps from Recv-1.0 to Recv-7.0 specified in clause 7.2.1.2.2 Generic Resource Request Procedure for Receiver.

Hosting CSE shall not perform steps from Recv-6.3 to Recv-6.6 and perform the following steps instead.

The Receiver shall find resources, which match all the configured *filterCriteria* and which the Originator has "Discover" access right, under the addressed resource".

In Recv-6.7, the Receiver shall include addresses for all the found resources.

The Receiver shall perform Recv-6.8 and the procedure is terminated.

7.2.2.4. Management Common Operations

7.2.2.4.1. Identify the managed entity and the management protocol

The Hosting CSE shall identify the managed entity to be managed via the <node> resource which is the parent resource in case of an addressed <mgmtObj> resource. In case of a <mgmtCmd> resource the entity to be managed is indicated in the execTarget attribute which addresses either a <node> resource or a group of resources of type <node>. Hence, in all cases the managed entity is ultimately identified through the <node> resource, from which the identifier of the device can be retrieved.

Then the Hosting CSE shall determine the management protocol to be used for communicating with the managed entity based on the objectID of the addressed <mgmtObj> resource. If the managed entity cannot be identified, the Hosting

CSE shall reject the request with the response status code set to "External object not reachable" in the Response primitive.

7.2.2.4.2. Locate the external management objects to be managed on the managed entity

The Hosting CSE shall locate the external management object information to be managed on the managed entity by the ***objectPath*** attribute of the <mgmtObj> resource addressed by the URI provided in the to primitive attribute. In the case that the to addresses an [objectAttribute] without ***objectPath*** attribute, the Hosting CSE shall locate the external management object information on the managed entity through the objectPath attribute of the <mgmtObj> resource of the addressed [objectAttribute], combining with their relative position in the external management object tree. If the external management object information cannot be located, the Hosting CSE shall reject the request with the response status code set to "External object not found " in the Response primitive.

In the case that the management server is external to the Hosting CSE, the Hosting CSE shall identify the management server that is capable of performing the operation on the external management object. If the management server cannot be identified, the Hosting CSE shall reject the request with the response status code set to “External object not reachable ” in the Response primitive.

7.2.2.4.3. Establish a management session with the managed entity or management server

In the case that the management server is embedded with the CSE, if there is no existing management session between the Hosting CSE and the managed entity, the Hosting CSE shall also trigger the managed entity to establish a management session with the Hosting CSE by sending triggering message to the managed entity using the determined management protocol in case such triggering mechanism is supported by the external management technology. If the triggering mechanism is not supported by the external management technology, the Hosting CSE shall reject the request with the response status code “Management session cannot be established”. If the management session cannot be established with the managed entity, the Hosting CSE shall reject the request with the response status code set to “Management session cannot be established”. If the management session cannot be established within a limited time span as per local policy, the Hosting CSE shall reject the request with the response status code set to "Management session establishment timeout " in the Response primitive.

In the case that the management server is external to the Hosting CSE, if there is no existing management session between the Hosting CSE and the management server that manages the external management objects, the Hosting CSE shall establish a session with the managed entity with the necessary access control privileges to perform the management request on the external management protocol. If the management session cannot be established with the management server, the Hosting CSE shall reject the request with the response status code set to “Management session cannot be established”. If the management session cannot be established within a limited time span as per local policy, the Hosting CSE shall reject the request with the response status code set to “Management session establishment timeout ” in the Response primitive.

7.2.2.4.4. Send the management request(s) to the managed entity corresponding to the received Request primitive

The Hosting CSE shall send the management request(s) to the managed entity or management server in the established management session in order to perform the management operation as requested by the received Request primitive. The management request shall address the external management object information on the managed entity as determined in clause 7.2.2.4 or in the primitive specific clauses. The management request being used is specific to the external management technology according to a pre-defined mapping relationship with the Request primitive. The internal data structure of the external management object addressed by the management request shall be determined based on the mapping relationship of the <mgmtObj>, or <mgmtCmd> resources and the external management objects or based on the generic mapping rule as specified in TS0001 clauses, 9.6.15, 9.6.16, and 9.6.17. The Hosting CSE shall extract the management results received from the managed entity or management server in order to prepare a Response primitive to be sent to the originator later. Unless explicitly stated, if the management request cannot be performed successfully, the Hosting CSE shall reject the Request primitive with the proper or management server in the Response primitive according to the mapping relationship withthe external management technology.

7.3. Resource Type-specific procedures and definitions

In the following clauses, each operation applicability on reference points is defined in clause 10.2 Resource Type-Specific Procedures [6].

7.3.1. Resource Type accessControlPolicy

7.3.1.1. Introduction

The <accessControlPolicy> resource is comprised of *privileges* and *selfPrivileges* attributes which represent a set of access control rules defining which entities (defined as accessControlOriginators) have the privilege to perform certain operations (defined as accessContolOperations) within specified contexts (defined as accessControlContexts) and are used by the CSEs in making access decision to specific resources.

The detailed description can be found in clause 9.6.2 in Architecture TS [6].

Table 7.3.1.1-1: Data Type Definition of accessControlPolicy

Data Type ID	File Name	Note
accessControlPolicyType	TBD.xsd	

Table 7.3.1.1-2: Applicable Common Attributes on accessControlPolicy

Attribute Name	Request Optionality				Default Value	Resource Specific Note
	C	R	U	D		
resourceType	NP	O	NP	NP		
resourceID	NP	O	NP	NP		
parentID	NP	O	NP	NP		
expirationTime	O	O	O	NP		
labels	O	O	O	NP		
creationTime	NP	O	NP	NP		
lastModifiedTime	NP	O	NP	NP		
announceTo	O	O	O	NP		
announcedAttribute	O	O	O	NP		

Table 7.3.1.1-3: Data Types for resource specific attributes

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
privileges	M	O	O	NP	m2m:privileges	
selfPrivileges	M	O	O	NP	m2m:privileges	

Table 7.3.1.1-4: Child resources of accessControlPolicy

Child Resource Type	Child Resource Name	Multiplicity	Ref. to Resource Type Definition
<subscription>	[variable]	0..n	Clause 7.3.7

7.3.1.2. accessControlPolicy Resource Specific Procedure on CRUD Operations

This sub-clause describes accessControlPolicy resource specific behaviour for CRUD operations.

7.3.1.2.1. Create

Originator:

No changes from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.3.1.2.2. Retrieve

Originator:

No changes from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.3.1.2.3. Update

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.3.1.2.4. Delete

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.3.2. Resource Type CSEBase

7.3.2.1. Introduction

A <CSEBase> resource shall represent a CSE. This <CSEBase> resource shall be the root for all the resources that are residing on the CSE. The detailed description can be found in clause 9.6.3 in Architecture TS([ref-Arch-TS]).

Table 7.3.2.1-1: Data Type Definition of <CSEBase>

Data Type ID	File Name	Note
CSEBaseType	*CDT-<<resource name>>-v1_0_0-<<date of published>>.xsd*	

Table 7.3.2.1-2: Common Attributes on <CSEBase>

Attribute Name	Request optionality				Default Value	Resource Specific Note
	C	R	U	D		
resourceType	NP	O	NP	NP		
creationTime	NP	O	NP	NP		
lastModifiedTime	NP	O	NP	NP		
accessControlPolicyIDs	NP	O	NP	NP		
labels	NP	O	NP	NP		

Table 7.3.2.1-3: Resource specific attributes on <CSEBase>

Attribute Name	Request optionality				Data Type	Default Value and Constraints
	C	R	U	D		

cseType	NP	O	NP	NP	m2m:cseTypeID	
CSE-ID	NP	O	NP	NP	m2m:id	
supportedResourceType	NP	O	NP	NP	m2m:commaList	
pointOfAccess	NP	O	NP	NP	list of xs:string	
nodeLink	NP	O	NP	NP	xs:anyURI	
notificationCongestionPolicy	NP	O	NP	NP	m2m:notificationCongestionPolicy	

Table 7.3.2.1-4: Reference to child resources of <CSEBase>

Child Resource Type	Child Resource Name	Multiplicity	Ref. to Resource Type Definition
<remoteCSE>	[variable]	0..n	Clause 7.3.3
<node>	[variable]	0..n	Clause 7.3.17
<AE>	[variable]	0..n	Clause 7.3.4
<container>	[variable]	0..n	Clause 7.3.5
<group>	[variable]	0..n	Clause 7.3.12
<accessControlPolicy>	[variable]	0..n	Clause 7.3.1
<subscription>	[variable]	0..n	Clause 7.3.7
<mgmtCmd>	[variable]	0..n	Clause 7.3.15
<locationPolicy>	[variable]	0..n	Clause 7.3.9
<statsConfig>	[variable]	0..n	Clause 7.3.22
<statsCollect>	[variable]	0..n	Clause 7.3.24
<request>	[variable]	0..n	Clause 7.3.11
<delivery>	[variable]	0..n	Clause 7.3.10

7.3.2.2. Operations

This clause describes <CSEBase> resource specific behaviour for CRUD operations.

7.3.2.2.1. Create

Originator:

The <CSEBase> resource shall not be created via API.

Receiver: The Receiver shall execute the following steps in order.

Rcv-C-Rq-1.0 "Create an unsuccessful Response" with responseCode TBD (Method not allowed).

Rcv-C-Rq-2.0 "Send Response primitive".

7.3.2.2.2. Retrieve

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.3.2.2.3. Update

Originator:

The <CSEBase> resource shall not be UPDATEed via API.

Receiver:

The Receiver shall execute the following steps in order.

Rcv-U-Rq-1.0 "Create an unsuccessful Response" with responseCode TBD (Method not allowed).

Rcv-U-Rq-2.0 "Send Response primitive".

7.3.2.2.4. Delete

Originator:

The <CSEBase> resource shall not be DELETEed via API.

Receiver:

The Receiver shall execute the following steps in order.

Rcv-D-Rq-1.0 "Create an unsuccessful Response" with responseCode TBD (Method not allowed).

Rcv-D-Rq-2.0 "Send Response primitive".

7.3.3. Resource Type remoteCSE

7.3.3.1. Introduction

A <remoteCSE> resource shall represent a remote CSE that is registered to the Registrar CSE. <remoteCSE> resources shall be located directly under the <CSEBase>.

Conversely each registered CSE shall also be represented as a sub-set of <remoteCSE> resource in the registering CSE's <CSEBase>.

The detailed description can be found in clause 9.6.4 in Architecture TS.

Table 7.3.3.1-1: Data Type Definition of <remoteCSE>

Data Type ID	File Name	Note
remoteCSEType	CDT-remoteCSE-v1_0_0-20140718.xsd	

Table 7.3.3.1-2: Common Attributes on <remoteCSE>

Attribute Name	Request optionality				Default Value	Resource Specific Note
	C	R	U	D		
resourceType	NP	O	NP	NP		
resourceID	NP	O	NP	NP		
parentID	NP	O	NP	NP		
creationTime	NP	O	NP	NP		
lastModifiedTime	NP	O	NP	NP		
expirationTime	O	O	O	NP		
accessControlPolicyIDs	O	O	O	NP		
labels	O	O	NP	NP		
announceTo	O	O	O	NP		
announcedAttribute	O	O	O	NP		

Table 7.3.3.1-3: Data Types for resource specific attributes

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
cseType	O	O	NP	NP	m2m:cseType	No default
pointOfAccess	O	O	O	NP	List of xs: string	Absent if the remote CSE is not request-reachable
CSEBase	M	O	NP	NP	xs:anyURI	No default
CSE-ID	M	O	NP	NP	m2m:id	No default
M2M-Ext-ID	O	O	O	NP	m2m:externalId	No default
Trigger-Recipient-ID	O	O	O	NP	m2m:triggerRecipientId	No default
requestReachability	M	O	O	NP	xs:boolean	No default
nodeLink	NP	O	NP	NP	xs:anyURI	No default

Table 7.3.3.1-4: Resource specific attributes on <remoteCSE>

Attribute Name	Request optionality				Data Type	Default Value and Constraints
	C	R	U	D		
cseType	M	O	NP	NP	m2m:cseTypeID	
pointOfAccess	O	O	O	NP	list of xs:string	
CSEBase	M	O	NP	NP	xs:anyURI	
CSE-ID	M	O	NP	NP	m2m:id	
M2M-Ext-ID	O	O	O	NP	m2m:externalId	
Trigger-Recipient-ID	O	O	O	NP	xs:unsignedInt	
requestReachability	M	O	O	NP	xs:boolean	
nodeLink	NP	O	NP	NP	xs:anyURI	

Table 7.3.3.1-5: Reference to child resources of <remoteCSE>

Child Resource Type	Child Resource Name	Multiplicity	Ref. to Resource Type Definition
AE	[variable]	0..n	Clause 7.3.4
container	[variable]	0..n	Clause 7.3.5
group	[variable]	0..n	Clause 7.3.12
accessControlPolicy	[variable]	0..n	Clause 7.3.1
subscription	[variable]	0..n	Clause 7.3.7
pollingChannel	[variable]	0..n	Clause 7.3.20
schedule	[variable]	0..n	Clause 7.3.8

7.3.3.2. Resource Specific Procedure on CRUD Operations

This sub-clause describes <remoteCSE> resource specific behaviour for CRUD operations.

7.3.3.2.1. Create

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.1.

7.3.3.2.2. Retrieve

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.1.

7.3.3.2.3. Update

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.1.

7.3.3.2.4. Delete

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.1.

7.3.4. Resource Type AE

7.3.4.1. Introduction

The <AE> resource represents information about an Application Entity known to a given Common Services Entity.

The detailed description can be found in clause 9.6.5 in Architecture TS [6].

Table 7.3.4.1-1: Data Type Definition of AE

Data Type ID	File Name	Note
AE	CDT-AE-v1_0_0-20140729.xsd	XSD schema for AE resource
AE	CDT-AE-v1_0_0-20140729.json	JSON schema for AE resource

Table 7.3.4.1-2: Applicable Common Attributes on AE

Attribute Name	Request Optionality				Default Value	Resource Specific Note
	C	R	U	D		
resourceType	NP	O	NP	NP		
parentID	NP	O	NP	NP		
accessControlPolicyIDs	O	O	O	NP		
creationTime	NP	O	NP	NP		
expirationTime	O	O	O	NP		
lastModifiedTime	NP	O	NP	NP		
labels	O	O	NP	NP		
announceTo	O	O	O	NP		
announcedAttribute	O	O	O	NP		

Table 7.3.4.1-3: Data Types for resource specific attributes

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
name	NP	O	O	NP	xs:string	
App-ID	NP	O	O	NP	xs:string	
AE-ID	NP	O	O	NP	m2m:id	
pointOfAccess	O	O	O	NP	list of xs:string	
ontologyRef	O	O	O	NP	xs:anyURI	
nodeLink	NP	O	NP	NP	xs:anyURI	

Table 7.3.4.1-4: Child resources of AE

Child Resource Type	Child Resource Name	Multiplicity	Ref. to Resource Type Definition
<subscription>	[variable]	0..n	Clause 7.3.7
<container>	[variable]	0..n	Clause 7.3.5
<group>	[variable]	0..n	Clause 7.3.12
<accessControlPolicy>	[variable]	0..n	Clause 7.3.1
<pollingChannel>	[variable]	0..n	Clause 7.3.20

7.3.4.2. AE Resource Specific Procedure on CRUD Operations

This sub-clause describes AE resource specific behaviour for CRUDN operations.

7.3.4.2.1. Create

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.3.4.2.2. Retrieve

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.3.4.2.3. Update

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.3.4.2.4. Delete

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.3.4.2.5. Notify

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.3.5. Resource Type container

7.3.5.1. Introduction

This resource represents a container for data instances. It is used to share information among other entities and potentially to track the data. A <container> resource has no associated content, only attributes and child resources.

The detailed description can be found in clause 9.6.6 in Architecture TS.

Table 7.3.5.1-1: Data Type Definition of <container>

Data Type ID	File Name	Note
containerType	(TBD)	

Table 7.3.5.1-2: Common Attributes on <container>

Attribute Name	Request Optionality				Default Value	Resource Specific Note
	C	R	U	D		
resourceType	NP	O	NP	NP		
resourceID	NP	O	NP	NP		
parentID	NP	O	NP	NP		
expirationTime	O	O	O	NP		
accessControlPolicyIDs	O	O	O	NP		
Labels	O	O	NP	NP		
creationTime	NP	O	NP	NP		
lastModifiedTime	NP	O	NP	NP		
stateTag	NP	O	NP	NP		
announceTo	O	O	O	NP		
announcedAttribute	O	O	O	NP		

Table 7.3.5.1-3: Resource specific attributes on <container>

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
Creator	NP	O	NP	NP	m2m:cseIdType or m2m:aeIdType	
currentByteSize	NP	O	NP	NP	xs:integer	
currentNbrOfInstances	NP	O	NP	NP	xs:integer	
Latest	NP	O	NP	NP	xs:string	
locationID	O	O	O	NP	xs:anyURI	
maxNrOfInstances	O	O	O	NP	xs:integer	
maxByteSize	O	O	O	NP	xs:integer	
maxInstanceAge	O	O	O	NP	xs:integer	
currentNrOfInstances	NP	O	NP	NP	xs:integer	
currentByteSize	NP	O	NP	NP	xs:integer	
ontologyRef	O	O	O	NP	xs:anyURI	

Editor's Note: Default values are filled out later.

Table 7.3.5.1-4: Reference to child resources of <container>

Child Resource Type	Child Resource Name	Multiplicity	Ref. to in Resource Type Definition
contentInstance	[variable]	0..n	Clause 7.3.5
Subscription	[variable]	0..n	Clause 7.3.6
Container	[variable]	0..n	Clause 7.3.4

7.3.4.2 <container> Resource Specific Procedure on CRUD Operations

This clause describes container resource specific behaviour for CRUD operations.

7.3.4.2.1 Create

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

When any <containerInstance> existed at same level in structured URI, CREATE request for <container> shall be rejected by STATUS_NOT_ALLOWED.

7.3.4.2.2 Retrieve

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.3.4.2.3 Update

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

.

7.3.4.2.4 Delete

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.3.6. Resource Type contentInstance

コメントの追加 **[A4]:** Fix the numbering—7.5 is missing.

7.3.6.1. Introduction

The <contentInstance> resource represents a data instance in the container..

The detailed description can be found in clause 9.6.7 in Architecture TS.

Table 7.3.6.1-1: Data Type Definition of <contentInstance>

Data Type ID	File Name	Note
containerType	(TBD)	

Table 7.3.6.1-2: Applicable Common Attributes on <contentInstance>

Attribute Name	Request Optionality				Default Value	Resource Specific Note
	C	R	U	D		
resourceType	NP	O	NP	NP		
resourceID	NP	O	NP	NP		
parentID	NP	O	NP	NP		
Labels	O	O	NP	NP		
creationTime	NP	O	NP	NP		
lastModifiedTime	NP	O	NP	NP		
stateTag	NP	O	NP	NP		
announceTo	O	O	NP	NP		
announcedAttribute	NP	O	NP	NP		
typeOfContent	M	O	NP	NP		
contentSize	O	O	NP	NP		
ontologyRef	O	O	NP	NP		

Table 7.3.6.1-3:Resource specific attributes on <contentInstance>

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
typeOfContent	M	O	NP	NP	xs:string	
contentSize	O	O	NP	NP	xs:nonNegativeInteger	
ontologyRef	O	O	NP	NP	xs:anyURI	
Content	M	O	NP	NP	xs:base64Binary	
encoding	M	O	NP	NP	m2m:encodingType	

Table 7.3.6.1-4: Reference to child resources of <contentInstance>

Child Resource Type	Child Resource Name	Multiplicity	Ref. to in Resource Type Definition
Subscription	[variable]	0..n	Clause 7.3.7

7.3.6.2. <contentInstance> Resource Specific Procedure on CRUD Operations

This clause describes container resource specific behaviour for CRUD operations.

7.3.6.2.1. Create

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

The Originator may omit the name of the targeted <contentInstance> unless the Originator need to refer specific content later.

7.3.6.2.2. Retrieve

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.The Originator may omit the name of the targeted <contentInstance> resource when the latest version of stored content is requested.

7.3.6.2.3. Update

The <contentInstance> resource shall not be Updated via API.

7.3.6.2.4. Delete

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.3.7. Resource Type subscription

7.3.7.1. Introduction

The <subscription> resource contains subscription information for its subscribed-to resource. The subscription resource is a child of the subscribed to resource.

The detailed description can be found in clause 9.6.8 in Architecture TS.

Table 7.3.7.1-1: Data Type Definition of subscription

Data Type ID	File Name	Note
subscriptionType	*TBD*	

Table 7.3.7.1-2: Applicable Common Attributes on subscription

Attribute Name	Request Optionality				Default Value	Resource Specific Note
	C	R	U	D		
resourceType	NP	O	NP	NP		
resourceID	NP	O	NP	NP		
parentID	NP	O	NP	NP		
accessControlPolicyIDs	O	O	O	NP		
creationTime	NP	O	NP	NP		
expirationTime	O	O	O	NP		
lastModifiedTime	NP	O	NP	NP		
labels	O	O	O	NP		

Table 7.3.7.1-3: Data Types for resource specific attributes

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
eventNotificationCriteria	O	O	O	NP	m2m:eventNotificationCriteria	
expirationCounter	O	O	O	NP	xs:positiveInteger	
notificationURI	M	O	O	NP	list of xs:anyURI	
groupName	O	O	O	NP	xs:anyURI	
notificationForwardingURI	O	O	O	NP	xs:anyURI	
batchNotify	O	O	O	NP	m2m:batchNotify	
rateLimit	O	O	O	NP	m2m:rateLimit	
preSubscriptionNotify	O	O	NP	NP	xs:positiveInteger	
pendingNotification	O	O	O	NP	m2m:pendingNotification	
notificationStoragePriority	O	O	O	NP	xs:positiveInteger	
latestNotify	O	O	O	NP	xs:boolean	
notificationContentType	O	O	O	NP	m2m:notificationContentType	
notificationEventCat	O	O	O	NP	m2m:eventCat	
creator	O	O	O	NP	m2m:id	
subscriberURI	O	O	NP	NP	xs:anyURI	

Table 7.3.7.1-4: Reference of child resources

Child Resource Type	Child Resource Name	Multiplicity	Ref. . to in Resource Type Definition
<schedule>	notificationSchedule	0..1	7.3.8

7.3.7.2. Resource Specific Procedure on NP CRUD Operations

This sub-clause describes <subscription> resource specific behaviour for CRUD operations.

7.3.7.2.1. Create

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

The followings are additional Hosting CSE procedures to the generic resource handling procedures (figure 7.2.1.2.2-2 in clause 7.2.1.2.2). The additional procedures shall be inserted from Recv-6.2 to Recv-6.8 as below.

The resource handling procedure for the Hosting CSE which receives <subscription> Create request shall perform the following procedures in order:

1. Recv-6.2
2. Recv-6.3
3. Check if the subscribed-to resource, addressed in *to* parameter in the Request, is subscribable. Subscribable resource types are defined in [6], they have <subscription> resource types as their child resources.

   If it is not subscribable, the Hosting CSE shall return the Notify response primitive with “Target is not subscribable” error.

4. Check if the Originator has privileges for retrieving the subscribed-to resource.

   If the Originator does not have the privilege, the Hosting CSE shall return the Notify response primitive with “Create error - no privilege” error.

5. If the *notificationURI* is not the Originator, the Hosting CSE should send a Notify request primitive to the *notificationURI* with *verificationRequest* parameter set as TRUE (clause 7.4.2.2).
    a. If the Hosting CSE cannot send the Notify request primitive, the Hosting CSE shall return the Notify response primitive with "Cannot initiate subscription verification" error.
    b. If the Hosting CSE sent the primitive, the Hosting CSE shall check if the Notify response primitive contains "Subscription verification failed" error. If so, the Hosting CSE shall return the Create response primitive with "Subscription verification failed" error to the Originator.
6. Recv-6.4
7. Recv-6.5

   If the *notificationURI* is not the Originator, the Hosting CSE shall store Originator ID to *creator* attribute.

8. Recv-6.6
9. Recv-6.7
10. Recv-6.8

7.3.7.2.2. Retrieve

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.3.7.2.3. Update

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

7.3.7.2.4. No change from the generic procedures in clause 7.2.1.2.2.Delete

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.3.8. Resource Type schedule

7.3.8.1. Introduction

The <schedule> resource shall represent scheduling information in the context of its parent resource. If a <schedule> resource is not present as a child resource then there are no time-constraints on the context of its parent resource. An Originator shall have the same access control privileges to the <schedule> resource as it has to its parent resource.

The detailed <schedule> resource description can be found in clause 9.6.9 of the Architecture TS.

Table 7.3.8.1-1: Data Type Definition of <schedule>

Data Type ID	File Name	Note
scheduleType	CDT-<<resource name>>-v1_0_0-<<date of published>>.xsd	

Table 7.3.8.1-2: Common Attributes on <schedule >

Attribute Name	Request Optionality				Default Value	Resource Specific Note
	C	R	U	D		
resourceType	NP	O	NP	NP		
resourceID	NP	O	NP	NP		
parentID	NP	O	NP	NP		
accessControlPolicyIDs	O	O	O	NP		
creationTime	O	O	NP	NP		
expirationTime	O	O	O	NP		
lastModifiedTime	NP	O	NP	NP		
labels	O	O	NP	NP		
announceTo	O	O	O	NP		
announcedAttribute	O	O	O	NP		

Table 7.3.8.1-3: Resource specific attributes on <schedule>

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
scheduleElement (L)	M	O	O	M	Extended Crontab Data String	No Default See data type definition for constraints

Editor's note; How to represent multiplicity need to clarify. *scheduleElement is 1..n*

Editor's Note: Optionality of Attributes should be referred in Arch-TS or self-contained ? *scheduleElement is Optionally announced for <scheduleAnnc>*

Editor's Note: It is not clear where in the document the following data type definition should be located since it is <schedule> resource specific.

The set of *scheduleElement* attributes together represent the defined schedule. All *scheduleElement attributes* shall be updated as a set. The set of *scheduleElement* attributes expresses time periods defined by second, minute, hour day of month, month, and year.

The "Extended Crontab Data String" Data Type is a string comprising 7 fields per Table below, separated by white space. The allowed values are used together with 6 special characters in each "scheduleElement".

Table 7.3.8.1-4: Extended Crontab Data String structure

Field name	Mandatory	Allowed values	Allowed special characters
Seconds	Yes	0-59	* / , -
Minutes	Yes	0-59	* / , -
Hours	Yes	0-23	* / , -
Day of Month	Yes	1-31	* / , - L #
Month	Yes	1-12 or JAN-DEC	* / , -
Day of Week	Yes	0-6 or SUN-SAT	* / , - L #
Year	Yes	1970–2099	* / , -

Asterisk (*)

The asterisk indicates that the expression matches for all values of the field. E.g., using an asterisk in the 5th field (month) indicates every month.

Slash (/)

Slashes describes periodicity within ranges. For example 3-59/15 in the 2nd field (minutes) indicates the third minute of the hour and every 15 minutes thereafter within the hour. The form "*/..." is equivalent to the form "start range-end range/periodicity", over the range of the field. */2 in the 2nd field (minutes) indicates every other minute. 2014/1 in the 7th field (years) indicates to repeat every year after 2014.

Comma (,)

Commas are used to separate items of a list. For example, using "MON,WED,FRI" in the 6th field (day of week) means Mondays, Wednesdays and Fridays.

Hyphen (-)

Hyphens define ranges. For example, 2010-2020 indicates every year between 2010 and 2020 AD, inclusive.

L

'L' stands for "last". When used in the day-of-week 6th field, it allows you to specify constructs such as "the last Friday" ("5L") of a given month. In the day-of-month 4th field, it specifies the last day of the month.

Hash (#)

'#' is allowed for the day-of-week 6th field, and must be followed by a number between one and five. It allows you to specify constructs such as "the second Friday" of a given month.

Editor's Note: It is unclear whether any examples should be included.

Examples:

Example 1 – Starting at 2am on 1 Oct, 2014 the device will be reachable between 2:00–2:05, 6:00-6:05, 10:00-10:05 and 14:00-14:05 every day.

scheduleElement = "* 0-5 2,6,10,14 1/1 10/1 * 2014/1"

Example 2 – Starting at 2am on 1 Oct, 2014, the device will be reachable between 2:00–2:05, 6:00-6:05, 10:00-10:10 and 14:00-14:10 every day.

scheduleElement = "* 0-5 2,6 * 10/1 * 2014/1"

scheduleElement = "* 0-10 10,14 * 10/1 * 2014/1"

Example 3 - Starting at 2am on 1 Oct, 2014, the device will be reachable between 2:00–2:05, 6:00-6:05, on Mondays, Wednesdays and Fridays, 10:00-10:05 and 14:00-14:05 on Tuesdays and Saturdays, and 4:00-4:15 and 16:00-16:15 on Sundays.

scheduleElement = "* 0-5 2,6 * 10/1 1,3,5 2014/1"

scheduleElement = "* 0-5 10,14 * 10/1 2,6 2014/1"

scheduleElement = "* 0-15 4,16 * 10/1 7 2014/1"

Example 4 - Starting at 1am on 1 Oct, 2014, the device will be reachable for 5 minutes after every 4 hours of the day

scheduleElement = "* 0-5 1/4 * 10/1 * 2014/1"

Table 7.3.8.1-5: Reference to child resources of <schedule >

Child Resource Type	Child Resource Name	Multiplicity	Ref. to in Resource Type Definition
Subscription	[variable]	0..n	Clause 7.3.6

7.3.8.2. Resource Specific Procedure on CRUD Operations

This sub-clause describes <schedule> resource specific behaviour for CRUD operations.

7.3.8.2.1. Create

This procedure follows the Generic Resource Create Request Procedure specified in clause 7.2.1.2.1.

7.3.8.2.2. Retrieve

This procedure follows the Generic Resource Retrieve Request Procedure specified in clause 7.2.2.

7.3.8.2.3. Update

This procedure follows the Generic Resource Update Request Procedure specified in clause 7.2.2.

7.3.8.2.4. Delete

This procedure follows the Generic Resource Delete Request Procedure specified in clause 7.2.2.

7.3.9. Resource Type locationPolicy

7.3.9.1. Introduction

The <locationPolicy> resource represents the method for obtaining and managing geographical location information of an M2M Node. The detailed description can be found in the clause 9.6.10 in [6].

The resource specific attributes information is defined in the annex A.

Table 7.3.9.1-1: Data Type Definition of locationPolicy

Data Type ID	File Name	Note
locationPolicyType	TBD	XSD schema for locationPolicy resource
locationPolicyType	TBD	JSON schema for locationPolicy resource

Table 7.3.9.1-2: Applicable Common Attributes on [locationPolicy]

Attribute Name	Request Optionality				Default Value	Resource Specific Note
	C	R	U	D		
resourceType	NP	O	NP	NP	"locationPolicy"	
resourceID	NP	O	NP	NP	None	
parentID	NP	O	NP	NP	None	
expirationTime	O	O	O	NP	Default is determined by Hosting CSE policy	
accessControlPolicyIDs	O	O	O	NP	Default is determined by Hosting CSE policy	
creationTime	NP	O	NP	NP	None	
lastModifiedTime	NP	O	NP	NP	None	
labels	O	O	O	NP	Empty	
announceTo	O	O	O	NP	None	
announceAttribute	O	O	O	NP	None	

Table 7.3.9.1-3: Data Types for resource specific attributes

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
locationSource	M	O	NP	NP	m2m:locationSource	
locationUpdatePeriod	O	O	O	NP	xs:duration	
locationTargetId	O	O	NP	NP	m2m:nodeId	
locationServer	O	O	NP	NP	xs:anyURI	
locationContainerID	NP	O	NP	NP	xs:anyURI	
locationContainerName	O	O	O	NP	xs:string	
locationStatus	NP	O	NP	NP	xs:string	

Table 7.3.9.1-4: Reference of child resources

Child Resource Type	Child Resource Name	Multiplicity	Ref. to Resource Type Definition
<subscription>	[variable]	0..n	Clause 7.3.7

7.3.9.2. Operations

This clause describes locationPolicy resource specific primitive behaviour for CRUD operations.

7.3.9.2.1. Create

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

The procedure of the Receiver written in the clause 7.2.1.2.2 (from *Recv 1.0 to Recv 6.5*) shall be the same as initial steps. The following steps are the <locationPolicy> resource type specific procedure for CREATE operation.

1) After the successful creation of <locationPolicy> resource, the Hosting CSE shall create <container> resource where the actual location information will be stored and the resource shall contain cross-references for the both resources, *locationContainerID* attribute for the <locationPolicy> resource and *locationID* attribute for the <container> resource. The name of the created <container> resource shall be determined by the *locationContainerID* attribute if it is applicable.

2) Check the *locationSource* and *locationUpdatePeriod* attributes:

   a) If the *locationSource* attribute is set by 'Network Based' and *locationUpdatePeriod* attribute is set by any duration value (higher than 0 second), then continue with the step 3.

   b) If the *locationSource* attribute is set by 'Device Based' and *locationUpdatePeriod* attribute is set by any duration value (higher than 0 second), then continue with the step 4.

   c) If the *locationSource* attribute is set by 'Sharing Based' and *locationUpdatePeriod* attribute is set by any duration value (higher than 0 second), then continue with the step 5.

3) The Hosting CSE shall retrieve the *locationTargetID* and *locationServer* attributes from the stored <location Policy> resource.

   In case either the *locationTargetID* or *locationServer* attribute cannot be obtained, the hosting CSE shall reject the request with the Response Status Code defined in clause 6.5.Then, the Hosting CSE shall transform the location-acquisition request into Location Server request [i.3], using the attributes stored in <locationPolicy> resource. The Hosting CSE shall also provide default values for other required parameters (e.g. quality of position) in the Location Server request according to local policies.

The Hosting CSE shall send this Location Server request to the location server using, for example, OMA Mobile Location Protocol [i.5] and OMA RESTful NetAPI for Terminal Location [i.3]. The location server performs positioning procedure based upon the Location Server request. Then continue with the step 6.

Based on the period information, *locationUpdatePeriod* attribute, this step can be periodically repeated or the location server can only notify the Hosting CSE of location information that performs periodically.

NOTE 1: The location server performs the privacy control and only responds successfully if the positioning procedure is permitted.

NOTE 2: The detail information on how the Location Server request message is converted into OMA RESTful NetAPI for Terminal Location message is described in Annex G..

4) The Hosting CSE shall perform positioning procedure using location determination modules and technologies (e.g. GPS). Then continue with the step 6.

Based on the period information, *locationUpdatePeriod* attribute, this step can be periodically repeated.

NOTE 3: The Hosting CSE can utilize the internal interface (e.g. System Call) to communicate with the modules and technologies. The detail procedure is out-of-scope.

5) The Hosting CSE shall collect information of topology of M2M Area Network using <node> resource and find the closest Node from the Originator that has registered with the Hosting CSE and has location information. The closest Node is determined by the minimum hop based on the collected topology information.

a) If the Hosting CSE can find the closest Node from the Originator, the location information of the closest Node shall be stored as the location information of the Originator into a <contentInstance> resource under the created <container> resource.

b) If the Hosting CSE cannot find the closest Node from the Originator, the location information of the Hosting CSE shall be stored as the location information of the Originator into a <contentInstance> resource under the created <container> resource.

6) The Hosting CSE shall receive the corresponding response and transform it into a Response primitive.

a) If the positioning procedure is failed, the Hosting CSE shall store a statusCode based on the error code in the locationStatus attribute in the created <locationPolicy> resource.

b) If the positioning procedure is successfully complete which means that the Hosting CSE acquires the location information, The Hosting CSE shall store the acquired location information into a <contentInstance> resource under the created <container> resource.

7) Recv-6.6

8) Recv-6.7

7.3.9.2.2. Retrieve

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.3.9.2.3. Update

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.3.9.2.4. Delete

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

The procedure of the Receiver written in the clause 7.2.1.2.2 (from *Rcv-D-1.0* to *Rcv-D-10.0*) shall be the same as initial steps. A following step is the <locationPolicy> resource type specific procedure for DELETE operation.

1) Once the <locationPolicy> resource is deleted, the Receiver shall delete the associated resources (e.g. <container>, <contentInstance> resources). If the locationSource attribute and the *locationUpdatePeriod* attribute of the <locationPolicy> resource has been set with appropriate value, the Receiver shall tear down the session. The specific mechanism used to tear down the session depends on the support of the Underlying Network and other factors.

7.3.10. Resource Type delivery

7.3.10.1. Introduction

In order to be able to initiate and manage the execution of data delivery in a resource-based manner, resource type delivery is defined. This resource type shall be used for forwarding requests from one CSE to another CSE when the ***da*** parameter in the request is set to ON. The detailed description can be found in clause 9.6.11 in Architecture TS [6].

Table 7.3.10.1-1: Data Type Definition of delivery

Data Type ID	File Name	Note
deliveryType	CDT-delivery-v1_0_0-<<date of published>>.xsd	

Table 7.3.10.1-2: Common Attributes on delivery

Attribute Name	Request Optionality				Default Value	Resource Specific Note
	C	R	U	D		
resourceType	NP	O	NP	NP	None	
resourceID	NP	O	NP	NP	None	
expirationTime	O	O	O	NP	Default is determined by Hosting CSE policy	
parentID	NP	O	NP	NP	None	
creationTime	NP	O	NP	NP	None	
lastModifiedTime	NP	O	NP	NP	None	
accessControlPolicyIDs	O	O	O	NP	Default is determined by Hosting CSE policy	
Labels	O	O	O	NP	Empty	
stateTag	NP	O	NP	NP	0	

Table 7.3.10.1-3: Resource Specific Attributes on delivery

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
Source	M	O	NP	NP	xs:anyURI	None
Target	M	O	NP	NP	xs:anyURI	None
Lifespan	M	O	O	NP	xs:dateTime	None
eventCat	M	O	O	NP	m2m:eventCat	None
deliveryMetaData	M	O	O	NP	m2m: deliveryMetaData	Default is determined by Hosting CSE policy
AggregatedRequest	O	O	O	NP	m2m:aggregatedRequest	None

Table 7.3.10.1-4: Child resources for delivery

Child Resource Type	Child Resource Name	Multiplicity	Ref. to Resource Type Definition
subscription	variable	0..n	7.3.6

7.3.10.2. Resource Specific Procedure on CRUD Operations

This clause describes <delivery> resource specific behaviour for CRUD operations.

7.3.10.2.1. Create

Originator:

Primitive specific operation on Org-1.0 "Compose Request primitive":

1) The Originator shall use a blocking request (i.e. ***rt***=blockingRequest).

2) The Originator shall provide the content of the <delivery> resource.

No change for the remaining steps from the generic procedures in clause 7.2.1.2.1.

Receiver:

Primitive specific operation on Rcv-1.0 "Check the syntax of received message":

1) If the request is received over Mca reference point, the Receiver CSE shall execute the following steps in order.

   a. "Create an unsuccessful Response" with responseStatusCode TBD (Method not allowed).

   b. "Send Response primitive".

NOTE: Determination of the reference point is to the discretion of the Receiver CSE implementation.

Primitive specific operation after Rcv-6.8 "Check the syntax of received message".

1) If the "target" attribute in <request> resource does not start with the CSEBase URI of the Receiver CSE, the Receiver CSE shall forward the request.

No change for the remaining steps from the generic procedures in clause 7.2.1.2.2.

7.3.10.2.2. Retrieve

Originator:

Primitive specific operation on Org-1.0 "Compose Request primitive":

1) The Originator shall use a blocking request (i.e. ***rt***=blockingRequest).

No change for the remaining steps from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.3.10.2.3. Update

Originator:

Primitive specific operation on Org-1.0 "Compose Request primitive":

1) The Originator shall use a blocking request (i.e. ***rt***=blockingRequest).

2) The Originator shall provide the content of the <delivery> resource.

No change for the remaining steps from the generic procedures in clause 7.2.1.2.1.

Receiver:

Primitive specific operation on Rcv-1.0 "Check the syntax of received message":

1) If the request is received over Mca reference point, the Receiver CSE shall execute the following steps in order.
   a. "Create an unsuccessful Response" with responseStatusCode TBD (Method not allowed).
   b. "Send Response primitive".

No change for the remaining steps from the generic procedures in clause 7.2.1.2.2.

7.3.10.2.4. Delete

Originator:

Primitive specific operation on Org-1.0 "Compose Request primitive":

1) The Originator shall use a blocking request (i.e. ***rt***=blockingRequest).

No change for the remaining steps from the generic procedures in clause 7.2.1.2.1.

Receiver:

Primitive specific operation on Rcv-1.0 "Check the syntax of received message":

1) If the request is received over Mca reference point, the Receiver CSE shall execute the following steps in order.
   a. "Create an unsuccessful Response" with responseStatusCode TBD (Method not allowed).
   b. "Send Response primitive".

No change for the remaining steps from the generic procedures in clause 7.2.1.2.2.

7.3.11. Resource Type request

7.3.11.1. Introduction

The <request> resource is used to represent information on locally issued requests (i.e. issued by an AE or CSE internal). This allows for robust synchronous and asynchronous request processing coping with various constraints on maximum blocking time. When an AE or CSE issues a request for targeting any other resource type or requesting a notification in non-blocking mode , i.e. the ***rt*** parameter of the request is set to either 'nonBlockingRequestSynch' or 'nonBlockingRequestAsynch', and if the Registrar CSE of the Originator supports the <request> resource type as indicated by the 'supportedResourceType' attribute of the <CSEBase> resource representing the Registrar CSE of the Originator, the Registrar CSE of the Originator shall create an instance of <request> to capture and expose the context of the associated non-blocking request. The detailed description can be found in clause 9.6.12 in Architecture TS.

Table 7.3.11.1-1: Data Type Definition of request

Data Type ID	File Name	Note
requestType	*CDT-request-v1_0_0-<<date of published>>.xsd*	

Table 7.3.11.1-2 : Applicable Common Attributes on <request>

Attribute Name	Request Optionality				Default Value	Resource Specific Note
	C	R	U	D		
resourceType	NP	O	NP	NP	None	
resourceID	NP	O	NP	NP	None	
expirationTime	NP	O	NP	NP	None	
parentID	NP	O	NP	NP	None	
creationTime	NP	O	NP	NP	None	
lastModifiedTime	NP	O	NP	NP	None	
accessControlPolicyIDs	NP	O	NP	NP	None	
labels	NP	O	NP	NP	None	
stateTag	NP	O	NP	NP	None	

Table 7.3.11.1-3 : Data Types for <request> resource specific attributes

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
operation	NP	O	NP	NP	m2m:operation	None
target	NP	O	NP	NP	xs:anyURI	None
originator	NP	O	NP	NP	xs:anyURI	None
requestIdentifier	NP	O	NP	NP	m2m:requestId	None
metaInformation	NP	O	NP	NP	m2m:metaInformation	None
content	NP	O	NP	NP	m2m:content	None
requestStatus	NP	O	NP	NP	m2m:requestStatus	None
operationResult	NP	O	NP	NP	m2m:operationResult	None

Table 7.3.11.1-4 : Reference of child resources

Child Resource Type Name	Child Resource Name	Multiplicity	Ref. to in Resource Type Definition
subscription	[variable]	0..n	Clause 7.3.7

7.3.11.2. Resource Specific Procedure on CRUD Operations

This clause describes request resource specific procedure on Resource Hosting CSE for CRUD operations.

7.3.11.2.1. Create

The <request> resource shall not be created via API.

The Receiver CSE of a non-blocking Request that was issued by either a Registrar AE of the Receiver CSE or a Registree/Registrar CSE of the Receiver CSE is the Hosing CSE for the <request> resource that shall be associated with the non-blocking request.

Hosting CSE: The Hosting CSE shall execute the following steps in order.

Hst-C-Rq-1.0 Assign an identifier to the <request> resource to be created.

Hst-C-Rq-2.0 Assign a value to the following common attributes:

1) parentID;

2) creationTime;

3) expirationTime: The Receiver shall assign a value that is consistent with the **rqet**, **rc**, **rset** and **rp** parameters effective for the associated non-blocking request that implied the creation of this <request> resource (within the restriction of the Receiver policies). If a value consistent with the **rqet**, **rc**, **rset** and **rp** parameters effective for the associated non-blocking request that implied the creation of this <request> resource cannot be supported, due to either policy or subscription restrictions, the Receiver will assign a new value.

4) lastModifiedTime: which is equals to the creationTime;

5) stateTag;

6) accessControlPolicyIDs: Populate with one ID of an <accessControlPolicy> that contains the following:

   a. In the 'privileges' child resource:

      i. Allow RUD operations to <request> resource being created to the Hosting CSE.

      ii. Allow RD operations to this <request> resource being created to the Originator of the associated non-blocking request, i.e. the value of the parameter ***fr*** in the associated non-blocking request that implied the creation of this <request> resource.

   b. In the 'selfPrivileges' child resource:

      i. Allow U operations the parent <accessControlPolicy> resource to the Originator of the associated non-blocking request, i.e. the value of the parameter fr in the associated non-blocking request that implied the creation of this <request> resource.

Hst-C-Rq-3.0 Assign any other RO (Read Only) attributes of <request> resource type within the restriction of the Receiver policies:

1) operation: Value of the parameter op in the associated non-blocking request that implied the creation of this <request> resource;

2) target: Value of the parameter to in the associated non-blocking request that implied the creation of this <request> resource;

3) originator: Value of the parameter fr in the associated non-blocking request that implied the creation of this <request> resource;

4) requestIdentifier: Value of the parameter ri in the associated non-blocking request that implied the creation of this <request> resource;

5) metaInformation: The content of this attribute is set to information in any other optional parameters described in clause 8.1. given in the associated non-blocking request that implied the creation of this <request> resource;

6) content: Value of the parameter cn - if any - in the associated non-blocking request that implied the creation of this <request> resource;

7) requestStatus: Information on the initial status of the associated non-blocking request that implied the creation of this <request> resource. The initial value of this attribute shall be identical to the status that is contained in the Acknowledgement response message of the associated non-blocking request. Possible values for status information contained in this attribute are specified in TS-0004. The value of this attribue is subject to changes according to the progress in processing of the non-blocking request that implied the creation of this <request> resource;

8) operationResult: Initially Empty. This attribute will be used for respresenting the result of the originally requested operation - if any - in line with the **rc** parameter in the associated non-blocking request that implied the creation of this <request> resource.

Hst-C-Rq-4.0 The Hosting CSE shall create the <request> resource.

Receiver: The Receiver shall execute the following steps in order.

Rcv-C-Rq-1.0 "Create an unsuccessful Response" with responseCode TBD (Method not allowed).

Rcv-C-Rq-2.0 "Send Response primitive".

7.3.11.2.2. Retrieve

Originator: the procedure of the Originator is the same as the clause 7.2.2.

Receiver: the procedure of the Receiver is the same as the clause 7.2.2.

7.3.11.2.3. Update

The <request> resource shall not be updated via API.

For a <request> resource explicit update requests shall not be supported. Changes in the attributes of a <request> resources can only be done by the Hosting CSE due to changes of the status of the associated non-blocking request that implied the creation of this <request> resource or due to reception of an operation result in response to the associated non-blocking request that implied the creation of this <request> resource.

Receiver: the Receiver shall execute the ollowing steps in order.

Rcv-U-Rq-1.0 "Create an unsuccessful Response" with responseCode TBD (Method not allowed).

Rcv-U-Rq-2.0 "Send Response primitive".

7.3.11.2.4. Delete

Originator: the procedure of the Originator is the same as the clause 7.2.2.

Receiver: the procedure of the Receiver is the same as the clause 7.2.2.

Editor's Note: Need to define how to cancel request resource.

7.3.12. Resource Type group

7.3.12.1. Introduction

The <group> resource represents a group of resources of the same or mixed types. The <group> resource can be used to do bulk manipulations on the resources represented by the membersList attribute. The <group> resource contains an attribute that represents the members of the group and a virtual resource (the <fanOutPoint>) that allows operations to be applied to the resources represented by those members. The detailed description can be found in clause 9.6.13 in Architecture TS.

Table 7.3.12.1-1: Data Type Definition of <group>

Data Type ID	File Name	Note
groupType	CDT-group-v1_0_0-<<date of published>>.xsd	

Table 7.3.12.1-2: Common Attributes on <group>

Attribute Name	Request Optionality				Default Value	Resource Specific Note
	C	R	U	D		
resourceType	NP	O	O	NP		
resourceID	NP	O	O	NP		
parentID	NP	O	NP	NP		
accessControlPolicyIDs	O	O	NP	NP		
creationTime	NP	O	NP	NP		
expirationTime	O	O	O	NP		
lastModifiedTime	NP	O	NP	NP		
labels	O	O	O	NP		
announceTo	O	O	O	NP		
announcedAttribute	O	O	O	NP		

Table 7.3.12.1-3: Resource Specific Attributes on <group>

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
memberType	M	O	O	NP	m2m:memberType	
currentNrOfMembers	NP	O	NP	NP	xs:integer	
maxNrOfMembers	M	O	O	NP	xs:integer	
membersList	M	O	O	NP	List of m2m:anyURI	
membersAccessControlPolicyIDs	O	O	O	NP	xs:anyURI	
memberTypeValidated	NP	O	NP	NP	xs:boolean	
consistencyStrategy	O	O	NP	NP	m2m:consistencyStrategy	
groupName	O	O	O	NP	xs:string	
creator	O	O	NP	NP	xs:anyURI	

Table 7.3.12.1-4: Reference of child resources

Child Resource Type	Child Resource Name	Multiplicity	Ref. to in Resource Type Definition
<subscription>	[variable]	0..n	7.3.7
<fanOutPoint>	fanOut (fixed)	1	7.3.13

7.3.12.2. Resource Specific Procedure on CRUD Operations

This clause describes <group> resource specific procedure on Resource Hosting CSE for CRUD operations.

7.3.12.2.1. Create

Primitive specific operation after Recv-C-6.4 "Check validity of resource representation for the given resource type" and before Recv-C-6.5 "Create/Update/Retrieve/Delete/Notify operation is performed ". See clause 7.2.1.2.2.

1) Primitive specific operation: Validate the provided attributes. It shall also check whether the number of URIs present in the *membersList* attribute of the group resource representation does not exceed the maximum as specified by the attribute *maxNrOfMembers*. If the maximum is exceeded, the request shall be rejected with a response status code ”Max number of member exceeded”..
   If the *memberType* attribute of the <group> resource is not "MIXED", the hosting CSE shall also verify that all the member URIs including sub-groups in the attribute *membersList* of the group resource representation provided in the request shall conform to the *memberType* of the group resource.

2) In the case that the <group> resource contains sub-group member resources, the receiver shall retrieve the ***memberType*** of the sub-group member resources to validate the ***memberType***. If the ***memberType*** cannot be retrieved due to lack of privilege, the request shall be rejected with a ”Retrieve error - no privilege”. If the sub-group member resources are temporarily unreachable, the receiver shall set the ***memberTypeValidated*** attribute of the <group> resource to FALSE and return the result to the originator in the response of the request. As soon as any unreachable sub-group resource becomes reachable, the receiver shall perform the ***memberType*** validation procedure. The originator may get to know the validation result by subscribe to the created resource if the ***memberTypeValidated*** attribute is FALSE. Upon unsuccessful validation, the receiver shall delete the <group> resource if the ***consistencyStrategy*** of the <group> resource is ABANDON_GROUP, or remove the inconsistent members from the <group> resource if the ***consistencyStrategy*** attribute is ABANDON_MEMBER, or set the ***memberType*** attribute of the <group> resource to "MIXED" if the ***consistencyStrategy*** attribute is MODIFY_TYPE.
   The ***memberTypeValidated*** attribute shall be set to TRUE if all the members have been validated successfully.

7.3.12.2.2. Retrieve

No primitive specific operations.

7.3.12.2.3. Update

1) Primitive specific operation after Recv-6.4 "Check validity of resource representation for the given resource type" and before Recv-6.5 " Create/Update/Retrieve/Delete/Notify operation is performed ". See clause 7.2.1.2.2.Primitive specific operation: If the *memberType* attribute of the <group> resource is not "MIXED", the hosting CSE shall verify that all the member URIs including sub-groups in the attribute *membersList* of the group resource representation provided in the request shall conform to the *memberType* of the group resource.

2) In the case that the <*group*> resource contains sub-group member resources, the receiver shall retrieve the ***memberType*** of the sub-group member resource to validate the ***memberType***. If the memberType cannot be retrieved due to lack of privilege, the request shall be rejected with a "Retrieve error - no privilege". If the sub-group member resources are temporarily unreachable, the receiver shall set the *memberTypeValidated* attribute of the <*group*> resource to FALSE and return the result to the originator in the response of the request. As soon as any unreachable sub-group resource becomes reachable, the receiver shall perform the *memberType* validation procedure. The originator may get to know the validation result by subscribe to the created resource if the *memberTypeValidated* attribute is FALSE. Upon unsuccessful validation, the receiver shall delete the <*group*> resource if the *consistencyStrategy* of the <*group*> resource is ABANDON_GROUP, or remove the inconsistent members from the <*group*> resource if the *consistencyStrategy* attribute is ABANDON_MEMBER, or set the *memberType* attribute of the <*group*> resource to "MIXED" if the *consistencyStrategy* attribute is MODIFY_TYPE.
The *memberTypeValidated* attribute shall be set to TRUE if all the members have been validated successfully.

3) Primitive specific operation: The hosting CSE shall check whether the number of provided *membersList* in the attribute members exceeds the limitation of *maxNrOfMembers*. If it exceeds, the hosting CSE shall reject the request with STATUS_NOT_ALLOWED.

7.3.12.2.4. Delete

No primitive specific operations.

7.3.13. Resource Type fanOutPoint

7.3.13.1. Introduction

The <fanOutPoint> resource is a virtual resource because it does not have a representation. It is the child resource of a <group> resource. Whenever the request is sent to the <fanOutPoint> resource, the request is fanned out to each of the members of the <group> resource indicated by the membersList attribute of the <group> resource. The responses (to the request) from each member are then aggregated and returned to the Originator. The detailed description can be found in clause 9.6.14 in Architecture TS.

There is no common attributes, resource specific attributes or xsd file to <fanOutPoint> resource because it's a virtual resource.

7.3.13.2. fanOutPoint operations

7.3.13.2.1. Validate the member types

Validate the provided attributes. If the memberType attribute of the addressed parent resource is not "MIXED", the group hosting CSE may check whether the type of resource to be created is consistent with the addressed parent resource. i.e. if the to parameter was …/fanOutPoint without any suffix, then the memberType attribute of the parent group resource determines the type of the addressed resource. Otherwise it is determined by the combination of the memberType and the child resources addressed in the to parameter after the fanOutPoint element in the path. If they are not consistent, the request shall be rejected with a "Member type inconsistent"..

7.3.13.2.2. Sub-group creation for members residing on the same CSE

The group hosting CSE shall obtain URIs of addressed resources from the attribute membersList of the parent <group> resource. The group hosting CSE may determine that multiple member resources belong to the same remote member hosting CSE, and may perform as an Originator to request to create a sub-group containing the specific multiple

member resources in that member hosting CSE. This sub-group is created in the member hosting CSE as described in clause 7.3.12.2.1. The to parameter of this group Create request may be <memberHosting cseBase>/ <groupHosting remoteCse>/ or <memberHosting cseBase>/ etc. The group hosting CSE shall also provide fr parameter (i.e. group hosting CSE) and sub-group resource representation that contains a member attribute with all the members residing on the addressed member Hosting CSE. The sub-group representation may include the attribute accessControlPolicyIDs, so that the group hosting CSE has the access right to this sub-group. The ID of the sub-group may be proposed by the group hosting CSE and determined by the member hosting CSE or it may be given by the member hosting CSE.
If there is already a sub-group resource defined in the remote member hosting CSE, then the group hosting CSE may utilize the existing sub-group resource.

7.3.13.2.3. Assign URI for aggregation of notification

In the case the created resource is <subscription> resource, the group hosting CSE shall validate if the subscription resource in the received request contains an *notificationForwardingURI* attribute. On successful validation, the group hosting CSE shall assign a new *notificationForwardingURI* to the attribute for receiving the notifications. The group hosting CSE shall locally maintain the mapping of the new *notificationForwardingURI* and the former *notificationForwardingURI* if it exists.

7.3.13.2.4. Fanout Request to each member

For each member hosting CSE, the group hosting CSE shall perform the following steps:

a) The primitive attributes ***fr*** and ***to*** shall be mapped to the primitive attributes of the corresponding Request to be sent out to each member of the group. The primitive attribute ***fr*** shall be directly used. The prefix of primitive attribute ***to*** i.e. <URI of group resource>/fanOutPoint shall be replaced by each URI of member resources derived from the attribute *membersList* of the group resource, but excluding the member resources which construct a sub-group. For these members resources contained in a sub-group, the primitive ***to*** of the composed Request shall be <URI of sub-group resource>/fanOutPoint. The group hosting CSE shall execute "Compose Request primitives". In addition, the group hosting CSE shall generate a unique group request identifier, add it as a primitive attribute to the Request and locally store the group request identifier as per the local policy.

b) "Send the Request to the receiver CSE".

c) "Wait for Response primitives".

The procedures between group hosting CSE and member hosting CSEs shall comply with the corresponding creation procedures as described in clause 7. The detailed procedures are according to the type of Resource provided in the Request primitive. During fanOutPoint manipulation, the member hosting CSE receiving a Request send from the group hosting CSE shall check if the Request contains a ***gid*** parameter. If the Request contains a ***gid*** parameter, the member hosting CSE shall compare the ***gid*** parameter to the ***gid*** locally stored. If a match is found, the member hosting CSE shall reject the request with the response status code set to "group request identifier exists " in the Response primitive. Otherwise, the member hosting CSE shall continue with the operations according to the Request and locally store the ***gid*** parameter.

7.3.13.3. <fanOutPoint> Resource Specific Procedure on CRUD Operations

This sub-clause describes <fanOutPoint> resource specific behaviour for CRUD operations.

7.3.13.3.1. Create

The primitives create the content of all member resources belonging to an existing group resource.

Originator:

Primitive specific operation after Orig-1.0 "Compose Request primitive" and before Orig-2.0 "Send the Request to the receiver CSE": In the case the Originator wants to subscribe to all the member resources of the group and the originator wants the group hosting CSE to aggregate all the notifications come from its member hosting CSEs, the Originator shall include *notificationForwardingURI* attribute in the subscription resource.

Receiver:

Primitive specific operation after Recv-6.2 "Check existence of the addressed resource" and before Recv-6.3 "Check authorization of the Originator": The ***to*** parameter consists of the URI of the group resource plus a suffix marked by /fanOutPoint or /fanOutPoint/.....

Primitive specific operation additional to Recv-6.3 "Check authorization of the Originator": The Group Hosting CSE shall check the authorization of the Originator based on the *membersAccessControlPolicyIDs* of the parent group resource. In the case the *membersAccessControlPolicyIDs* is not provided, the *accessControlPolicyIDs* of the parent group resource shall be used.

Primitive specific operation to replace Recv-6.5 "Create/Update/Retrieve/Delete/Notify operation is performed" and Recv-6.6 "Announce/De-announce the resource" in the generic procedure:

1) Validate the member types, refer to7.3.12.2.1
2) Sub-group creation for members residing on the same CSE, refer to 7.3.12.2
3) Assign URI for aggregation of notification, refer to 7.3.12.3
4) Fanout Request to each member, refer to 7.3.12.4
5) The group hosting CSE shall aggregate the Responses after receiving responses from its member resources and sub-groups and aggregate the Responses into a single Response:

Primitive specific operation additional to Recv-6.7 "Create a successful Response", the Response shall include the aggregated Responses.

7.3.13.4. Retrieve

The primitives retrieve the content of all member resources belonging to an existing group resource.

Originator:

No primitive specific operations.

Receiver:

Primitive specific operation after Recv-6.2 "Check existence of the addressed resource" and before Recv-6.3 "Check authorization of the Originator": The ***to*** parameter consists of the URI of the group resource plus a suffix marked by /fanOutPoint or /fanOutPoint/.....

Primitive specific operation additional to Recv-6.3 "Check authorization of the Originator": The Group Hosting CSE shall check the authorization of the Originator based on the *membersAccessControlPolicyIDs* of the parent group resource. In the case the *membersAccessControlPolicyIDs* is not provided, the *accessControlPolicyIDs* of the parent group resource shall be used.

Primitive specific operation to replace Recv-6.5 "Create/Update/Retrieve/Delete/Notify operation is performed" and Recv-6.6 "Announce/De-announce the resource" in the generic procedure:

1) Sub-group creation for members residing on the same CSE, refer to 7.3.12.2
2) Fanout Request to each member, refer to 7.3.12.4
3) The group hosting CSE shall aggregate the Responses after receiving responses from its member resources and sub-groups and aggregate the Responses into a single Response:

Primitive specific operation additional to Recv-6.7 "Create a successful Response", the Response shall include the aggregated Responses.

7.3.13.4.1. Update

The primitives update the contnt of all member resources belonging to an existing group resource.

Originator:

No primitive specific operations.

Receiver:

Primitive specific operation after Recv-6.2 "Check existence of the addressed resource" and before Recv-6.3 "Check authorization of the Originator": The ***to*** parameter consists of the URI of the group resource plus a suffix marked by /fanOutPoint or /fanOutPoint/.....

Primitive specific operation additional to Recv-6.3 "Check authorization of the Originator": The Group Hosting CSE shall check the authorization of the Originator based on the *membersAccessControlPolicyIDs* of the parent group resource. In the case the *membersAccessControlPolicyIDs* is not provided, the *accessControlPolicyIDs* of the parent group resource shall be used.

Primitive specific operation to replace Recv-6.5 "Create/Update/Retrieve/Delete/Notify operation is performed" and Recv-6.6 "Announce/De-announce the resource" in the generic procedure:

1) Validate the member types ., refer to 7.3.12.1
2) Sub-group creation for members residing on the same CSE , refer to 7.3.12.2
3) Fanout Request to each member ., refer to 7.3.12.4
4) The group hosting CSE shall aggregate the Responses after receiving responses from its member resources and sub-groups and aggregate the Responses into a single Response:

Primitive specific operation additional to Recv-6.7 "Create a successful Response", the Response shall include the aggregated Responses.

7.3.13.4.2. Delete

The primitives delete the content of all member resources belonging to an existing group resource.

Originator:

No primitive specific operations.

Receiver:

Primitive specific operation after Recv-6.2 "Check existence of the addressed resource" and Recv-6.3 "Check authorization of the Originator": The ***to*** parameter consists of the URI of the group resource plus a suffix marked by /fanOutPoint or /fanOutPoint/.....

Primitive specific operation additional to Recv-6.3 "Check authorization of the Originator": The Group Hosting CSE shall check the authorization of the Originator based on the *membersAccessControlPolicyIDs* of the parent group resource. In the case the *membersAccessControlPolicyIDs* is not provided, the *accessControlPolicyIDs* of the parent group resource shall be used.

Primitive specific operation to replace Recv-6.5 "Create/Update/Retrieve/Delete/Notify operation is performed" and Recv-6.6 "Announce/De-announce the resource" in the generic procedure:

1) Validate the member types , refer to 7.3.12.1
2) Sub-group creation for members residing on the same CSE , refer to 7.3.12.2
3) Fanout Request to each member , refer to 7.3.12.4
4) The group hosting CSE shall aggregate the Responses after receiving responses from its member resources and sub-groups and aggregate the Responses into a single Response:

Primitive specific operation additional to Recv-6.7 "Create a successful Response", the Response shall include the aggregated Responses.

7.3.14. Resource Type mgmtObj

7.3.14.1. Introduction

The mgmtObj resource contains management data which represents individual M2M management functions. It represents a general structure to map to external management technology data models.

Table 7.3.14.1-1: Data Type Definition of <mgmtObj>

Data Type ID	File Name	Note
mgmtObjType	CDT-<<resource name>>-v1_0_0-<<date of published>>.xsd	

Table 7.3.14.1-2: Common Attributes on <mgmtObj>

Attribute Name	Request Optionality				Default Value	Resource Specific Note
	C	R	U	D		
resourceType	NP	O	O	NP		
resourceID	NP	O	O	NP		
parentID	NP	O	NP	NP		
accessControlPolicyIDs	O	O	NP	NP		
creationTime	NP	O	NP	NP		
expirationTime	O	O	O	NP		
lastModifiedTime	NP	O	NP	NP		
labels	O	O	O	NP		

Table 7.3.14.1-3: Resource Specific Attributes on <mgmtObj>

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
objectIDs	O	O	NP	NP	List of xs:string	
objectPaths	O	O	NP	NP	List of xs:string	
mgmtDefinition	M	O	NP	NP	m2m:mgmtDefinition	
mgmtLink	O	O	O	NP	List of m2m:anyURI	
description	O	O	O	NP	xs:string	

Table 7.3.14.1-4: Child resources of <mgmtObj>

Child Resource Type	Child Resource Name	Multiplicity	Ref. to Resource Type Definition
<subscription>	[variable]	0..n	7.3.7

7.3.14.2. Resource Specific Procedure on CRUD Operations

This clause describes <mgmtObj> resource specific procedure on resource Hosting CSE for CRUD operations.

7.3.14.2.1. Create

Primitive specific operation before Orig-C-1.0 “Compose Request primitive”:

1) Primitive specific operation: If the originator is the managed entity, it shall generate the <mgmtObj> resource representation based on the external management object information of the managed entity to be exposed. The objectID and objectPath attribute may be set with the Request.

Primitive specific operation after Recv-6.5 “Create/Update/Retrieve/Delete/Notify operation is performed” and before Recv-6.6 “Announce/De-announce the resource” if the originator is an IN-AE:

1) "Identify the managed entity and the management protocol ".

Primitive specific operation: the receiver shall generate the external management object to be added to the managed entity based on the <mgmtObj> resource representation provided in the Request primitive. The receiver may determine the target location on the managed entity where the generated external management object shall be added based on the "objectID" and "objectPath" provided in the request primitive and the protocol specific data model being used. The receiver may also choose to let the managed entity decide the target location where the generated external management object shall be added using protocol specific mechanism.

1) "Establish a management session with the managed entity".

2) "Send the management request(s) to the managed entity corresponding to the received Request primitive ". If the receiver receives an error response from the managed entity because the external management object to be added already exists on the managed entity, the receiver shall check (by using e.g. OMA-DM Get command or TR069 GetParameterValues/GetParameterAttributes command) if the existing external management object is the same as the one to be added, then it shall consider the requested primitive as successfully performed instead of sending an unsuccessful Response primitive; otherwise, it shall reject the request with the response status code set to " Create error - already exists " in the Response primitive. The receiver shall also record the location where the external management object is added to the managed entity in the successful case. The objectID and objectPath attribute may be set with the Request.

3) The receiver may repeat Step 4 in order to add to the managed entity the external management objects that are mapped from the mandatory sub-resources (including any descendents) that are required to be created automatically with the default attribute values.

7.3.14.2.2. Retrieve

Primitive specific operation after Recv-6.5 “Create/Update/Retrieve/Delete/Notify operation is performed” and before Recv-6.6 “Announce/De-announce the resource” if the originator is an IN-AE:

1) "Identify the managed entity and the management protocol".

2) "Locate the external management objects to be managed on the managed entity".

3) "Establish a management session with the managed entity".

4) "Send the management request(s) to the managed entity corresponding to the received Request primitive". The receiver may also update the <mgmtObj> resource representation with the retrieved external management object information if required according to the local policy.

7.3.14.2.3. Update

The Update primitive is used for the update of the resource as well as the execution of a management procedure. The execution is performed using an Update primitive which without any content as the payload part of the primitive by addressing specific attribute to start the management procedure.

Primitive specific operation after Recv-6.5 “Create/Update/Retrieve/Delete/Notify operation is performed” and before Recv-6.6 “Announce/De-announce the resource” if the originator is IN-AE.

1) "Identify the managed entity and the management protocol".

2) "Locate the external management objects to be managed on the managed entity".

3) "Establish a management session with the managed entity".

4) "Send the management request(s) to the managed entity corresponding to the received Request primitive". The receiver may also update the <mgmtObj> resource representation with the retrieved external management object information if required according to the local policy.

7.3.14.2.4. Delete

Primitive specific operation after Recv-6.5 “Create/Update/Retrieve/Delete/Notify operation is performed” and before Recv-6.6 “Announce/De-announce the resource” if the originator is IN-AE.

1) "Identify the managed entity and the management protocol".

2) "Locate the external management objects to be managed on the managed entity".

3) "Establish a management session with the managed entity".

4) "Send the management request(s) to the managed entity corresponding to the received Request primitive". The receiver may also update the <mgmtObj> resource representation with the retrieved external management object information if required according to the local policy.

7.3.15. Resource Type mgmtCmd

7.3.15.1. Introduction

The <mgmtCmd> resource shall contain the following attributes and child resource as illustrated in table 7.3.15.1-2 and table 7.3.15.1-3. The data type and default value of these attributes and child resources are included in the tables.

Table 7.3.15.1-1: Data Type Definition of <mgmtCmd>

Data Type ID	File Name	Note
Actual Data Type ID	CDT-<<resource name>>-v1_0_0-<<date of published>>.xsd	

Table 7.3.15.1-2: Data Types for resource specific attributes

Attribute Name	Data Type	Default	Value Restrictions	Notes
description	xs:string	Not applicable	size: 256	
cmdType	m2m:cmdType	Not applicable	RESET, REBOOT, UPLOAD, DOWNLOAD, SOFTWAREINSTALL, SOFTWAREUNINSTALL	
execReqArgs	m2m: execReqArgsListType	Not applicable	A list of entries which are dependent on execReqArgsListType. If cmdType=Reset, execReqArgsListType=ResetArgsType. If cmdType=REBOOT, execReqArgsListType=RebootArgsType. If cmdType=UPLOAD, execReqArgsListType=UploadArgsType. If cmdType=DOWNLOAD, execReqArgsListType=DownloadArgsType. If cmdType=SOFTWAREINSTALL, execReqArgsListType=SoftwareInstallArgsType. If cmdType= SOFTWAREUNINSTALL, execReqArgsListType=SoftwareUninstallArgsType. For different execReqArgsListType, the list of entries is different which are described in corresponding .xsd file.	
execEnable	xs: anyURI	Not applicable		
execTarget	m2m:anyURI	Not applicable		
execMode	m2m:execModeType	IMMEDIATEONCE	IMMEDIATEONCE, IMMEDIATEREPEAT, RANDOMONCE, RANDOMREPEAT	
execFrequency	xs:duration	Not applicable		
execDelay	xs:duration	0		
execNumber	xs: nonNegativeInteger	1		

Table 7.3.15.1-3: Reference of child resources

Child Resource Type Name	Data Type ID	Ref. in ArchTS
<execInstance>	xs:anyURI	9.6.17

The <mgmtCmd> shall be executed for the following modes:

- If execMode is IMMEDIATEONCE, <mgmtCmd> shall be executed immediately and only once. In this mode, execFrequency, execDelay, and execNumber shall not be used.
- If execMode is IMMEDATEREPEAT, <mgmtCmd> shall be executed immediately and repeated multiple times as determined by execNumber and the time interval between each execution is specified by execFrequency. In this mode, execDelay shall not be used.
- If execMode is RANDONONCE, <mgmtCmd> shall be executed only once at a delayed time which is specified by execDelay. In this mode, execFrequency and execNumber shall not be used.
- If execMode is RANDOMREPEAT, <mgmtCmd> shall be executed multiple times as specified by execNumber but the first execution shall be executed at a delayed time. execDelay specifies the delayed time. The time interval between each execution is specified by execFrequency.

7.3.15.2. Resource Specific Procedures

This clause describes <mgmtCmd> resource specific procedures for CRUD operations.

7.3.15.2.1. Create

This procedure shall use the Create common operations detailed in clause 7.2.1.2.1 without primitive specific actions. The Originator shall use the steps Orig-C-1.0, Orig-C-2.0, and Orig-C-3.0 as described in clause 7.2.1.2.1. The Receiver shall use the steps Rcv-C-1.0 to Rcv-C-11.0 as described in clause 7.2.1.2.1.

The Originator shall provide the <mgmtCmd> resource representation to the Receiver (e.g. IN-CSE). The Receiver may generate one of the following status codes and send it to the Originator.

If the Originator provides an invalid cmdType value in the Create primitive, the Receiver shall generate the status code "Create mgmtCmd – invalid cmdtype".

If the name/value entry in execReqArgs does not match the value of cmdType in the Create primitive, the Receiver shall generate the status code "Create mgmtCmd – invalid arguments".

If the name/value entries in execReqArgs do not contain mandatory arguments as required by cmdType, the Receiver shall generate the status code "Create mgmtCmd – insufficient arguments"

7.3.15.2.2. Retrieve

This procedure shall use the Retrieve common operations detailed in clause 7.2.2 without primitive specific actions. The Originator shall use the steps Orig-R-1.0, Orig-R-2.0, and Orig-R-3.0 as described in clause 7.2.2. The Receiver shall use the steps Rcv-R-1.0 to Rcv-R-9.0 as described in clause 7.2.2.

7.3.15.2.3. Update

7.3.15.2.3.1 Update (Normal)

If the Update primitive does not address the execEnable attribute of the <mgmtCmd> or the URI provided as the value of the execEnable , it results in update of all or part of the information of an existing <mgmtCmd> resource with the new attribute values. The procedure uses the common Update operations detailed in clause 7.2.2, without primitive specific actions.

The Originator shall use the steps Orig-U-1.0, Orig-U-2.0, and Orig-U-3.0 as described in clause 7.2.2. The Receiver shall use the steps Rcv-U-1.0 to Rcv-U-11.0 as described in clause 7.2.2.

If the Originator attempts to update attributes resorceType, resourceID or cmdType, the Receiver shall generate the status code “Update error – no privilege”.

If the Originator attempts to update attributes execTarget, execMode, but the <mgmtCmd> has child resource <execInstance> already created, the Receiver shall generate the status code “Update error - unacceptable contents”.

If the Originator attempts to update attributes any attribute with a value which is not allowed, the Receiver shall generate the status code “Update error - unacceptable contents”.

If the Update primitive for <mgmtCmd> does address the execEnable attribute of the <mgmtCmd>, it effectively triggers an Execute <mgmtCmd> procedure, see clause 7.3.15.2.3.2.

7.3.15.2.3.2 Update (Execute)

The execute operation is triggered by an Update primitive, if the primitive addresses the execEnable attribute of the <mgmtCmd> or the URI provided as the value of the execEnable. The procedure uses the Update common operations detailed in clause 7.2.2 with the following primitive specific operation after Rcv-U-4.0 and before Rcv-U-5.0:

1) The Receiver shall identify the managed entity and the management protocol. The execTarget attribute of <mgmtCmd> indicates the managed entity.

The Receiver shall automatically create an <execInstance> based on the <mgmtCmd> resource. If the execTarget attribute addresses a <group> resource, the Receiver shall create multiple <execInstance> sub-resources based on the value of currentNrOfMembers attribute.

The Receiver shall copy the following attributes from <mgmtCmd> to each <execInstance> created: execMode, execFrequency, execDelay, execNumber, and execReqArgs. The execStatus of <execInstance> is set as INITIATED. The Receiver shall set the execTarget attribute of each <execInstance> sub-resource to the URI of each target <node> resource.

The Receiver shall determine if the <mgmtCmd> shall be executed immediately or postponed according to the combination of execMode, execFrequency, execDelay, and execNumber. If the <mgmtCmd> shall be executed immediately (e.g. execMode is IMMEDIATEONCE), the following steps shall be performed; otherwise the following steps shall be postponed and skipped until the delay is expired (e.g. as indicated by execDelay).

The Receiver shall establish a management session with the identified managed entity.

The Receiver shall perform management command conversion and execution and set the execStatus attribute of <execInstance> to PENDING. If the Receiver cannot perform the command conversion successfully (e.g. execReqArgs does not have sufficient name/value pairs), the Receiver shall generate status code “MgmtCmd – conversion error”.

After receiving completion response from the managed entity, the Receiver shall set execStatus attribute of corresponding <execInstance> to FINISHED.

If the Update primitive for <mgmtCmd> does not address the execEnable attribute of the <mgmtCmd>, it effectively triggers an Update <mgmtCmd> procedure, see clause 7.3.15.2.3.1.

7.3.15.2.4. Delete

This procedure is based on the Delete common operations detailed in clause 7.2.2.

The Receiver shall determine:

- If there are related management operations pending on the managed entity by checking if the execStatus attribute of all <execInstance> sub-resources are PENDING.
- If the related management operations are cancellable by checking the cmdType attribute of <mgmtCmd>.

If there are no management commands pending on the remote entity the Receiver shall delete the addressed <mgmtCmd> resource and send a successful response to the Originator.

If there are cancellable management commands still pending on any remote entity, the Receiver shall perform the following steps:

1) The Receiver shall identify the managed entity and the management protocol. The execTarget attribute of each <execInstance> sub-resource which has execStatus of PENDING indicates the managed entity.

2) The Receiver shall establish a management session with each managed entity.

3) The Receiver shall perform management command conversion and execution resulting in cancellation of the commands which are pending on the managed entity.

4) For each successful cancellation RPC the execStatus attribute of the corresponding <execInstance> is set to CANCELLED. For each un-successful cancellation RPCs the execStatus attribute of the corresponding <execInstance> is determined from the reported fault codes for the unsuccessful RPCs.

5) Upon completion of all the cancellation operations, if any fault codes are returned by the managed entity, an unsuccessful Response to the Delete primitive with status code "Delete mgmtCmd- execInstance cancellation error" is returned, and the <mgmtCmd> resource is not deleted. 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 are non-cancellable management commands still pending on the remote entity, the Receiver shall send an un-successful Response to the Delete request to the Originator, with the status code "Delete mgmtCmd- execInstance cancellation error". The execStatus attribute of the specific <execInstance> sub-resource is changed to STATUS_NON_CANCELLABLE.

7.3.16. Resource Type execInstance

7.3.16.1. Introduction

The <execInstance> resource shall contain the following child resource and attributes.

Table 7.3.16.1-1: Data Type Definition of <execInstance>

Data Type ID	File Name	Note
Actual Data Type ID	CDT-<<resource name>>-v1_0_0-<<date of published>>.xsd	

Table 7.3.16.1-2: Data Types for resource specific attributes

Attribute Name	Data Type	Default	Value Restrictions	Notes
execStatus	m2m:execStatusType	INITIATED	INITATED, PENDING, FINISHED, CANCELLING, CANCELLED	
execResult	xs:execResultType	Not applicable		
execDisable	xs:anyURI	Not applicable		
execTarget	m2m:nodeID	Not applicable		
execMode	m2m:execModeType	IMMEDIATEONCE	IMMEDIATEONCE, IMMEDIATEREPEAT, RANDOMONCE, RANDOMREPEAT	
execFrequency	xs:duration	Not applicable		
execDelay	xs:duration	0		
execNumber	xs:nonNegativeInteger	1		
execReqArgs	m2m:execReqArgsListType	Not applicable		

7.3.16.2. Resource Specific Procedures

This clause describes <execInstance> resource specific procedures for CRUD operations.

7.3.16.2.1. Update (Cancel)

The <execInstance> Cancel operation is triggered by an Update primitive, if the primitive addresses the execDisable attribute or the URI provided as the value of the execDisable. The procedure is based on Update common operations detailed in clause 7.2.2.

The Receiver shall determine:

- If there are related management operations pending on the managed entity by checking the execStatus attribute of the addressed <execInstance> sub-resource is PENDING.
- If the related management operations are cancellable by checking the cmdType attribute of the parent <mgmtCmd> resource.

If the are no management commands still pending on the remote entity, an unsuccessful Response to the Update primitive with status code "Cancel execInstance – already complete" is returned to the Originator.

If there are cancellable management commands still pending on the remote entity, the Receiver shall perform the following steps:

1) The Receiver shall identify the managed entity and the management protocol. The execTarget attribute of the addressed <execInstance> indicates the managed entity.
2) The Receiver shall establish a management session with the managed entity.
3) The Receiver shall perform management command conversion and execution resulting in cancellation of the commands which are pending on the managed entity.
4) If the cancellation is successfully executed on the managed entity, the Receiver shall return a successful Response to the Originator and shall set execStatus of <execInstance> to CANCELLED.
5) If the cancellation is unsuccessful on the managed entity, the Receiver shall return an unsuccessful Response to the Originator with status code "Cancel execInstance – cancellation error". The execStatus attribute is determined from the fault codes reported by the managed entity.

If there are non-cancellable management commands still pending on the remote entity, the Receiver shall return an unsuccessful Response to the Originator with the status code "Cancel execInstance – not cancellable", and the execStatus attribute is changed to STATUS_NON_CANCELLABLE.

7.3.16.2.2. Retrieve

This procedure shall use the Retrieve common operations detailed in clause 7.2.2, without primitive specific actions. The Originator shall use the steps Orig-R-1.0, Orig-R-2.0, and Orig-R-3.0 as described in clause 7.2.1.2. The Receiver shall use the steps Rcv-R-1.0 to Rcv-R-9.0 as described in clause 7.2.2.

7.3.16.2.3. Delete

This procedure is based on the Delete common operations detailed in clause 7.2.2.

The Receiver shall determine:

- If there are related management operations pending on the managed entity by checking the execStatus attribute of the addressed <execInstance> sub-resource is PENDING.
- If the related management operations are cancellable by checking the cmdType attribute of the parent <mgmtCmd> resource.

If there are no management commands still pending on the remote entity, the Receiver shall delete the addressed resource and send a successful Response to the Originator.

If there are cancellable management commands still pending on the remote entity, the Receiver shall perform the following steps:

1) The Receiver shall identify the managed entity and the management protocol. The execTarget attribute of the addressed <execInstance> indicates the managed entity.

2) The Receiver shall establish a management session with the managed entity.

3) The Receiver shall perform management command conversion and execution resulting in cancellation of the commands which are pending on the managed entity.

4) If the cancellation is successfully executed on the managed entity, the Receiver shall return a successful Response to the Delete request to the Originator and shall delete the <execInstance> resource.

5) If the cancellation is unsuccessful on the managed entity, the Receiver shall return an unsuccessful Response to the Delete request to the Originator with status code "Delete execInstance – cancellation failed". The execStatus attribute is determined from the fault codes reported by the managed entity.

If there are non-cancellable management commands still pending on the remote entity, the Receiver shall return an unsuccessful Response to the Delete request to the Originator with status code "Delete execInstance – not cancellable". The execStatus attribute is set to STATUS_NOT_CANCELLABLE.

7.3.17. Resource Type node

7.3.17.1. Introduction

The <node> resource represents specific information that provides properties of an oneM2M Node that can be utilized by other oneM2M operations. The <node> resource has <mgmtObj> as its child resources.

Table 7.3.17.1-1: Data Type Definition of <node>

Data Type ID	File Name	Note
nodeType	CDT-<<resource name>>-v1_0_0-<<date of published>>.xsd	

Table 7.3.17.1-2: Common Attributes on <node>

Attribute Name	Request Optionality				Default Value	Resource Specific Note
	C	R	U	D		
resourceType	NP	O	O	NP		
resourceID	NP	O	O	NP		
parentID	NP	O	NP	NP		
accessControlPolicyIDs	O	O	NP	NP		
creationTime	NP	O	NP	NP		
expirationTime	O	O	O	NP		
lastModifiedTime	NP	O	NP	NP		
labels	O	O	O	NP		

Table 7.3.17.1-3: Resource Specific Attributes on <node>

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
nodeID	M	O	O	NP	xs:string	
hostedCSEID	O	O	NP	NP	xs:anyURI	

Table 7.3.17.1-4: Child resources of <node>

Child Resource Type	Child Resource Name	Multiplicity	Ref. to Resource Type Definition
<subscription>	[variable]	0..n	7.3.7

7.3.17.2. Resource Specific Procedure on CRUD Operations

This clause describes <node> resource specific procedure on Resource Hosting CSE for CRUD operations.

7.3.17.2.1. Create

No primitive specific operations.

7.3.17.2.2. Retrieve

No primitive specific operations.

7.3.17.2.3. Update

No primitive specific operations.

7.3.17.2.4. Delete

No primitive specific operations.

7.3.18. Resource Type m2mServiceSubscriptionProfile

7.3.18.1. Introduction

The <m2mServiceSubscriptionProfile> resource represents an M2M Service Subscription Profile. It is used to represent all data pertaining to the M2M Service Subscription Profile, i.e., the technical part of the contract between an M2M Application Service Provider and an M2M Service Provider.

The detailed description can be found in clause 9.6.19 in Architecture TS [6].

Table 7.3.18.1-1: Data Type Definition of <m2mServiceSubscriptionProfile>

Data Type ID	File Name	Note
m2mServiceSubscriptionProfileType	(TBD)	

Table 7.3.18.1-2: Common Attributes on <m2mServiceSubscriptionProfile>

Attribute Name	Request Optionality				Default Value	Resource Specific Note
	C	R	U	D		
resourceType	NP	O	NP	NP		
resourceID	NP	O	NP	NP		
parentID	NP	O	NP	NP		
expirationTime	O	O	O	NP		
accessControlPolicyIDs	O	O	O	NP		
creationTime	NP	O	NP	NP		
labels	O	O	O	NP		
lastModifiedTime	NP	O	NP	NP		

Table 7.3.18.1-3: Resource Specific Attributes on <m2mServiceSubscriptionProfile>

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
serviceRoles	O	O	O	NP	M2m:serviceRoles	

Table 7.3.18.1-4: Child resources of <authorizedNodeProfile>

Child Resource Type	Child Resource Name	Multiplicity	Ref. to Resource Type Definition
<subscription>	[variable]	0..n	Clause 7.3.7
<authorizedNode>	[variable]	0..n	Clause 7.3.19

7.3.18.2. Operations

This clause describes <m2mServiceSubscriptionProfile> resource specific behaviour for CRUD operations.

7.3.18.2.1. Create

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.3.18.2.2. Retrieve

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2..

7.3.18.2.3. Update

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2..

7.3.18.2.4. Delete

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2..

7.3.19. Resource Type authorizedNode

7.3.19.1. Introduction

The <authorizedNode> resource represents M2M Node information that is needed as part of the M2M Service Subscription resource. It shall contain information about the M2M Node as well as application identifiers of the Applications running on that Node.

The detailed description can be found in clause 9.6.20 in Architecture TS [6].

Table 7.3.19.1-1: Data Type Definition of <authorizedNode>

Data Type ID	File Name	Note
authorizedNodeType	(TBD)	

Table 7.3.19.1-2: Common Attributes on <authorizedNode >

Attribute Name	Request Optionality				Default Value	Resource Specific Note
	C	R	U	D		
resourceType	NP	O	NP	NP		
resourceID	NP	O	NP	NP		
parentID	NP	O	NP	NP		
expirationTime	O	O	O	NP		
accessControlPolicyIDs	O	O	O	NP		
creationTime	NP	O	O	NP		
labels	O	O	O	NP		
lastModifiedTime	NP	O	NP	NP		

Table 7.3.19.1-3: Resource Specific Attributes on <authorizedNode >

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
nodeID	M	O	O	NP	m2m:nodeId	
AE-IDs	O	O	O	NP	m2m:id	
CSE-ID	O	O	O	NP	m2m:id	

Table 7.3.19.1-4: Child resources of <authorizedNode>

Child Resource Type	Child Resource Name	Multiplicity	Ref. to Resource Type Definition
<subscription>	[variable]	0..n	7.3.7

7.3.19.2. Operations

This clause describes <authorizedNode> resource specific behaviour for CRUD operations.

7.3.19.2.1. Create

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.3.19.2.2. Retrieve

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2..

7.3.19.2.3. Update

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2..

7.3.19.2.4. Delete

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2..

7.3.20. Resource Type pollingChannel

7.3.20.1. Introduction

The <pollingChannel> resource is used to perform service layer long polling when an AE/CSE cannot receive a request from other entities, however it can get a request as a response to a long polling request. Actual long polling can be performed on the <pollingChannelURI> resource which is the child resource of the <pollingChannel> resource.

The detailed description can be found in clause 9.6.21 in TS-0001 [6].

Table 7.3.20.1-1: Data Type Definition of pollingChannel

Data Type ID	File Name	Note
pollingChannelType	TBD	

Editor's Note: Data Type definition should be posted after general agreement on technical issues.

Table 7.3.20.1-2: Reference of child resources

Child Resource Type	Child Resource Name	Data Type ID	Ref. in ArchTS
<pollingChannelURI>	pollingChannelURI	1	9.6.21

7.3.20.2. Operations

This clause describes <pollingChannel> resource specific behaviour for CRUD operations.

7.3.20.2.1. Create

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.3.20.2.2. Retrieve

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.3.20.2.3. Update

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.3.20.2.4. Delete

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.3.21. Resource Type pollingChannelURI

7.3.21.1. Introduction

The <pollingChannelURI> resource is the virtual child resource which is automatically generated during the parent <pollingChannel> resource creation. The detailed description can be found in clause 9.6.22 in TS-0001[6].

There is no data type definition for <pollingChannelURI> resource because it's a virtual resource type.

7.3.21.2. Operations

This clause describes <pollingChannelURI> resource specific behaviour for the Retrieve operation as a service layer long polling request. CUDN requests to the <pollingChannelURI> resource shall be rejected.

7.3.21.2.1. Create

The present document does not define Create operation on a <pollingChannelURI> resource. A Create request for the resource shall be rejected.

A <pollingChannelURI> virtual resource shall only be created during its parent <pollingChannel> resource creation procedure.

7.3.21.2.2. Retrieve

Originator: shall execute Originator actions in clause 7.2.1.2.1 as a service layer long polling request. It's the Originator's responsibility to initiate this procedure after it gets long polling response either successful or unsuccessful. The Originator shall send this Retrieve request as blocking request (clause 8.2.1 in [6]).

Receiver: shall execute the following steps in order and these are modifications to the generic procedure from Recv-6.3 to Recv-6.5 in clause 7.2.1.2.2:

Recv-6.3 Check if the request Originator is the *creator* of the parent <pollingChannel> resource. If it is not the creator, the Hosting CSE shall send access denied error response.

Recv-6.4 No change from the generic procedure.

Recv-6.5

If there is a pending request(s) to be sent to the Originator

Create a Response primitive including Pending Requests primitive parameter.

Else

Wait for a request for the Originator until the *rqet* of the Originator's request. If a request is available before the *rqet* timeout, create a Response primitive including Pending Requests primitive parameter. Otherwise, create an unsuccessful Response. Error information is request timeout.

7.3.21.2.3. Update

The present document does not define Update operation on a <pollingChannelURI> resource. An Update request for the resource shall be rejected.

7.3.21.2.4. Delete

The present document does not define Delete operation on a <pollingChannelURI> resource. A Delete request for the resource shall be rejected.

7.3.22. Resource Type statsConfig

7.3.22.1. Introduction

The <statsConfig> resource is used to store configuration data for collecting statistics for AEs. The <eventConfig> child resource is a mechanism for defining events that trigger statistics collection activity. Additional description of the <statsConfig> resource is contained in clauses 9.6.22 and 10.2.15 of oneM2M TS-0001 [6].

Table 7.3.22.1-1: Data Type definition of <statsConfig>

Data Type ID	File Name	Note
Actual Data Type ID	*CDT-statsConfig-v1_0_0-<<date of published>>.xsd*	

Table 7.3.22.1-2: Data Types for resource specific attributes of <statsConfig>

Attribute Name	Data Type	Default Value and Constraints	Entity which allows operation
None			

Table 7.3.22.1-3: Reference of child resources of <statsConfig>

Child Resource Type Name	Data Type ID	Ref. to in Resource Type Definition
<eventConfig>	list of xs:anyURI	7.3.30
<subscription>	list of xs:anyURI	7.3.6

7.3.22.2. <statsConfig> resource-specific procedure on CRUD operations

7.3.22.2.1. Create

This procedure follows the Generic Resource Create Request Procedure specified in clause 7.2.1.2.1 with the following <statsConfig> resource-specific updates.

Resource-specific operation before Rcv-C-5.0:

1) If the **To** primitive parameter addresses a receiver CSE that is not an IN-CSE, then the request shall be rejected with a "STATUS_BAD_REQUEST".

7.3.22.2.2. Retrieve

This procedure follows the Generic Resource Retrieve Request Procedure specified in clause 7.2.2.

7.3.22.2.3. Update

This procedure follows the Generic Resource Update Request Procedure specified in clause 7.2.2.

7.3.22.2.4. Delete

This procedure follows the Generic Resource Delete Request Procedure specified in clause 7.2.2.

7.3.23. Resource Type eventConfig

7.3.23.1. Introduction

The <eventConfig> resource defines events that trigger statistics collection activity on an IN-CSE. Additional description of the <eventConfig> resource is contained in clauses 9.6.23 and 10.2.15 of oneM2M TS-0001 [6].

Table 7.3.23.1-1: Data Type definition of <eventConfig>

Data Type ID	File Name	Note
Actual Data Type ID	CDT-eventConfig-v1_0_0-<<date of published>>.xsd	

Table 7.3.23.1-2: Resource-specific attributes of <eventConfig>

Attribute Name	Presence in CREATE request	Presence in UPDATE request	Presence in Response	Description
eventID	O	NP	M	see Table A-1
eventType	M	M	M	see Table A-1
eventStart	O	O	O	see Table A-1
eventEnd	O	O	O	see Table A-1
transactionType	O	O	O	see Table A-1
dataSize	O	O	O	see Table A-1

Table 7.3.23.1-3: Reference of child resources of <eventConfig>

Child Resource Type Name	Data Type ID	Ref. to in Resource Type Definition
<subscription>	list of xs:anyURI	7.3.6

7.3.23.2. <eventConfig> resource-specific procedure on CRUD operations

7.3.23.2.1. Create

This procedure follows the Generic Resource Create Request Procedure specified in clause 7.2.1.2.1, with the following <eventConfig> resource-specific updates.

Resource-specific operation before Orig-C-1.0 "Compose Request primitive":

1) If event-based statistics collection will be used, the Originator shall generate the representation of the <eventConfig> child resource instance to produce the desired trigger condition for the intended event. For example, one representation of <eventConfig> could have eventType set to "DATA OPERATION" and transactionType set to "RETRIEVE". In another example, a representation could have eventType set to "TIMER-BASED", eventStart set to midnight tomorrow and eventEnd set to midnight of the day after tomorrow. See Table A-1 for value restrictions and default settings pertaining to the attributes of <eventConfig>.

7.3.23.2.2. Retrieve

This procedure follows the Generic Resource Retrieve Request Procedure specified in clause 7.2.2.

7.3.23.2.3. Update

This procedure follows the Generic Resource Update Request Procedure specified in clause 7.2.2.

7.3.23.2.4. Delete

This procedure follows the Generic Resource Delete Request Procedure specified in clause 7.2.2.

7.3.24. Resource Type statsCollect

7.3.24.1. Introduction

The <statsCollect> resource controls the collection of statistics information on an IN-CSE. Information in an associated <eventConfig> resource shall be used by the IN-CSE or IN-AE to define specific event-related triggers. Additional description of the <statsCollect> resource is contained in clauses 9.6.24 and 10.2.15 of oneM2M TS-0001 [6].

Table 7.3.24.1-1: Data Type definition of <statsCollect>

Data Type ID	File Name	Note
Actual Data Type ID	CDT-statsCollect-v1_0_0-<<date of published>>.xsd	

Table 7.3.24.1-2: Resource-specific attributes of <statsCollect>

Attribute Name	Presence in CREATE request	Presence in UPDATE request	Presence in Response	Description
statsCollectID	NP	NP	M	see Table A-1
collectingEntityID	M	NP	M	see Table A-1
collectedEntityID	M	NP	M	see Table A-1
status	M	O	M	see Table A-1
statModel	M	O	M	see Table A-1
subscriberID	M	NP	M	see Table A-1
collectPeriod	O	O	O	see Table A-1
eventID	O	O	O	see Table A-1

Table 7.3.24.1-3: Reference of child resources of <statsCollect>

Child Resource Type Name	Data Type ID	Ref. to in Resource Type Definition
<subscription>	list of xs:anyURI	7.3.6

7.3.24.2. <statsCollect> resource-specific procedure on CRUD operations

7.3.24.2.1. Create

This procedure follows the Generic Resource Create Request Procedure specified in clause 7.2.1.2.1, with the following <statsCollect> resource-specific updates.

Resource-specific operation before Orig-C-1.0:

1) The Originator shall generate and populate a representation of the <statsCollect> resource to produce the desired collection scenario, with the exception of statsCollectID (which is populated by the IN-CSE). If ***statModel*** is set to "EVENT-BASED" then the Originator shall provide a value for ***eventID*** that corresponds to an eventID value stored in a <eventConfig> resource (which defines the event triggers to be used). See Table A-1 for value restrictions and default settings pertaining to the attributes of <statsCollect>.

Resource-specific operation before Rcv-C-5.0:

2) If the ***To*** primitive parameter addresses a receiver CSE that is not an IN-CSE, then the request shall be rejected with a "STATUS_BAD_REQUEST".

Resource-specific operation before Rcv-C-9.0 and after Rcv-C-8.0:

1) The receiver IN-CSE shall generate and store a unique (within the Service Provider domain) value for ***statsCollectID***.

2) If the *status* attribute is set to "ACTIVE", the IN-CSE shall begin monitoring the conditions defined by the <statsCollect> resource and generating Service Statistics Collection Records as the conditions are met.

7.3.24.2.2. Retrieve

This procedure follows the Generic Resource Retrieve Request Procedure specified in clause 7.2.2.

7.3.24.2.3. Update

This procedure follows the Generic Resource Update Request Procedure specified in clause 7.2.2.

Resource-specific operation before Rcv-U-9.0 and after Rcv-U-8.0:

1) If the *status* attribute is set to "ACTIVE", the IN-CSE shall begin monitoring the conditions defined by the <statsCollect> resource and generating Service Statistics Collection Records as the conditions are met.

2) If the ***status*** attribute is set to "INACTIVE", the IN-CSE shall stop monitoring the conditions defined by the <statsCollect> resource.

7.3.24.2.4. Delete

This procedure follows the Generic Resource Delete Request Procedure specified in clause 7.2.2.

7.3.25. Announced Resource Type

7.3.25.1. Introduction

A resource can be announced to one or more remote CSEs to inform the remote CSEs of the existence of the original resource. An announced resource can have a limited set of attributes and a limited set of child resources from the original resource. The announced resource includes a link to the original resource hosted by the original resource-hosting CSE.

All announced resources have the same procedures regardless of the announced resource types.

Table 7.3.25.1-1: Data Type Definition of Announced Resource

Data Type ID	File Name	Note
Actual Data Type ID	CDT-accessControlPolicy-v1_0_0-<<date of published>>.xsd CDT-remoteCSE-v1_0_0-<<date of published>>.xsd CDT-AE-v1_0_0-<<date of published>>.xsd CDT-container-v1_0_0-<<date of published>>.xsd CDT-contentInstance-v1_0_0-<<date of published>>.xsd CDT-schedule-v1_0_0-<<date of published>>.xsd CDT-locationPolicy-v1_0_0-<<date of published>>.xsd CDT-group-v1_0_0-<<date of published>>.xsd CDT- accessControlPolicy-v1_0_0-<<date of published>>.xsd	

Table 7.3.X.1-2: Applicable Common Attributes on Announced Resource

Attribute Name	Request Optionality				Default Value	Resource Specific Note
	C	R	U	D		
resourceType	NP	O	NP	NP	the resource type of the announced resource shall be provided.	
resourceID	NP	O	NP	NP		
parentID	NP	O	NP	NP		
accessControlPolicyIDs	O	O	O	NP		
creationTime	NP	O	NP	NP		
expirationTime	O	O	O	NP		
lastModifiedTime	NP	O	NP	NP		
labels	O	O	O	NP		
link	M	O	O	NP		

Each announced resource type has the resource specific attributes that is the subset of the original resource.

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
Name of attribute specified as MA	M	M	O	NP	the same data type defined at the original resource	this attribute shall be set to the same value with the attribute at the original resource
Name of attribute specified as OA	O	O	O	NP	the same data type defined at the original resource	this attribute shall be set to the same value with the attribute at the original resource

7.3.25.2. Resource Specific Procedure on CRUD Operations

This clause describes announced resource specific procedure for CRUD operations.

The original resource hosting CSE shall create/update/delete the announced resource as specified at the clause 7.2.2.3.9 and clause 7.2.1.2.2.

7.3.25.2.1. Create

.

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.3.25.2.2. Retrieve

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

In case of the **rc** information is set to the "original-resource", the Rcv-R-6.5 shall be changed as follows:

Rcv-R-6.5"Read the original resource whose address is provided by the link attribute of the announced resource"

7.3.25.2.3. Update

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

7.3.25.2.4. No change from the generic procedures in clause 7.2.1.2.2.Delete

Originator:

No change from the generic procedures in clause 7.2.1.2.1.

Receiver:

No change from the generic procedures in clause 7.2.1.2.2.

7.4. Notification definition and procedures

7.4.1. Definition of Notification

7.4.1.1. Introduction

Notification has no resource type representation in the Architecture TS. Rather, Notify request and response primitive formats are defined in clause 7.2.1.1.

Table 7.4.1.1-1: Data Type Definition of notification

Data Type ID	File Name	Note
notificationType	*CDT-<<resource name>>-v1_0_0-<<date of published>>.xsd*	

Table 7.4.1.1-2: Data Types for notification attributes

Attribute Name	Request Optionality	Data Type	Default Value and Constraints
	N		
aggregatedNotification	O	m2m:aggregatedNotification	
singleNotification	O	m2m:singleNotification	

7.4.2. Notification Procedures

Notification is sent as Notify Request primitive as defined in clause 7.2.1.1. The procedures for Notify request and response primitive shall be happen in the following cases.

7.4.2.1. Notification for Subscription

When the notification is forwarded or aggregated by transit CSEs, the Originator or a transit CSE shall check whether there are notification policies to enforce between subscription resource Hosting CSE and the notification target. In that case, the transit CSE as well as the Originator shall process Notify request primitive(s) by using the corresponding policy and send processed Notify request primitive(s) to the next CSE with notification policies related to the enforcement so that the transit CSE is able to enforce the policy defined by the subscriber. The notification policies related to the enforcement at this time is verified by using the subscription reference in the Notify request primitive. In the notification policies, the *latestNotify* attribute is only enforced in the transit CSE as well as the Originator.

Originator: When an event is generated, the Originator shall execute the following steps in order:

Step 1.0 Check the *eventNotificationCriteria* attribute:

If the *eventNotificationCriteria* attribute is set, then the Originator shall check whether the corresponding event matches with the event criteria. In that case, go to the step 2.0. Otherwise, the Originator shall discard the corresponding event

If the eventNotificationCriteria attribute is not configured, then continue with the step 2.0

Step 2.0 The Originator shall check the notification policy as described in the below steps, but the notification policy may be checked in the different order. After checking the notification policy in the step 2.0 (i.e., from the step 2.1to the step 2.6), then continue with the step 3.0

Step 2.1 The Originator shall determine the type of the notification per the *notificationContentType* attribute. The values of for *notificationContentType* are 'modifiedAttribute', 'wholeResource' or optionally 'referenceOnly'

If the value of notificationContentType is set to 'modifiedAttribute', the Notify request primitive shall be included modified attribute(s) only

If the value of notificationContentType is set to 'wholeResource', the Notify request primitive shall be included a whole subscribed-to resource

If the value of notificationContentType is set to 'referenceOnly', the Notify request primitive shall be included an URI of a corresponding <subscription> resource

Step 2.2 Check the *notificationEventCat* attribute:

If the notificationEventCat attribute is set, the Notify request primitive shall have the ec set to the notificationEventCat attribute. Then continue with other step

If the notificationEventCat attribute is not configured, it shall be determined as a default value by the CMDH policy. Then continue with other step

Step 2.3 Check the *rateLimit* attribute:

Step 2.4 Check the *batchNotify* attribute:

Step 2.5 Check the *latestNotify* attribute:

If the latestNotify attribute is set, the Originator shall assign attribute ec of value 'latest' of the notifications generated pertaining to the subscription created. Then continue with other step

Step 2.6 Check the *preSubscriptionNotify* attribute:

Editor's Note: How to operate for the *rateLimit*, *batchNotify*, preSubscriptionNotify attributes is TBD.

Step 3.0 The Originator shall check the notification and reachability schedules, but the notification schedules may be checked in the different order.

- If the notificationSchedule resource is set, then the Originator shall check the time periods by using the scheduleElement attribute
- Also, the Originator shall check the reachability schedule associated with the Receiver by using the <schedule> resource. If reachability schedules are not present in a Node then that Node is considered to be always reachable
- If the notificationSchedule and reachability schedule are allowed, then go to the step 5.0. Otherwise, go to the step 4.0
- In particular, if the *notificationEventCat* attribute is set to 'immediate' and the notificationSchedule resource is not allowed, then go to step 5.0 to send the corresponding Notify request primitive by temporarily ignoring the Originator's notification schedule

Step 4.0 Check the *pendingNotification* attribute:

- If the pendingNotification attribute is set, then the Originator shall cache pending Notify request primitives according to the pendingNotification attribute. The possible values are 'sendLatest' and 'sendAllPending'. If the value of pendingNotification is set to 'sendLatest', the most recent Notify request primitive is cached in the Originator. If it is set to 'sendAllPending', all Notify request primitives are cached in the Originator. If the pendingNotification attribute is not configured, the Originator shall discard the corresponding Notify request primitive. The processed Notify request primitive by the pendingNotification attribute is sent to the Receiver after the reachability recovery (see the step 6.0)

Step 5.0 Check the *expirationCounter* attribute:

- If the expirationCounter attribute is set, then the expirationCounter shall be decreased by one when the Originator successfully sends the Notify request primitive. If the counter meets zero, the corresponding subscription resource is deleted. Then end the 'Compose Notify Request Primitive' procedure
- If the expirationCounter attribute is not configured, then end the 'Compose Notify Request Primitive' procedure

Originator: After reachability recovery, the Originator shall execute the following steps in order:

Step 6.0 If the *pendingNotification* attribute is set, the Originator shall send the processed Notify request primitive by the *pendingNotification* attribute, then continue with the step 7.0

Step 7.0 Check the *expirationCounter* attribute:

- If the expirationCounter attribute is set, then the expirationCounter shall be decreased by one when the Originator successfully sends the Notify request primitive. If the counter meets zero, the corresponding subscription resource is deleted. Then end the 'Compose Notify Request Primitive' procedure
- If the expirationCounter attribute is not configured, then end the 'Compose Notify Request Primitive' procedure

Receiver: When the Hosting CSE receives a Notify request primitive, the Hosting CSE check validity of the primitive parameters. In the case the Receiver is a transit CSE which forwards or aggregates Notify request primitives before sending to the subscriber or the other transit CSEs, upon receiving the Notify request primitive with the ***ec*** set to 'latest', the Receiver shall identify the latest Notify request primitive with the same subscription reference while storing Notify request primitives locally. When the Receiver as a transit CSE needs to send pending Notify request primitives, it shall send the latest Notify request primitive.

7.4.2.2. Subscription Verification during Subscription Creation

Originator:

When the Originator is triggered to perform subscription verification (clause 7.3.7.2.1) during <subscription> creation procedure, it performs the following in order.

1. Add *verificationRequest* parameter set as TRUE into the Notify request primitive.
2. Add *creator* attribute set as the Originator ID of the <subscription> creation into the primitive.
3. Add *to* parameter set as *notificationURI* into the primitive. If the *notificationURI* contains more than one URI, then set each URI to the different primitives.
4. Send the primitive(s).

Receiver:

When the Hosting CSE receives a Notify request primitive including *verificationRequest* parameter set as TRUE, the Hosting CSE shall check if the creator and the Originator have NOTIFY privilege to the *notificationURI*.

If it fails, the Hosting CSE shall return “Subscription verification failed” error with the Notify response primitive. Otherwise, it shall return successful response primitive.

7.4.2.3. Notification for Subscription Deletion

Originator:

When the <subscription> resource is deleted, the Originator shall send a Notify request primitive with subscriptionDeletion attribute set as TRUE and subscriptionRef attribute set as URI of the <subscription> resource.

7.4.2.4. Notification for Asynchronous Non-blocking Request

Editor’s Note: Contributions needed..

Annex A(normative): Resource attributes

Resource Attributes are specified in oneM2M TS-0001 [6]. The type and values shall be supported according to the description given in table A-1.

The attributes are specified by trhe following information in the table:

- ResourceType: indicates the resource where the attribute is used, in case that the attribute is present in all resources the tag "ALL" is used.
- Attribute Name: indicates the name of the Attribute from [6].
- Short Name: indicates the acronym for the correspondent Attribute
- Data type: indicate the used type for the attribute, simple data types are defined in clause 6.3.1 and complex data types are defined in clause 6.3.2.
- Default: specifies the default value of the attribute as set by hosting CSE if no specific value was provided in the operation CREATE or UPDATE request, or if the provided value was unacceptable for the hosting CSE and is not specified to respond with an error then the hosting CSE is allowed to set a value.
- Value restrictions: indicates if the value of the attributes has a specific limitation.

Table A-1: Resource attributes

ResourceType	Attribute Name	Short Name	Data Type	Default	Value restrictions	Notes
ALL except <accessControlPolicy>	accessControlPolicyID	aRI	xs:string	NONE	See TBD	If the attribute is absent, all the entities that correspond to ancestor resources shall have the full set of permissions
firmware	activate					
firmware	activateStatus					
<subscription>	aggregationURI		xs:anyURI			
TBD	announceAttribute	aA	xs:string	NONE		
TBD	announceTo	aT	xs:anyURI	NONE		
<application>, <m2mServiceSubscription>, <nodeInfo>	App-ID		xs:string			
<application>	App-Inst-ID		xs:string			
<cmdhEcDefParamValues>, <cmdhNetworkAccessRules>, <cmdhBuffer>	applicableEventCategory	aEC				
<areaNwkDeviceInfo>	areaNwkId					
<areaNwInfo>	areaNwkType					
<capabilityInstance>	attached					
<cmdhNwAccessRule>	backOffParameters	bOP				
<subscription>	batchNotify					
<battery>	batteryLevel					
<battery>	batteryStatus					
<capabilityInstance>	capabilityActionStatus					
<capabilityInstance>	capabilityName					
<pollingChannel>	channelHandle					
<mgmtCmd>	cmdType					
<statsCollect>	collectedEntityID					
<statsCollect>	collectingEntityID					
<statsCollect>	collectPeriod					
<group>	consistencyStrategy	cS				
<instance>	content	c	Content	NONE		
<request>	content	?	?			
<instance>	contentSize	cS	xs:long			Set by the CSE to the actual size of the received content of

ResourceType	Attribute Name	Short Name	Data Type	Default	Value restrictions	Notes
						the instance
ALL	creationTime	cT	xs:dateTime			Generated by the hosting SCE. The value is set to the actual time of creation of the resource.
TBD	creator	cr	xs:anyURI			Generated by the hosting
<remoteCSE>	cseBase	bRt	xs:anyURI			
<CSEBase>, <remoteCSE>, <nodeInfo>	CSE-ID		xs:string			
<CSEBase> and <remoteCSE>	cseType	csT	TBD			
<container>	currentByteSize	nb	xs:long		Limited by maxByteSize attribute of the same container resource	Set to the actual number of bytes of data stored in the Container resource
<container>	currentNrOfInstances	ni	xs:long		Limited by maxNrOfInstances attribute of the same container resource	Set to the actual number of instances resource in the Container
<group>	currentNrOfMembers	nM	xs:long			
<delivery>	data					
<eventConfig>	dataSize					
firmware	deactivate					
<cmdhEcDefParamValues>	defaultDelAggregation					
<cmdhEcDefParamValues>	defaultOpExecTime					
<cmdhEcDefParamValues>	defaultRequestExpTime	dRqET				
<cmdhEcDefParamValues>	defaultRespPersistence					
<cmdhEcDefParamValues>	defaultResultExpTime	dRsET				
<cmdhDefEcValue>	defEcValue	dev				
<delivery>	deliveryMetaData					
<mgmtObj>, <parameters>, <mgmtCmd>, memory, software, firmware,<areaNwInfo>, <areaNwkDeviceInfo>, battery, <deviceInfo>,	description		xs:string			

ResourceType	Attribute Name	Short Name	Data Type	Default	Value restrictions	Notes
<deviceCapability>, <capabilityInstance>, reboot, <eventLog>						
<deviceInfo>	deviceLabel					
<deviceInfo>	deviceType					
<areaNwkDeviceInfo>	devId					
<areaNwkDeviceInfo>	devType					
<capabilityInstance>	disable					
<capabilityInstance>	enable					
<delivery>	eventCat					
<eventConfig>	eventEnd		xs:dateTime			
<eventConfig>, <statsCollect>	eventID					
<eventConfig>	eventStart		xs:dateTime			
<eventConfig>	eventType					
<execInstance>	execDelay					
<execInstance>	execDisable					
<mgmtCmd>	execEnable					
<execInstance>	execFrequency					
<execInstance>	execMode					
<execInstance>	execNumber					
<mgmtCmd>, <execInstance>	execReqArgs					
<execInstance>	execResult					
<execInstance>	execStatus					
<execInstance>	execTarget					
<subscription>	expirationCounter					
ALL, except <CSEBase>, <instance> and <parameters>	expirationTime	eT	xs:dateTime			The value may be determined by CSE policy. If a value is provided, the CSE shall try to find an acceptable value that is as close as possible to the requested value.
<reboot>	factoryReset					
<subscription>	filterCriteria		FilterCriteria			
<deviceInfo>	fwVersion					
<group>	groupName	gN	xs:string	NONE		
<node>	hostedCSEID		Link			
<deviceInfo>	hwVersion					

ResourceType	Attribute Name	Short Name	Data Type	Default	Value restrictions	Notes
firmware	install					
firmware	installStatus					
<subscription>	interimEventNotify					
ALL	labels	lBs	xs:string	NONE		
ALL	lastModifiedTime	lMT	xs:dateTime	NONE		
<container>	latest	lt	xs:anyURI		Limited to the URI of the instances resources in the container	Set by the CSE to the URI of the latest added instance resource to the container
<subscription>	latestNotify					
<delivery>	lifespan					
<cmdhLimits>	limitsDelAggregation	lDA				
<cmdhLimits>	limitsOpExecTime	lOET				
<cmdhLimits>	limitsRequestExpTime	lRqET				
<cmdhLimits>	limitsRespPersistence	lRP				
<cmdhLimits>	limitsResultExpTime	lRsET				
TBD	link	ln	Link	NONE		
<areaNwInfo>	listOfDevices					
<areaNwkDeviceInfo>	listOfNeighbors					
<locationPolicy>	locationContainerID		xs:anyURI			
<locationPolicy>	locationContainerName		xs:string			
<container>	locationID	LID	xs:anyURI	NONE		
<locationPolicy>	locationServer					
<locationPolicy>	locationSource					
<locationPolicy>	locationTargetId					
<locationPolicy>	locationUpdatePeriod					
<eventLog>	logActionStatus					
<eventLog>	logData					
<eventLog>	logStart					
<eventLog>	logStop					
<eventLog>	logTypeId					
<deviceInfo>	manufacturer					
<cmdhBuffer>	maxBufferSize	mBS				
<container>	maxByteSize	mb	xs:long			Determined by the CSE policy. If a value is provided, the CSE shall try to find an acceptable value that is as close as possible to the requested value.

ResourceType	Attribute Name	Short Name	Data Type	Default	Value restrictions	Notes
<container>	maxInstanceAge	ma	xs:duration			Determined by the CSE policy. If a value is provided, the CSE shall try to find an acceptable value that is as close as possible to the requested value.
<container>	maxNrOfInstances	mi	xs:long			Determined by CSE policy. If a value is provided, the CSE shall try to find an acceptable value that is as close as possible to the requested value.
<group>	maxNrOfMembers	nM	xs:long			Determined by the CSE policy. If a value is provided, the CSE shall try to find an acceptable value that is as close as possible to the requested value.
memory	memAvailable					
<group>	memberAccessRightID	mAR	xs:string			
<group>	memberList		AnyURIList			
<group>	memberType	mT	MemberType	NONE		
<group>	memberTypeValidated	mTV	xs:boolean	NONE		
memory	memTotal					
<request>	metaInformation					
<mgmtObj>, <parameters>, memory, software, firmware,<areaNwInfo>, <areaNwkDeviceInfo>, battery, <deviceInfo>, <deviceCapability>, <capabilityInstance>, reboot, <eventLog>	mgmtDefinition		xs:string			
<cmdhNwAccessRule>	minReqVolume	mRV				
<deviceInfo>	model					
<application>,<cmdhPolicy>, software, firmware	name		xs:string			

ResourceType	Attribute Name	Short Name	Data Type	Default	Value restrictions	Notes
<node>, <nodeInfo>	nodeID		Link			
<CSEBase> and <application>	nodeLink		Link			
<subscription>	notificationDeliveryPriority					
<subscription>	notificationEventCat					
<subscription>	notificationStoragePriority					
<subscription>	notificationStructure					
<subscription>	notificationURI		xs:anyURI			
<mgmtObj>, <parameters>, memory, software, firmware,<areaNwInfo>, <areaNwkDeviceInfo>, battery, <deviceInfo>, <deviceCapability>, <capabilityInstance>, reboot, <eventLog>	objectID		xs:anyURI			
<mgmtObj>, <parameters>, memory, software, firmware,<areaNwInfo>, <areaNwkDeviceInfo>, battery, <deviceInfo>, <deviceCapability>, <capabilityInstance>, reboot, <eventLog>	objectPath					
<application>, <container>, <instance>	ontologyRef		xs:anyURI			
<request>	operation					
<request>	operationResult					
<cmdhDefEcValue>, <cmdhLimits>	order	or				
<request>	originator					
<cmdhNwAccessRule>	otherConditions					
ALL, except <CSEBase>	parentID	pID	xs:anyURI	NONE	See TBD	
<CSEBase>, <remoteCSE> and <application>	pointOfAccess	pOA	AnyURIList			
<subscription>	priorSubscriptionNotify					
<accessControlPolicy>	privileges	ps	Privileges			
<subscription>	rateLimit					
<reboot>	reboot					
<cmdhDefEcValue>, <cmdhLimits>	requestCharacteristics					
<cmdhDefEcValue>, <cmdhLimits>	requestContextNotification					

ResourceType	Attribute Name	Short Name	Data Type	Default	Value restrictions	Notes
<cmdhDefEcValue>, <cmdhLimits>	requestContext	rctxt				
	requestEventCategory	IEC				
<cmdhDefEcValue>, <cmdhLimits>	requestOrigin	roig				
<remoteCSE>	requestReachability		xs:boolean			
<request>	requestStatus					
ALL	resourceType	rT	ResourceType	NONE	See TBD	
<schedule>	scheduleElement		Schedule			
<accessControlPolicy>	selfPrivileges	sP	Privileges			
<areaNwkDeviceInfo>	sleepDuration					
<areaNwkDeviceInfo>	sleepInterval					
delivery	source					
<statsCollect>	statModel					
<statsCollect>	statsCollectID					
<cmdhPolicy>, <areaNwkDeviceInfo>	status		xs:boolean	TRUE = active, FALSE=inactive		
<statsCollect>	status	??	??	Editor's note: this is a different status of active/inactive		
<cmdhBuffer>	storagePriority	sP				
<statsCollect>	subscriberID					
<m2mServiceSubscription>	subsGroup					
<m2mServiceSubscription>	subsSer&RoleList					
<CSEBase>	supportedResourceType	sRT	SupporteResourceType			
<deviceInfo>	swVersion					
<delivery>, <request>	target					
<cmdhNwAccessRule>	targetNetwork	tNet				
<eventConfig>	transactionType					
<CSEBase>, <remoteCSE>	Trigger-Recipient-ID		TBD			
<instance>	typeOfContent	tOC	ContentType	NONE		
software	update					
software	updateStatus					
software, firmware	URL		xs:anyURI			
software, firmware	version					
TBD	versionTag	vT	xs:nonNegativeInteger	0		

2

Annex B(normative): Device Triggering

B.1. Providing Device Triggering service by means of 3GPP networks

B.1.1. Introduction

3GPP Underlying Network has defined a dedicated interface for requesting device triggering. The normative references for applicable interfaces are as follows: 3GPP TS 23.682 [15]. The specification for the interface Tsp is described in 3GPP TS 29.368 [16]. Tsp interface uses Diameter Base Protocol as specified in IETF RFC 3588 [13], in order to use such an interface the CSE shall act as a Diameter client as described in IETF RFC 6733 [14].

Editors Note: IETF RFC 3588 Reference needs to be checked to determine that it is current.

Before the CSE initiates the device triggering, the CSE and MTC-IWF shall execute the procedures once as specified in 3GPP TS29.368 [16].

B.1.2. Device Action Request command

When a CSE needs to issue a device triggering request to the MTC-IWF, the CSE shall send a Device-Action-Request (DAR) command (for detail, see TS 29.368 [16]). The following list provides the parameters mapping between the oneM2M and 3GPP.

Either External-Id or MSISDN: the CSE maps it to the M2M-External-ID, see clause 6.3.2.3.

SCS identifier: the CSE maps it to the CSE-ID, see clause 6.3.2.1.

Application Port Identifier: the CSE maps it to Trigger-Recipient-ID, see clause 6.2.

B.1.3. Device Action Answer command

As a result of device triggering request to MTC-IWF, the CSE receives a Device-Action-Answer (DAA) command (for detail, see TS 29.368 [16]).

B.1.4. Device Notification Request command

As a report of the result for device triggering delivery by 3GPP network, the CSE receives a Device-Notification-Request (DNR) command (for detail, see TS 29.368 [16]).

B.1.5. Device Notification Answer command

As a result of device notification request to MTC-IWF, the CSE sends a Device-Notification-Answer (DNA) command (for detail, see TS 29.368 [16]).

Annex C(informative): XML Examples

C.1. XML Schema for container resource type

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
Copyright Notification

No part of this document may be reproduced, in an electronic retrieval system or
otherwise, except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.
© 2014, oneM2M Partners Type 1 (ARIB, ATIS, CCSA, ETSI, TIA, TTA, TTC). All rights
reserved.

Notice of Disclaimer & Limitation of Liability

The information provided in this document is directed solely to professionals who have
the appropriate degree of experience to understand
and interpret its contents in accordance with generally accepted engineering or other
professional standards and applicable regulations.
No recommendation as to products or vendors is made or should be implied.

NO REPRESENTATION OR WARRANTY IS MADE THAT THE INFORMATION IS TECHNICALLY ACCURATE OR
SUFFICIENT OR CONFORMS TO ANY STATUTE,
GOVERNMENTAL RULE OR REGULATION, AND FURTHER, NO REPRESENTATION OR WARRANTY IS MADE OF
MERCHANTABILITY OR FITNESS FOR ANY
PARTICULAR PURPOSE OR AGAINST INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS.
NO oneM2M PARTNER TYPE 1 SHALL BE LIABLE, BEYOND THE AMOUNT OF ANY SUM RECEIVED IN
PAYMENT BY THAT PARTNER FOR THIS DOCUMENT, WITH RESPECT TO
ANY CLAIM, AND IN NO EVENT SHALL oneM2M BE LIABLE FOR LOST PROFITS OR OTHER INCIDENTAL OR
CONSEQUENTIAL DAMAGES.
oneM2M EXPRESSLY ADVISES ANY AND ALL USE OF OR RELIANCE UPON THIS INFORMATION PROVIDED IN
THIS DOCUMENT IS AT THE RISK OF THE USER.

-->
<xs:schema xmlns="http://www.w3.org/2001/XMLSchema"
   targetNamespace="http://www.onem2m.org/xml/protocols"
   xmlns:m2m="http://www.onem2m.org/xml/protocols"
   elementFormDefault="unqualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:include schemaLocation="common_types-v1_0_0.xsd" />

   <xs:element name="container">
      <xs:complexType>
         <xs:complexContent>
            <!-- Inherit Common Attributes from regularResourceType -->
            <xs:extension base="m2m:regularResourceType">
               <!-- Resource Specific Attributes -->
               <xs:sequence>
                  <xs:element name="maxNrOfInstances" type="xs:nonNegativeInteger"
                     minOccurs="0" />
                  <xs:element name="maxByteSize" type="xs:nonNegativeInteger"
                     minOccurs="0" />
                  <xs:element name="maxInstanceAge" type="xs:nonNegativeInteger"
                     minOccurs="0" />
                  <xs:element name="currentNrOfInstances" type="xs:nonNegativeInteger" />
                  <xs:element name="currentByteSize" type="xs:nonNegativeInteger" />
                  <xs:element name="latest" type="xs:anyURI" minOccurs="0" />
                  <xs:element name="locationID" type="xs:anyURI"
                     minOccurs="0" />
                  <xs:element name="ontologyRef" type="xs:anyURI"
                     minOccurs="0" />

                  <!-- Child Resources -->
                  <xs:element name="childResource" type="m2m:childResourceType"
```

```
                    minOccurs="0" maxOccurs="unbounded" />
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

NOTE: The XML Schema documents need to contain some legal infor-mation at the top. Please use the statements shown in this example for now - thought it is likely that they will need to be replaced prior to external shipment.

C.2. Container resource that conforms to the Schema given above (see clause C.1)

```
<?xml version="1.0" encoding="UTF-8"?>
<m2m:container xmlns:m2m="http://www.onem2m.org/xml/protocols"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation=
   "http://www.onem2m.org/xml/protocols CDT-container-v1_0_0-20140609.xsd "
    name="12xx">
  <parentID>//IN-CSEID.m2m.myoperator.org/96734</parentID>
  <accessControlPolicyIDs>//IN-CSEID.m2m.myoperator.org/93405</accessControlPolicyIDs>
  <creationTime>2013-12-31T12:00:00</creationTime>
  <expirationTime>2013-12-31T12:30:00</expirationTime>
  <lastModifiedTime>2013-12-31T12:00:00</lastModifiedTime>
  <stateTag>0</stateTag>
  <labels>label1 label2</labels>

  <maxNrOfInstances>5</maxNrOfInstances>
  <maxByteSize>104857600</maxByteSize>
  <maxInstanceAge>3600</maxInstanceAge>
  <currentNrOfInstances>2</currentNrOfInstances>
  <currentByteSize>6</currentByteSize>
  <latest>//IN-CSEID.m2m.myoperator.org/96739</latest>
  <locationID>//IN-CSEID.m2m.myoperator.org/1112</locationID>
  <ontologyRef>http://tempuri.org/ontologies/xyz</ontologyRef>

  <childResource name="instance1234" type="instance">//IN-CSEID/1722</childResource>
  <childResource name="instance1235" type="instance">//IN-CSEID/34722</childResource>
  <childResource name="1923" type="subscription">//IN-CSEID/2323</childResource>

</m2m:container>
```

Annex D(Normative): <mgmtObj> Resource Instances

D.1. Introduction

D.2. Resource [firmware]

The detailed description can be found in clause D.2 of Architecture TS-0001 [6].

Table D.2-1: Data Type Definition of [firmware]

Data Type ID	File Name	Note
firmwareType	CDT-firmware-v1_0_0-<<date of published>>.xsd	

Table D.2-2: Data Types for resource specific attributes

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
mgmtDefinition	M	O	NP	NP	See clause 7.3.13	Fixed string “firmware”
objectID	O	O	NP	NP	See clause 7.3.13	
objectPath	O	O	NP	NP	See clause 7.3.13	
description	O	O	O	NP	See clause 7.3.13	
version	M	O	O	NP	xs:string	
name	M	O	O	NP	xs:string	
URL	M	O	O	NP	xs:anyURI	
update	M	NP	O	NP	xs:boolean	
updateStatus	NP	O	O	NP	m2m:actionStatus	

D.3. Resource [software]

The detailed description can be found in clause D.3 of Architecture TS-0001 [6].

Table D.3-1: Data Type Definition of [software]

Data Type ID	File Name	Note
softwareType	CDT-software-v1_0_0-<<date of published>>.xsd	

151 **Table D.3-2: Data Types for resource specific attributes**

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
mgmtDefinition	M	O	NP	NP	See clause 7.3.13	Fixed string “software”
objectID	O	O	NP	NP	See clause 7.3.13	
objectPath	O	O	NP	NP	See clause 7.3.13	
description	O	O	O	NP	See clause 7.3.13	
version	M	O	O	NP	xs:string	
name	M	O	O	NP	list of xs:anyURI	
URL	M	O	O	NP	xs:anyURI	
install	NP	NP	O	NP	xs:boolean	
uninstall	NP	NP	O	NP	xs:boolean	
installStatus	NP	O	NP	NP	m2m:actionStatus	
activate	NP	NP	O	NP	xs:boolean	
deactivate	NP	NP	O	NP	xs:boolean	
activateStatus	NP	O	NP	NP	m2m:actionStatus	

152

153 # D.4. Resource [memory]

154 The detailed description can be found in clause D.4 of Architecture TS-0001 [6].

155 **Table D.4-1: Data Type Definition of [memory]**

Data Type ID	File Name	Note
memoryType	CDT-memory-v1_0_0-<<date of published>>.xsd	

156

157 **Table D.4-2: Data Types for resource specific attributes**

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
mgmtDefinition	M	O	NP	NP	See clause 7.3.13	Fixed string “memory”
objectID	O	O	NP	NP	See clause 7.3.13	
objectPath	O	O	NP	NP	See clause 7.3.13	
description	O	O	O	NP	See clause 7.3.13	
memAvailable	M	O	O	NP	xs:unsignedLong	Unit: Byte.
memTotal	M	O	O	NP	xs:unsignedLong	Unit: Byte.

158

159 # D.5. Resource [areaNwkInfo]

160 The detailed description can be found in clause D.5 of Architecture TS-0001 [6].

161 **Table D.5-1: Data Type Definition of [areaNwkInfo]**

Data Type ID	File Name	Note
areaNwkInfoType	CDT-areaNwkInfo-v1_0_0-<<date of published>>.xsd	

162

Table D.5-2: Data Types for resource specific attributes

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
mgmtDefinition	M	O	NP	NP	xs:string	"areaNwkInfo"
objectID	O	O	NP	NP	xs:string (See clause 7.3.14)	
objectPath	O	O	NP	NP	xs:string (See clause 7.3.14)	
description	O	O	O	NP	xs:string (See clause 7.3.14)	
areaNwkType	M	O	O	NP	xs:string	
listOfDevices	M	O	O	NP	list of xs:anyURI	

D.6. Resource [areaNwkDeviceInfo]

The detailed description can be found in clause D.6 of Architecture TS-0001 [6].

Table D.6-1: Data Type Definition of [areaNwkDeviceInfo]

Data Type ID	File Name	Note
areaNwkDeviceInfoType	CDT-areaNwkDeviceInfo-v1_0_0-<<date of published>>.xsd	

Table D.6-2: Data Types for resource specific attributes

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
mgmtDefinition	M	O	NP	NP	xs:string	"areaNwkDeviceInfo"
objectID	O	O	NP	NP	xs:string (See clause 7.3.14)	
objectPath	O	O	NP	NP	xs:string (See clause 7.3.14)	
description	O	O	O	NP	xs:string (See clause 7.3.14)	
devID	M	O	O	NP	xs:string	
devType	M	O	O	NP	xs:string	
areaNwkId	M	O	O	NP	xs: anyURI	
sleepInterval	O	O	O	NP	xs:NonNegativeInteger	Unit: second
sleepDuration	O	O	O	NP	xs:NonNegativeInteger	Unit: second
Status	O	O	O	NP	xs:string	
listOfNeighbors	M	O	O	NP	list of xs:anyURI	

D.7. Resource [battery]

The detailed description can be found in clause D.7 of Architecture TS-0001 [6].

Table D.7-1: Data Type Definition of [battery]

Data Type ID	File Name	Note
batteryType	CDT-battery-v1_0_0-<<date of published>>.xsd	

Table D.7-2: Data Types for resource specific attributes

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
mgmtDefinition	M	O	NP	NP	See clause 7.3.13	Fixed string “memory”
objectID	O	O	NP	NP	See clause 7.3.13	
objectPath	O	O	NP	NP	See clause 7.3.13	
description	O	O	O	NP	See clause 7.3.13	
batteryLevel	M	O	O	NP	Xs:unsignedInt	Range: 0-100 Unit: Percent
batteryStatus	M	O	O	NP	m2m:batteryStatus	

D.8. Resource [deviceInfo]

The Resource [deviceInfo] is used to provide information regarding the device.

The detailed description can be found in clause D.8 of Architecture TS-0001 [6].

Table D.8-1: Data Type Definition of [deviceInfo]

Data Type ID	File Name	Note
deviceInfoType	CDT-deviceInfo-v1_0_0-<<date of published>>.xsd	

Table D.8-2: Data Types for resource specific attributes

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
mgmtDefinition	M	O	NP	NP	See table 7.3.13.1-2	Fixed string "deviceInfo"
objectID	O	O	NP	NP	See table 7.3.13.1-2	
objectPath	O	O	NP	NP	See table 7.3.13.1-2	
description	O	O	O	NP	See table 7.3.13.1-2	
deviceLabel	M	O	O	NP	xs:string	
manufacturer	M	O	O	NP	xs:string	
model	M	O	O	NP	xs:string	
deviceType	M	O	O	NP	xs:string	
fwVersion	M	O	O	NP	xs:string	
swVersion	M	O	O	NP	xs:string	
hwVersion	M	O	O	NP	xs:string	

D.9. Resource [deviceCapability]

The Resource [deviceCapability] is used to provide information regarding the device.

The detailed description can be found in clause D.9 of Architecture TS-0001 [6].

Table D.9-1: Data Type Definition of [deviceCapability]

Data Type ID	File Name	Note
deviceCapabilityType	CDT-deviceCapability-v1_0_0-<<date of published>>.xsd	

Table D.9-2: Data Types for resource specific attributes

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		

mgmtDefinition	M	O	NP	NP	See table 7.3.13.1-2	Fixed string "deviceCapability"
objectID	O	O	NP	NP	See table 7.3.13.1-2	
objectPath	O	O	NP	NP	See table 7.3.13.1-2	
description	O	O	O	NP	See table 7.3.13.1-2	
capabilityName	M	O	O	NP	xs:string	
attached	M	O	O	NP	xs:boolean	2. true: currently attached to the device 3. false: currently dettached to the device
capabilityActionStatus	M	O	O	NP	m2m: actionStatus	The action (i.e., enable, disable) and the related status. See the Table 6.3.2.3 1
currentState	M	O	O	NP	xs:boolean	• true: the device capability is enabled • false: the device capability is disabled
enable	O	NP	O	NP	xs:boolean	this attribute shall not have any values
disable	O	NP	O	NP	xs:boolean	this attribute shall not have any values

D.10. Resource [reboot]

The Resource [reboot] is used to provide information regarding the device.

The detailed description can be found in clause D.10 of Architecture TS-0001 [6].

Table D.10-1: Data Type Definition of [reboot]

Data Type ID	File Name	Note
rebootType	CDT-reboot-v1_0_0-<<date of published>>.xsd	

Table D.10-2: Data Types for resource specific attributes

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
mgmtDefinition	M	O	NP	NP	See table 7.3.13.1-2	Fixed string "reboot"
objectID	O	O	NP	NP	See table 7.3.13.1-2	
objectPath	O	O	NP	NP	See table 7.3.13.1-2	
description	O	O	O	NP	See table 7.3.13.1-2	
reboot	M	NP	O	NP	xs:boolean	this attribute shall not have any values
factoryReset	M	NP	O	NP	xs:boolean	this attribute shall not have any values

D.11. Resource [eventLog]

The Resource [eventLog] is used to provide information regarding the device.

The detailed description can be found in clause D.11 of Architecture TS-0001 [6].

Table D.11-1: Data Type Definition of [eventLog]

Data Type ID	File Name	Note
eventLogType	CDT-eventLog-v1_0_0-<<date of published>>.xsd	

Table D.11-2: Data Types for resource specific attributes

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
mgmtDefinition	M	O	NP	NP	See table 7.3.13.1-2	Fixed string "eventLog"
objectID	O	O	NP	NP	See table 7.3.13.1-2	
objectPath	O	O	NP	NP	See table 7.3.13.1-2	
description	O	O	O	NP	See table 7.3.13.1-2	
logTypeId	M	O	O	NP	m2m:logTypeId	See Table 6.3.2.2.x-1
logData	M	O	O	NP	xs:string	the content and format of this attribute is out of this specification.
logStatus	M	O	O	NP	m2m:logStatus	See Table 6.3.2.2.x-1
logStart	M	NP	O	NP	xs:boolean	this attribute shall not have any values
logStop	M	NP	O	NP	xs:boolean	this attribute shall not have any values

D.12. Resource [cmdhPolicy]

The Resource [cmdhPolicy] represents a set of rules defining which 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.

The detailed description can be found in clause D.12 of Architecture TS-0001 [6].

Table D.12-1: Data Type Definition of [cmdhPolicy]

Data Type ID	File Name	Note
cmdhPolicyType	CDT-cmdhPolicy-v1_0_0-<<date of published>>.xsd	

Note that the optional <subscription> child resources are not used for CMDH policies.

Table D.12-2: Data Types for resource specific attributes

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
mgmtDefinition	M	O	NP	NP	m2m:mgmtDefinition	“cmdhPolicy”
objectID	O	O	NP	NP	xs:string	See Table 7.3.14.1.2
objectPath	O	O	NP	NP	xs:string	See Table 7.3.14.1.2
description	O	O	O	NP	xs:string	See Table 7.3.14.1.2
name	M	O	O	NP	xs:string	None
mgmtLink	M	O	O	NP	m2m:mgmtLink	1 link to [cmdhDefaults] resource instance, 1 or more link(s) to [cmdhLimits] resource instance(s), 1 or more link(s) to [cmdhNetworkAccessRules] resource instance(s), 1 or more link(s) to [cmdhBuffer] resource instance(s)

The Resource Specific Procedure on CRUD Operations as specified in clause 7.3.14 for the generic <mgmtObj> resource type apply.

D.12.1. Resource [activeCmdhPolicy]

The resource [activeCmdhPolicy] provides a link to the currently active set of CMDH policies.

The detailed description can be found in clause D.12.1 of Architecture TS-0001 [6].

Table D.12.1-1: Data Type Definition of [activeCmdhPolicy]

Data Type ID	File Name	Note
activeCmdPolicyType	CDT-activeCmdhPolicy-v1_0_0-<<date of published>>.xsd	

Table D.12.1-2: Data Types for resource specific attributes

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
mgmtDefinition	M	O	NP	NP	m2m:mgmtDefinition	"activeCmdhPolicy"
objectID	O	O	NP	NP	xs:string	See Table 7.3.14.1.2
objectPath	O	O	NP	NP	xs:string	See Table 7.3.14.1.2
description	O	O	O	NP	xs:string	See Table 7.3.14.1.2
mgmtLink	M	O	O	NP	m2m:mgmtLink	1 link to the instance of [cmdhPolicy] resource that is active

D.12.2. Resource [cmdhDefaults]

The resource [cmdhDefaults] defines default CMDH policy values. The detailed description can be found in clause D.12.2 of Architecture TS-0001 [6].

Table D.12.2-1: Data Type Definition of [cmdhDefaults]

Data Type ID	File Name	Note
cmdhDefaultsType	CDT-cmdhDefaults-v1_0_0-<<date of published>>.xsd	

Table D.12.2-2: Data Types for resource specific attributes

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
mgmtDefinition	M	O	NP	NP	m2m:mgmtDefinition	"cmdhDefaults"
objectID	O	O	NP	NP	xs:string	See Table 7.3.14.1-2
objectPath	O	O	NP	NP	xs:string	See Table 7.3.14.1-2
description	O	O	O	NP	xs:string	See Table 7.3.14.1-2
mgmtLink	M	O	O	NP	m2m:mgmtLink	1 or more link(s) to [cmdhDefEcValue] resource instance(s)

D.12.3. Resource [cmdhDefEcValue]

The resource [cmdhDefEcValue] represents a value for the ec (event category) parameter of an incoming request. The detailed description can be found in clause D.12.3 of Architecture TS-0001 [6].

Table D.12.3-1: Data Type Definition of [cmdhDefEcValue]

Data Type ID	File Name	Note
cmdhDefEcValueType	CDT-cmdhDefEcValue-v1_0_0-<<date of published>>.xsd	

233 **Table D.12.3-2: Data Types for resource specific attributes**

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
mgmtDefinition	M	O	NP	NP	m2m:mgmtDefinition	“cmdhDefEcValue”
objectID	O	O	NP	NP	xs:string	See Table 7.3.14.1-2
objectPath	O	O	NP	NP	xs:string	See Table 7.3.14.1-2
description	O	O	O	NP	xs:string	See Table 7.3.14.1-2
order	M	O	O	NP	xs:positiveInteger	None
defEcValue	M	O	O	NP	m2m:eventCat	None
requestOrigin	M	O	O	NP	m2m:listOfM2MID	None
requestContext	O	O	O	NP	xs:string	None
requestContextNotification	O	O	O	NP	xs:boolean	None
requestCharacteristics	O	O	O	NP	xs:string	None

234

235 D.12.4. Resource [cmdhEcDefParamValues]

236 The resource [cmdhEcDefParamValues] represents a specific set of default values for the CMDH related parameters
237 rqet (request expiration timestamp), rset (result expiration timestamp), oet (operational execution time), rp (response
238 persistence) and da (delivery aggregation) that are applicable for a given ec (event category) if these parameters are not
239 specified in the request. The detailed description can be found in clause D.12.4 of Architecture TS-0001 [6].

240 **Table D.12.4-1: Data Type Definition of [cmdhEcDefParamValues]**

Data Type ID	File Name	Note
cmdhEcDefParamValuesType	CDT-cmdhEcDefParamValues-v1_0_0-<<date of published>>.xsd	

241

242 **Table D.12.4-2: Data Types for resource specific attributes**

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
mgmtDefinition	M	O	NP	NP	m2m:mgmtDefinition	“cmdhEcDefParamValues”
objectID	O	O	NP	NP	xs:string	See Table 7.3.14.1-2
objectPath	O	O	NP	NP	xs:string	See Table 7.3.14.1-2
description	O	O	O	NP	xs:string	See Table 7.3.14.1-2
applicableEventCategory	M	O	O	NP	m2m:listOfEventCat	None
defaultRequestExpTime	M	O	O	NP	xs:long	-1 means infinity, unit: ms
defaultResultExpTime	M	O	O	NP	xs:long	-1 means infinity, unit: ms
defaultOpExecTime	M	O	O	NP	xs:long	-1 means infinity, unit: ms
defaultRespPersistence	M	O	O	NP	xs:long	-1 means infinity, unit: ms
defaultDelAggregation	M	O	O	NP	xs:boolean	None

243

244 D.12.5. [cmdhLimits] Resource

245 The Resource [cmdhLimits] represents limits for CMDH related parameter values. The detailed description can be
246 found in clause D.12.5 of Architecture TS-0001 [6].

247 **Table D.12.5-1: Data Type Definition of [cmdhLimits]**

Data Type ID	File Name	Note
cmdhLimitsType	CDT-cmdhLimits-v1_0_0-<<date of published>>.xsd	

248

Table D.12.5-2: Data Types for resource specific attributes

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
mgmtDefinition	M	O	NP	NP	m2m:mgmtDefinition	"cmdhLimits"
objectID	O	O	NP	NP	xs:string	See Table 7.3.14.1-2
objectPath	O	O	NP	NP	xs:string	See Table 7.3.14.1-2
description	O	O	O	NP	xs:string	See Table 7.3.14.1-2
order	M	O	O	NP	xs:positiveInteger	None
requestOrigin	M	O	O	NP	m2m:listOfM2MID	None
requestContext	O	O	O	NP	xs:string	None
requestContextNotification	O	O	O	NP	xs:boolean	None
requestCharacteristics	O	O	O	NP	xs:string	None
limitsEventCategory	M	O	O	NP	m2m:listOfEventCat	None
limitsRequestExpTime	M	O	O	NP	m2m:listOfMinMax	-1 means infinity, unit: ms
limitsResultExpTime	M	O	O	NP	m2m:listOfMinMax	-1 means infinity, unit: ms
limitsOpExecTime	M	O	O	NP	m2m:listOfMinMax	-1 means infinity, unit: ms
limitsRespPersistence	M	O	O	NP	m2m:listOfMinMax	-1 means infinity, unit: ms
limitsDelAggregation	M	O	O	NP	m2m:listOfBoolean	None

D.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. The detailed description can be found in clause D.12.6 of Architecture TS-0001 [6].

Table D.12.6-1: Type Definition of [cmdhNetworkAccessRules]

Data Type ID	File Name	Note
cmdhNetworkAccessRulesType	CDT-cmdhNetworkAccessRules-v1_0_0-<<date of published>>.xsd	

Table D.12.6-2: Data Types for resource specific attributes

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
mgmtDefinition	M	O	NP	NP	m2m:mgmtDefinition	"cmdhNetworkAccessRules"
objectID	O	O	NP	NP	xs:string	See Table 7.3.14.1.2
objectPath	O	O	NP	NP	xs:string	See Table 7.3.14.1.2
description	O	O	O	NP	xs:string	See Table 7.3.14.1.2
applicableEventCategories	M	O	O	NP	m2m:listOfEventCat	None
mgmtLink	O	O	O	NP	m2m:mgmtLink	Zero or more links to [cmdhNwAccessRule] resource instance(s)

D.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. The detailed description can be found in clause D.12.7 of Architecture TS-0001 [6].

Table D.12.7-1: Data Type Definition of [cmdhNwAccessRule]

Data Type ID	File Name	Note
cmdhNwAccessRuleType	CDT-cmdhNwAccessRule-v1_0_0-<<date of published>>.xsd	

Table D.12.7-2: Data Types for resource specific attributes

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
mgmtDefinition	M	O	NP	NP	m2m:mgmtDefinition	"cmdhNwAccessRule"
objectID	O	O	NP	NP	xs:string	See Table 7.3.14.1-2
objectPath	O	O	NP	NP	xs:string	See Table 7.3.14.1-2
description	O	O	O	NP	xs:string	See Table 7.3.14.1-2
targetNetwork	M	O	O	NP	m2m:listOfM2MID	None
minReqVolume	M	O	O	NP	xs:nonNegativeInteger	Unit: byte
backOffParameters	M	O	O	NP	m2m: backOffParameters	Ordered sequence of 3 values: backoffTime, backoffTimeIncrement, maximumBackoffTime, Unit: ms
otherConditions	O	O	O	NP	xs:string	None
mgmtLink	M	O	O	NP	m2m:mgmtLink	Link to an instance "allowedSchedule" of a <schedule> resource

D.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. The detailed description can be found in clause D.12.8 of Architecture TS-0001 [6].

Table D.12.8-1: Data Type Definition of [cmdhBuffer]

Data Type ID	File Name	Note
cmdhBufferType	CDT-cmdhBuffer-v1_0_0-<<date of published>>.xsd	

Table D.12.8-2: Data Types for resource specific attributes

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
mgmtDefinition	M	O	NP	NP	m2m:mgmtDefinition	"cmdhBuffer"
objectID	O	O	NP	NP	xs:string	See Table 7.3.14.1-2
objectPath	O	O	NP	NP	xs:string	See Table 7.3.14.1-2
description	O	O	O	NP	xs:string	See Table 7.3.14.1-2
applicableEventCategory	M	O	O	NP	m2m:listOfEventCat	None
maxBufferSize	M	O	O	NP	xs:nonNegativeInteger	Unit: byte
storagePriority	M	O	O	NP	xs:positiveInteger	The range of storage priority is from 1 to 10.

Annex E (informative) Procedures for accessing resources

E.1. Accessing Resources in CSEs – Blocking Requests

The result of a Request is send back to the originator together with the Response of the Request. The Originator of the Request may hold the connection to the Registrar CSE until the Response comes back. This communication mode probably result in long blocking times.

The interaction employing blocking mode needs to execute the following steps in order:

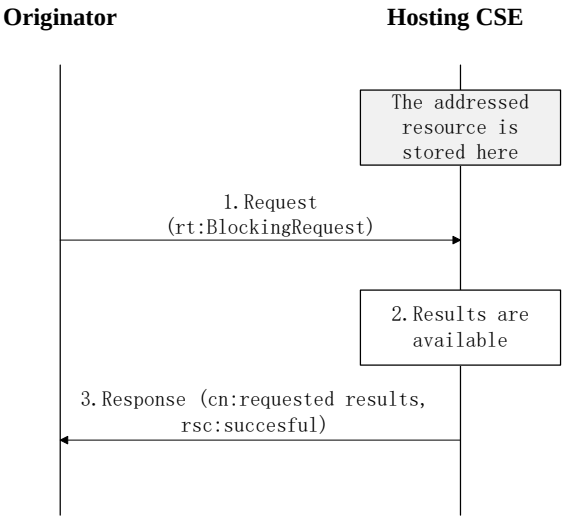


Figure E.1-1: Blocking accesse to resource

1. The Originator sends a request for accessing resources, the *response type* parameter of request is set to 'blockingRequest' , or if this parameter is not provided in the request, it needs to be a blocking request by default.
2. The Hosting CSE receives the request, and it completes the requested processing of resources.
3. Hosting CSE responds to Originator, the response contains the requested results in *resource content*, and the *response status code* parameter of response needs to be set to successful, the value is TBD.

E.2. Accessing Resources in CSEs - non-Blocking Requests

For some reasons, the originator would not wait a long time for a response, it could ask for an Acknowledgement of the request, which provides a reference to the result of the requested operation, then the originator can retrieve the result at a later time. The non-Blocking mode can be used to handle this situation.

E.2.1. Synchronous Case

The Originator demands a non-Blocking Communication, with the *response type* parameter of the Request setting to 'nonBlockingRequestSynch', eg. the Hosting CSE responds after acceptance with an Acknowledgement confirming, then it will further process the Request. The Hosting CSE of the Request s needs to locally create a <request> resource pertaining to the Request received and repond with an acknowledge Response with the reference of the created

<request> resource as the cn of the Response. Then the Recevier needs to continue forward the Request to the next CSE if the Hosting CSE is not the Hosting CSE of the addressed resource. Or the Hosting CSE needs to start handling the Request if the Hosting CSE is the Hosting CSE of the addressed resource.

The Originator if the Request may retrieve the <request> resource afterwards to inspect the final result of the Request if it is available.

The interaction employing non-blocking mode needs to execute the following steps in order:

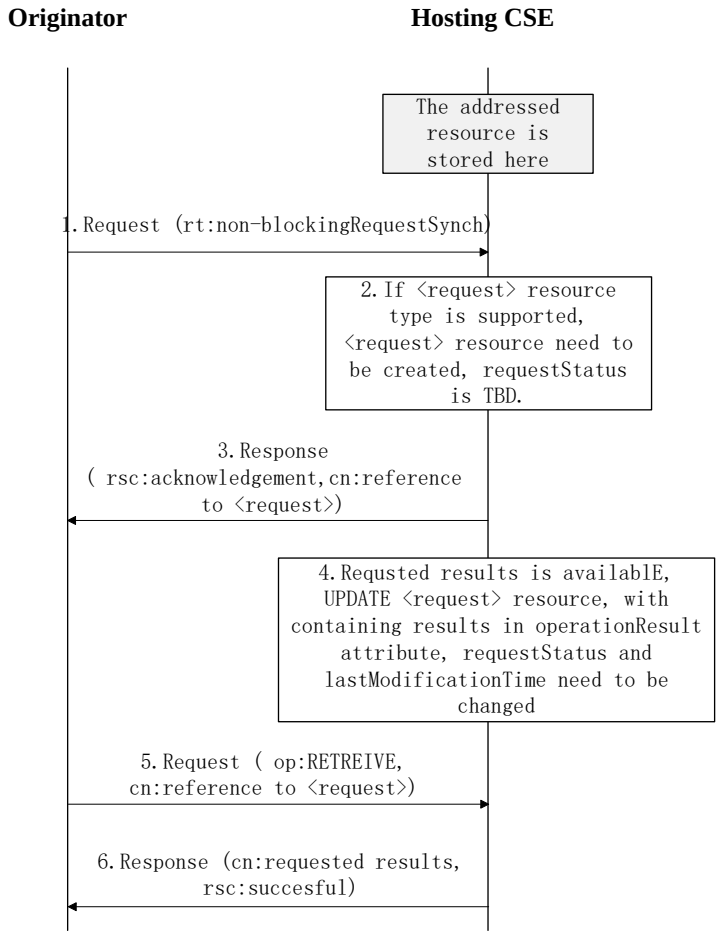


Figure E.2.1-1: non-Blocking accesse to resource in synchronous mode (no hop)

1. The originator sends a request for accessing resources, the *response type* parameter of request is set to 'nonblockingRequestSynch'.

2. In case of the Hosting CSE supports the <request> resource type, it will create an instance of <request> resource, the *response status code* parameter of response needs to be set to aknowledegement, the value is TBD, and a reference to <request> resource is provided in the *content*. Please refer to Table 7.3.11.1-2 for other attributes.

3. Hosting CSE sends a response to the Originator, the *response status code* parameter of response needs to be set to aknowledegement, the value is TBD , and a reference to <request> resource is provided in the *content*.

4. After the requested operation has finished, Hosting CSE will UPDATE the <request> resource, the requested results needs to be contained in the *operationResult* attribute, the values of *requestStatus* and *lastModifiedTime* needs to be changed.

5. Originator requests to RETREIVE the original requested results by addressing the <request> resource.

6. Hosting CSE responds to Originator, the response contains the requested results in *resource content*, and the *response status code* parameter of response needs to be set to successful, the value is TBD.

A variation of synchronous case is depicted in the following clauses. In this variation it is assumed that the addressed resource is not stored in the Registrar CSE, then the Registrar CSE needs to be a Transit CSE to forward the request to the hosting CSE.

The interaction needs to execute the following steps in order:

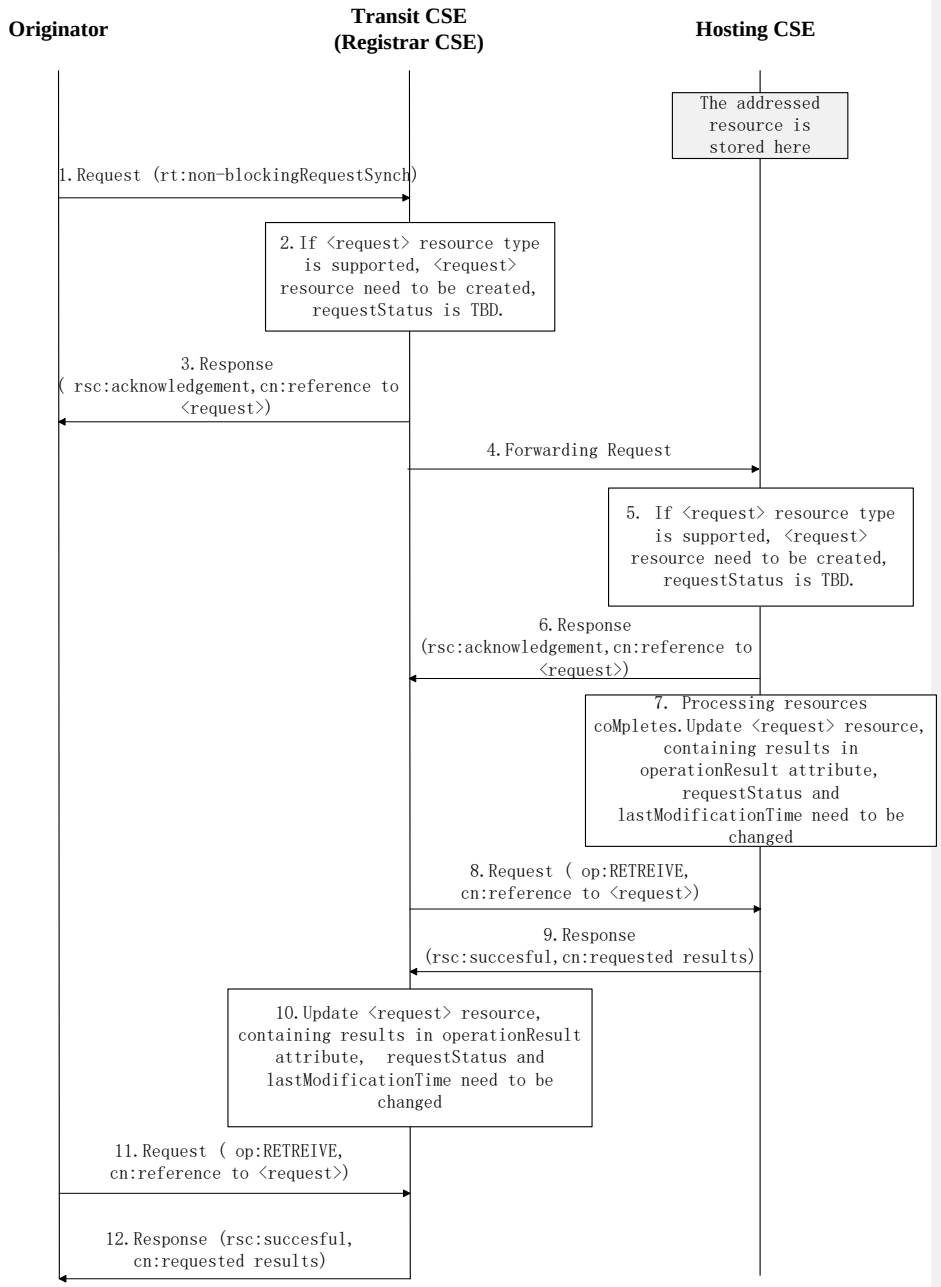


Figure E.2.1-2: non-Blocking accesse to resource in synchronous mode (one hop)

1. The originator sends a request for accessing resources to Registrar CSE (Transit CSE, not Hosting CSE), the *response type* parameter of request is set to 'nonblockingRequestSynch'.

2. In case of the Transit CSE supports the <request> resource type, it will create an instance of <request> resource. The *requestStatus* needs to be set, the value is TBD. Please refer to Table 7.3.11.1-2 for other attributes.

3. Transit CSE sends a response to the Originator, the *response status code* parameter of response needs to be set to aknowledegement, the value is TBD, and a reference to <request> resource is provided in the *content*.

4. Transit CSE forwards the original request to Hosting CSE.

5. In case of the Hosting CSE supports the <request> resource type, it will create an instance of <request> resource. The *requestStatus* needs to be set, the value is TBD. Please refer to Table 7.3.11.1-2 for other attributes.

6. Hosting CSE sends a response to the Originator, the *response status code* parameter of response needs to be set to aknowledegement, the value is TBD, and a reference to <request> resource is provided in the *content*.

7. Hosting CSE processes the resource according to the requested operation, when the operation completes, Hosting CSE updates the <request> resource, the results needs to be contained in the *operationResult* attribute, and the values of *requestStatus* and *lastModifiedTime* needs to be changed.

8. Transit CSE requests to RETREIVE the original requested results by addressing the <request> resource.

9. Hosting CSE send a response to the Transit CSE. The requested result needs to be contained in the *content* of request.

10. Transit CSE updates the <request> resource, the results needs to be contained in the *operationResult* attribute, and the values of *requestStatus* and *lastModifiedTime* needs to be changed.

11. Originator requests to RETREIVE the original requested results by addressing the <request> resource.

12. Hosting CSE responds to Originator, the response contains the requested results in *resource content*, and the *response status code* parameter of response needs to be set to successful, the value is TBD.

E.2.2. Asynchronous Case

Annex F (infomative): Guidelines for one M2M resource type XSD

This Annex contains rules to be followed when creating XML Schemas to represent the oneM2M resources. The schemas themselves form part of the oneM2M protocol specification, but the rules used to construct them do not, hence this Annex is informative.

The purpose of these rules is:

- To keep a consistent style between the schemas for different resources
- To keep the XML Schemas simple
- To allow individual resource schemas to be authored and maintained separately, while minimising the risk of conflict when they are all used together

1) Each Resource Schema will contain a single Global Element Declaration whose name is the name of the Resource Type in accordance with [6]. This means that the root element of a Resource (when represented in XML) contains an m2m: (or equivalent) namespace prefix. It shall not contribute anything to the m2m: namespace other than this root element.

2) The root element of each resource shall have an attribute called “name” which gives an identifier for that particular resource instance. A URI to the resource instance can be constructed by taking the URI of its parent and appending /<name> where <name> is the value of the name attribute.

3) Each resource attribute is represented as a child element of the top level element. It shall be declared as an element that is local to the resource that contains it, and so does not have a namespace prefix in any XML representation of the resource.

4) Each child resource shall be represented as a child element of the top level element which named as ‘childResource’ which shall be used to describe a non-hierarchical URI for the associated child resource. This element shall have two attributes(in XSD) : a) type; Data type ID of instances, b) name; the name of a child resource instance.

5) Each attribute shall be declared to use one of the following data types:

   a. A data type listed in clause 6.3.1 or 6.3.2.

   b. A list of one of the data types listed in clause 6.3.1 or 6.3.2. If the list type is not already included in 6.3.2 it may be defined inside the XSD file for the resource, but if so it must be defined as an anonymous type in the attribute declaration itself.

   c. A data type derived by restriction from one of the types listed in clause 6.3.1 or 6.3.2. This may be defined inside the XSD file for the resource, but if so it must be defined as an anonymous type in the attribute declaration itself.

   d. An anonymous complex type defined as part of the attribute declaration (inside the XSD file for the resource). The complex type should only be composed out of the types listed in clause 6.3.1 or 6.3.2.

6) If a type is used by one than more attribute (either in the same resource or in two different resources) it must be included in 6.3.2, and referenced by each attribute that uses it. Options 5b, 5c, 5d should only be used in cases where the type is only used by one attribute.

7) With the exception of CSEBase, all Resource types will extend one of the XML complex types defined in the file CDT-commonTypes-v1_0<date>.xsd.

8) The resource-specific attributes and child resources shall appear as an XSD sequence, with their order being detemined by the order shown in the tables in clause 9.6 of [6].

9) Each XSD file shall include an XML comment that contains a oneM2M copyright and legal statement, and a change history. The change history is to be filled in only after the initial release.

Annex G(Normative): Location Request

Location Request is a means by which a CSE requests the geographical or physical location information of a target Node to the location server located in the Underlying Network over Mcn reference point. This annex describes only the case of location request when the attribute *locationSource of <locationPolicy> resource type* is set to Network Based. Please see the clause 7.3.8.

The specific interface used for this request depends on the capabilities of the Underlying Network and other factors. This annex provides the interfaces for location request used for the communication between the CSE and the location server.

G.1. Location Request by means of OMA-REST-NetAPI-TerminalLocation Interface

G.1.1. Introduction

This OMA REST Network API for Terminal Location specification v1.0 [i.6]is generally used to open up service capabilities, especially location capability, in the underlying network toward applications. This clause introduces the resources structure and procedures to handle the oneM2M-specified location request. In addition, since this OMA Network API uses only HTTP as underlying message protocol, some binding mapping are mentioned in the procedures in the clause G.1.3.

G.1.2. Resource Structure of OMA NetAPI for Terminal Location

When a CSE needs to request the geographical or physical location information of a target CSE or AE hosted in a M2M Node toward a location server located in the Underlying Network over Mcn reference point. The CSE shall request Terminal Location Query following Procedures for Terminal Location (see Annex.G.1.3).

The OMA REST NetAPI for Terminal Location allows CSE to obtain information about geographical location of a terminal (e.g. Node in oneM2M architecture). In order to obtain location information, CSE shall use one of two services of the Terminal Location API:

- request the current Terminal Location in a single query toward a Location Server
- subscribe to notifications of periodic Terminal Location updates.

Addtionally, in order to track the terminal's movement in relation to the geographic area (circle), crossing in and out (more detail usage is defined in the annex E of TS-0003) it is also proposed to use a service of the Terminal Location API:

- subscribe to notification of area updates

Since oneM2M system utilizes the three services mentioned above, this clause introduces the capabilities that is related to the services from OMA REST NetAPI for Terminal Location [i.7].

Note: A CSE and a Node shall act as an application and a terminal respectively as described in [i.8].

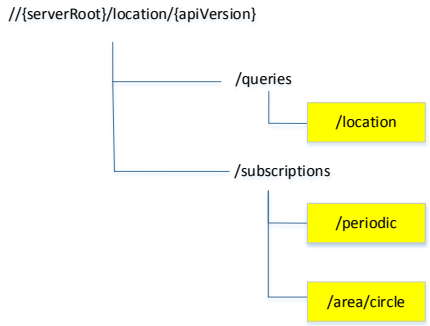


Figure G.1.2-1: Resource Structure defined by NetAPI for Terminal Location

The two capabilities used for oneM2M system location request are 'Terminal location'. 'Periodic location notification subscriptions' and 'area notification subscriptions'. The table below describes the URL structure, data structure and mapping with CRUD operation of each resource.

Table G.1.2-3: Applicable NetAPI for Terminal Location

Capability	URL Base URL:	Resource Type	Operations			
			C	R	U	D
Terminal location	/location	*TerminalLocation*	no	return current location of the terminal	no	no
Periodic location notification subscriptions	/periodic	*PeriodicNotificationSubscription* (used for CREATE)	create new subscription	return all subscriptions	no	No
Area notification subscription	/area/circle	*CircleNotificationSubscription (used for CREATE)*	create a new subscription	return all subscriptions	No	no

Based on the table above, three resource types, *TerminalLocation*, *PeriodicNotificationSubscription* and *CircleNotificationSubscription* shall be used for the location request specified in the oneM2M system. The resource types are described in the tables below. The table also contains the relevant attributes column that is correlated with either <locationPolicy> or <accessControlPolicy> resource type defined [17]. Only attributes that may be utilized by oneM2M system are described. For the detailed information, see the [i.9].

447 **Table G.1.2-4: Resource Type Definition – TerminalLocation**

Attributes	OMA NetAPI Defined Type	Description	Relevant Attribute defined by oneM2M
Address	xsd:anyURI	Address of the terminal to which the location information applies	*locationTargetID* in the <locationPolicy> resource type
locationRetrievalStatus	common:RetrievalStatus	Status of retrieval for this terminal address.	*locationStatus* in the <locationPolicy> resource type
currentLocation	LocationInfo	Location of terminal.	*Content* in the <contentInstance> resource type

448

449 **Table G.1.2-5: Resource Type Definition – PeriodicNotificationSubscription**

Attributes	OMA NetAPI Defined Type	Description	Relevant Attribute defined by oneM2M
address	xsd:anyURI	Addresses of terminals to monitor	*locationTargetID* in the <locationPolicy> resource type
frequency	xsd:int	Maximum frequency (in seconds) of notifications (can also be considered minimum time between notifications) per subscription.	*locationUpdatePeriod* in the <locationPolicy> resource type
duration	xsd:int	Period of time (in seconds) notifications are provided for. If set to "0" (zero), a default duration time, which is specified by the service policy, will be used. If the parameter is omitted, the notifications will continue until the maximum duration time, which is specified by the service policy, unless the notifications are stopped by deletion of subscription for notifications.	*locationUpdatePeriod* in the <locationPolicy> resource type

450

451 **Table 0-6: Resource Type Definition – CircleNotificationSubscription**

Attributes	OMA NetAPI Defined Type	Description	Relevant Attribute defined by oneM2M
Latitude	xsd:float	Latitude of center point.	*accessControlLocationRegion* in the <accessControlPolicy> resource type
longitude	xsd:float	Longitude of center point.	*accessControlLocationRegion* in the <accessControlPolicy> resource type
Radius	xsd:float	Radius of circle around center point in meters.	*accessControlLocationRegion* in the <accessControlPolicy> resource type
checkImmediate	xsd:boolean	Check location immediately after establishing subscription.	

452

G.1.3. Procedures for Terminal Location

G.1.3.1. Request in a Single Query toward a Location Server

This procedure shows how to request and return location for a M2M Node.

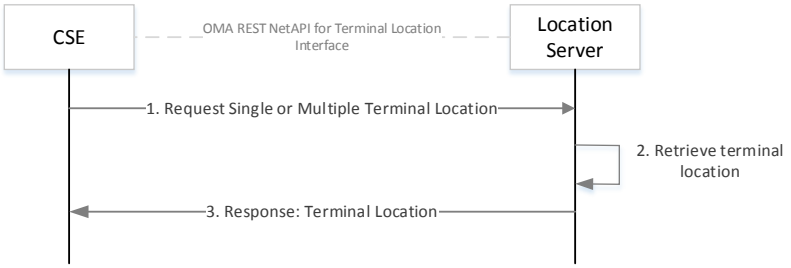


Figure G.1.3.1-1: Single Query Toward Location Server

1. A Hosting CSE requests location for a single terminal (Node) by means of OMA REST NetAPI for terminal location API. This request message shall contain terminal address and Request URL with the address of Location Server using RETRIEVE operation.
   In this step, the *TerminalLocation* resource type described in Table G.1.2-3 shall be used with RETRIEVE operation.

   NOTE: GET operation shall be used for this RETRIEVE operation.

2. The Location Server shall retrieve the location information of the terminal.
3. After the successful retrieve, the Hosting CSE receives the location information.

G.1.4. Subscribe to Notifications for Periodic Location Updates

This procedure shows how to control subscriptions for periodic notifications about terminal location.

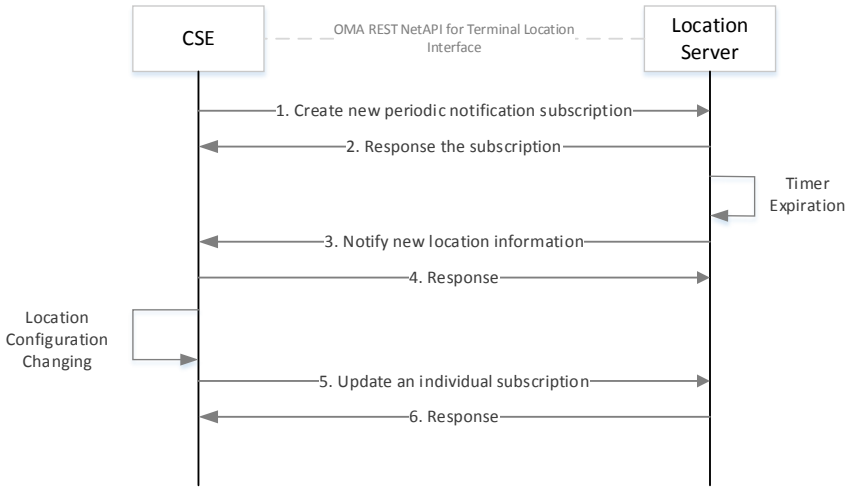


Figure G.1.4-1: Subscribe to Notification for Periodic Location Updates

1. A Hosting CSE shall create a new periodic notification subscription for obtaining location information of a terminal periodically.
   In this step, the PeriodicNotificationSubscription resource type described in Table G.1.2-3 shall be used with CREATE operation.

   NOTE: POST operation shall be used for this CREATE operation.

2. After the successful creation of subscription, the Hosting CSE shall receive the response.
3. When the set up timer is expires, the location server shall notify the application of current location information.
   In this step, the notification message shall be used as NOTIFY operation.

   NOTE: Alternatively, the hosting CSE obtains the notifications using a Notification Channel [i.4]. This is repeated at specific frequency (periodic information) when the CSE is not reachable.

   NOTE: POST operation shall be used for this NOTIFY operation

4. After the successful receiver of notification, the Hosting CSE shall send a response to the location server.
5. Based upon the location configuration change by the Hosting CSE, it updates an individual subscription for periodic location notification.
   In this step, the PeriodicNotificationSubscription resource type described in the Table G.1.2-3 shall be used with UPDATE operation.

   NOTE: PUT operation shall be used for this UPDATE operation.

G.1.5. Subscribe to Notifications for Area Updates

This procedure shows how to subscribe to area update notification.

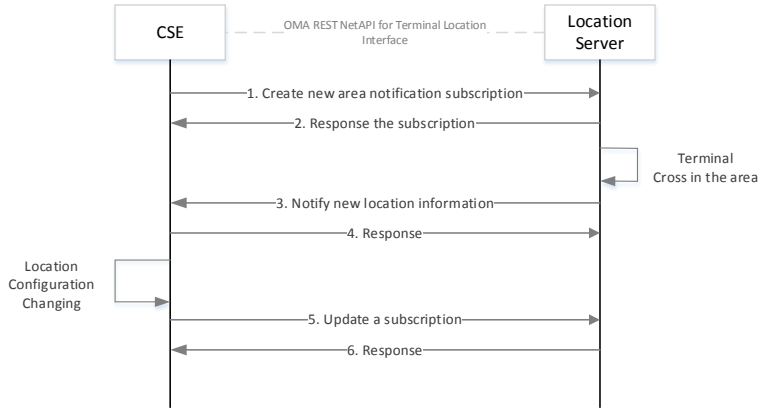


Figure G.1.5-1: Subscribe to Notification for Area Updates

1. A Hosting CSE shall create a new area notification subscription to track the terminal's movement in relation to the geographical area (circle), crossing in and out. In this step, the *CircleNotificationSubscription* resource type described in the table-G.1-3 shall be used with CREATE operation.
   NOTE: POST operation shall be used for this CREATE operation.

2. After the successful creation of subscription, the Hosting CSE shall receive the response.

3. When the target terminal crosses in or out the specified area (circle), the location server shall notify the application of current location information.
   In this step, the notification message shall be used as NOTIFY operation.

   NOTE: Alternatively, the hosting CSE obtains the notifications using a Notification Channel [i.4].

   NOTE: POST operation shall be used for this NOTIFY operation

4. After the successful receiver of notification, the Hosting CSE shall send a response to the location server.

5. Based upon the location configuration change by the Hosting CSE, it updates an individual subscription for area location notification.
   In this step, the *CircleNotificationSubscription* resource type described in the table-G.1-3 shall be used with UPDATE operation.

   NOTE: PUT operation shall be used for this UPDATE operation.

Annex H(Normative): CMDH Message Processing

H.1. Pre-Requisites

The scope of CMDH processing is to decide at which time and via which communication path to forward request or response messages from a receiver CSE to another CSE. A number of message parameters impact the CMDH processing. CMDH-related request message parameters are:

- ***ec***: Event Category
- ***rqet***: Request expiration time
- ***rset***: Result expiration time
- ***oet***: operation execution time
- ***rp***: result persistence
- ***da***: delivery aggregation

CMDH-related response message parameters are:ta

- ***ec***: Event Category
  - ***'ec'*** is needed for response messages as well since response messages can go over multiple hops and CMDH needs to know how to handle them.
- ***rset***: Result expiration time
- ***da***: delivery aggregation
  - When a request message was carried inside a <delivery> resource type, also the corresponding response message shall be carried in a <delivery> resource, i.e. the CSE requested to carry out an operation indicated in a request message that reaged that CSE via a <delivery> resource, shall also send the response within a <delivery> resource.

The details on how those parameters impact the CMDH processing are described in the next clauses.

In the following description it is assumed that the CSE behavior for CMDH processing is governed by CMDH policies that are represented by [cmdhPolicy] resources and their child resources which are effective for the respective CSE. If legacy device management technologies are used to provision these policies, the information represented by the effective [cmdhPolicy] resources and their child resources may not be available as oneM2M defined resources on the field nodes hosting the respective CSE. This CMDH related policy information may only be available in form of managed objects specific to the used device management technology. In that case the mapping from oneM2M specified [cmdhPolicy] resources and their child resources to equivalent objects of the deployed legacy device management technology shall be used to substitute the respective information contained in [cmdhPolicy] resources and their child resources in the description below. Therefore, whenever reference to [cmdhPolicy] resources, child resources thereof or any attributes of [cmdhPolicy] resources and their children are used in the description of CMDH processing below, they shall be read as a placeholder for the equivalent objects provided by legacy device management technologies on field nodes that are provisioned with such legacy device management technologies.

For a CSE that is processing request or response messages in CMDH, exactly one set of policies represented by a [cmdhPolicy] resource shall be active, as defined by the [activeCmdhPolicy] child resource of the <node> resource that represents the node which hosts the respective CSE. In case of field nodes that are managed via legacy device management technologies, the active CMDH policy can be represented by management objects of that device management technology. For the sake of simplicity, the term 'active [cmdhPolicy]' is used in this and the following clauses to refer to the active CMDH policy information even if no oneM2M specified resources are used to represent CMDH policies. Before any provisioning of CMDH policies has occurred, the 'active [cmdhPolicy]' and its corresponding managed objetcs defined for legacy device management technologies shall contain the specified default values as described in the [cdmhPolicy] specific procedures and procedures specific for all its child resources. For that reason, it can be assumed that information for an 'active [cmdhPolicy]' is always present on a CMDH capable CSE.

In addition, the active [cmdhPolicy] can have at least one or more [cmdhLimits] child resources and the active [cmdhPolicy] hosting CSE shall lookup all [cmdhLimits] child resources. If the attribute '*requestContextNotification*' of any of found [cmdhLimits] resources is present and set to true, the CSE shall establish a subscription to the dynamic context information of the CSE defined in '*requestContext*' attribute of the found [cmdhLimits] as well as subscription

to this [cmdhLimits] resource for all AEs corresponding to the AE-ID or an App-ID appearing in the '*requestOrigin*' attribute. The subscription(s) shall be established when the [cmdhPolicy] is provisioned or re-provisioned and any of found [cmdhLimits] child resource has the attribute '*requestContextNotification*' that is set to true. Hence, both this policy establishment and changes of the context information and the [cmdhLimits] resource shall be notified to the respective AEs and the notification shall contain the limits for CMDH related parameter values defined in [cmdhLimits], context information and subscription reference ID. After this, the AEs received the notification shall send only allowed '*ec*' messages if '*ec*' is specified by the AEs.

H.2. CMDH Processing: Processing request or response messages requiring the Receiver CSE to forward information to another CSE

H.2.1. Applicability of CMDH processing

If a request or response message that is targeting an entity or a resource in the '***to***' parameter that is not among any of

- the receiver CSE itself,
- an AE registered with the receiver CSE,
- a resource hosted on the receiver CSE,

and if the message is not a response message with an acknowledgement response code, the receiver CSE of that message needs to forward the message to another CSE via CMDH processing, see also the description in Clause 7.2.1.2. *Description of Generic Procedures* of this TS. For forwarding a message to the target CSE indicated by the '**to**' parameter of the message, the receiver CSE shall determine to which CSE the message needs to be forwarded next. In the following clauses this CSE is referred to as the 'next CSE'. CMDH processing shall be carried out as described in the following clauses.

H.2.2. Partitioning of CMDH processing

The CMDH processing consists of two parts:

A. CMDH message validation: This includes message parameter pre-processing, deciding on acceptance for transporting the message, and buffering of messages. This procedure defines how incoming request or response messages that need to be forwarded to other CSE(s) shall be pre-processed, how a decision on acceptance of the message for forwarding to another CSE shall be derived and how the messages shall be queued up before the actual forwarding can happen. Details of CMDH validation are defined in clause H.2.3. .

B. CMDH message forwarding: This includes selecting buffered messages and communication path for forwarding the message to another CSE.
This procedure defines how to select among the messages buffered for forwarding to other CSEs the ones that need to be transported at a certain time and how to select an appropriate communication path for transporting the message(s). Details of CMDH message forwarding are defined in Annex H.2.4. .

CMDH message validation (Part A) will be carried out for each incoming new message for which CMDH processing is applicable.

If CMDH message validation is successful, the received message shall be queued up for the CMDH message forwarding process (Part B) including the associated 'storagePriority' value as defined in the applicable [cmdhBuffer] resource (see details in the CMDH message validation procedure).

If the queued message was a request message and it was done in non-blocking mode then:

- if the Receiver CSE supports the <request> resource type, it shall create a <request> resource representing the pending non-blocking request

- the Receiver CSE shall send an acknowledgement response message to the entity that sent the request message directly via Mca or Mcc to the receiver CSE indicating the acceptance of the request

- if the receiver CSE supports the <request> resource type it shall provide a reference to the created <request> resource in the ***cn*** parameter of the response.

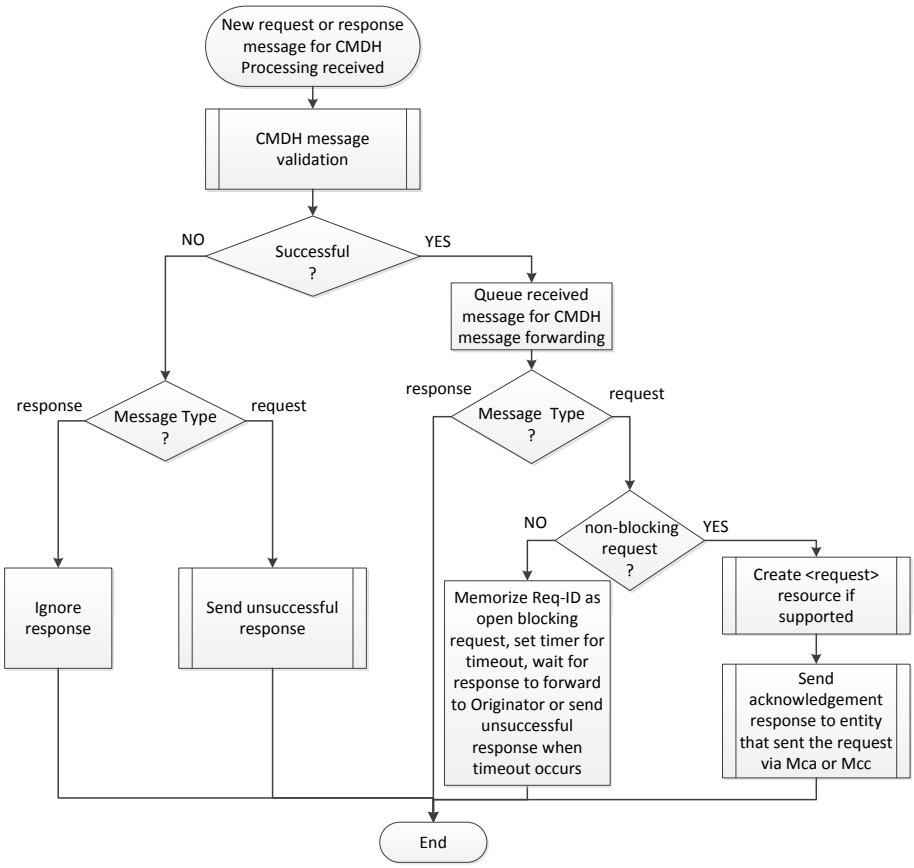
After successful forwarding of such a request message, any incoming response message matching with the Request-ID and the Originator in the <request> resource shall be parsed to update the corresponding attributes of the <request> resource. In case a non-blocking synchronous request was forwarded successfully and a response with acknowledgement was received, it is the responsibility of the CSE that forwarded the message to periodically poll the status of the <request> resource created on the next CSE and update the locally created <request> resource accordingly. When the locally created <request> resource expires the hosting CSE can remove it. Details on <request> resource specific procedures for polling results are defined in clause 7.2.2.1.

If the queued message was a request message and it was done in blocking mode then memorize the open blocking request by storing its Request-ID and Originator and set a timer for a timeout until which a matching response message with the same Request-ID and Originator shall be received by the CSE processing this message. If no matching response is received when the timeout expires, the receiver CSE shall send a response message to the entity that sent the request to the Receiver CSE indicating unsuccessful processing of the request, unless the Receiver CSE and the Originator are the same. If Receiver CSE and Originator are the same, the Originator can decide internally whether to retry forwarding of the message.

If CMDH message validation is not successful, then the received message shall either get ignored – in case the received message is a response message – or a new unsuccessful response message shall be sent back to the entity that sent the message to the Receiver CSE – in case the received message is a request message and the Originator is not the Receiver CSE. If Receiver CSE and Originator are the same, the Originator can decide internally whether to create a new request message.

The CMDH message forwarding process (Part B) will handle all queued up messages that shall be forwarded to another CSE. This process shall always be carried out when messages are pending for forwarding to another CSE.

The flow of CMDH processing is depicted in the flowchart below:



H.2.3. CMDH message validation procedure

In CMDH message validation, pre-processing of CMDH related parameters of a message for which CMDH-processing applies, deriving the decision on acceptance of a message and the buffering of that messages shall be carried out in line with the following steps. A summary of this processing is depicted in the flow chart at the end of this clause.

1. Filling in missing CMDH-related parameters:

    1.1. Determine the value that shall be used for the '***ec***' parameter of the processed message

        1.1.1. If the message contains an '***ec***' parameter: Use the value of the '***ec***' parameter provided in the message.

        1.1.2. If the message does not contain an '***ec***' parameter:

1.1.2.1. Lookup all [cmdhDefEcValue] child resources of the [cmdhDefaults] resource that is a child resource of the provisioned active [cmdhPolicy] resource.

1.1.2.2. If the message is a request message and any of the attributes '*requestContext',* and *'requestCharacteristics'* are present in the found [cmdhDefEcValue] resources, discard all [cmdhDefEcValue] resources from the list of found items for which the context conditions or the request characteristics at time of processing the request message are not met, respectively.

1.1.2.3. Among the remaining found [cmdhDefEcValue] resources do the following selection:

1.1.2.3.1. If present, select the [cmdhDefEcValue] resource containing the AE-ID in the list defined by the '*requestOrigin*' attribute which matches with the '***fr***' parameter in case of a request message or with the '***to***' parameter in case of a response message. If multiple [cmdhDefEcValue] resources match, select the one with the lowest value in the '*order'* attribute. Continue processing with step 1.1.2.4

1.1.2.3.2. If present, select the [cmdhDefEcValue] resource containing the App-ID in the list defined by the '*requestOrigin*' attribute which matches with the '***fr***' parameter in case of a request message or with the '***to***' parameter in case of a response message. If multiple [cmdhDefEcValue] resources match, select the one with the lowest value in the '*order'* attribute. Continue processing with step 1.1.2.4

1.1.2.3.3. If present, select the [cmdhDefEcValue] resource containing the string 'localAE' in the list defined by the '*requestOrigin*' attribute in case of processing a message where the '***fr***' parameter is the AE-ID of an AE registered with the CSE processing this message. If multiple [cmdhDefEcValue] resources match, select the one with the lowest value in the '*order'* attribute. Continue processing with step 1.1.2.4

1.1.2.3.4. If present, select the [cmdhDefEcValue] resource containing the string 'thisCSE' in the list defined by the '*requestOrigin*' attribute in case of processing a message where the '***fr***' parameter is the CSE-ID of the CSE processing this message. If multiple [cmdhDefEcValue] resources match, select the one with the lowest value in the '*order'* attribute. Continue processing with step 1.1.2.4

1.1.2.3.5. Select the [cmdhDefEcValue] resource containing the string 'default' in the list defined by the '*requestOrigin*' attribute in case of processing a message where no other matches were found.

1.1.2.4. If a [cmdhDefEcValue] resource has been selected in steps 1.1.2.3.1 through 1.1.2.3.4: Use the value of the '*defEcValue*' attribute of the selected [cmdhDefEcValue] resource as the value for the ***'ec'*** parameter of the message. Else use the default value of '**bestEffort**' for the '***ec***'

parameter of the message.

1.2. Filling in values that shall be used for the remaining CMDH-related parameters of messages

1.2.1. If the message contains any of the CMDH-related parameters '***rqet***', '***rset***', '***oet***', '***rp***': The provided values of the respective parameters in the message shall be used. No filling in is needed for those parameters. If any of the parameters '***rqet***', '***rset***', '***oet***', '***rp***' present in the message is represented with a duration, the receiving CSE shall translate the values of those parameters into absolute times by adding the duration to the originating timestamp in the '***ot***' parameter of the message. This '***ot***' parameter is an optional message parameter and in case it is not present in a message, it shall be filled in by the first receiving CSE of a message using the time when the message was received.

1.2.2. If the message parameter '***ec***' has a value of '**bestEffort'**, use the following values for any missing CMDH-related parameters: For a request message use '***rqet***' = 'infinite', '***rset***' = 'infinite', '***oet***' = 'now', '***rp***' = 'none', '***da***' = ON. For a response message use '***rset***' = 'infinite', '***da***' = ON. Continue with step 2.

1.2.3. If the message parameter '***ec***' has a value of '**immediate'**, do not fill in any remaining missing CMDH-related parameters and continue with step 2.

1.2.4. For any of the missing CMDH-related parameters fill in values as follows:

1.2.4.1. Lookup all [cmdhEcDefParamValues] child resources of the [cmdhDefaults] resource that is a child resource of the provisioned active [cmdhPolicy] resource.

1.2.4.2. Among the found [cmdhEcDefParamValues] resources do the following selection:

1.2.4.2.1. If present, select the [cmdhEcDefParamValues] resource containing the value of the '***ec***' parameter of the message in the list defined by the '*applicableEventCategory*' attribute. If a match is found, continue processing with step 1.2.4.3

1.2.4.2.2. Select the [cmdhEcDefParamValues] resource that contains the string 'default' in the list defined by the '*applicableEventCategory*'.

1.2.4.3. Use the following attributes of the selected [cmdhEcDefParamValues] resource to fill in any missing CMDH-related message parameters: Fill in the value of the attribute '*defaultRequestExpTime'* for the parameter '***rqet***' if it is missing. Fill in the value of the attribute '*defaultResultExpTime'* for the parameter '***rset***' if it is missing. Fill in the value of the attribute '*defaultOpExecTime'* for the parameter '***oet***' if it is missing. Fill in the value of the attribute '*defaultRespPersistence'* for the parameter '***rp***' if it is missing. Fill in the value of the attribute '*defaultDelAggregation'* for the parameter '***da***' if it is missing.

2. Compare CMDH parameters with allowed CMDH parameter limits:
Check if CMDH-related parameters effective for the message are with allowed limits.

2.1. Lookup all [cmdhLimits] child resources of the provisioned active [cmdhPolicy] resource.

2.2. If the message is a request message and any of the attributes '*requestContext',* and *'requestCharacteristics'* are present in the found [cmdhLimits] resources, discard all [cmdhLimits] resources from the list of found items for which the context conditions or the request characteristics at time of processing the request message are not met, respectively.

2.3. Among the remaining found [cmdhLimits] resources do the following selection:

2.3.1. If present, select the [cmdhLimits] resource(s) containing the AE-ID in the list defined by the '*requestOrigin*' attribute which matches with the '***fr***' parameter in case of a request message or with the '***to***' parameter in case of a response message. If multiple [cmdhLimits] resources match, select the one with the lowest value in the '*order'* attribute. Continue processing with step 2.4

2.3.2. If present, select the [cmdhLimits] resource(s) containing the App-ID in the list defined by the '*requestOrigin*' attribute which matches with the '***fr***' parameter in case of a request message or with the '***to***' parameter in case of a response message. If multiple [cmdhLimits] resources match, select the one with the lowest value in the '*order'* attribute. Continue processing with step 2.4

2.3.3. If present, select the [cmdhLimits] resource(s) containing the string 'localAE' in the list defined by the '*requestOrigin*' attribute in case of processing a message where the '***fr***' parameter is the AE-ID of an AE registered with the CSE processing this message. If multiple [cmdhLimits] resources match, select the one with the lowest value in the '*order'* attribute. Continue processing with step 1.1.2.4

2.3.4. If present, select the [cmdhLimits] resource(s) containing the string 'thisCSE' in the list defined by the '*requestOrigin*' attribute in case of processing a message where the '***fr***' parameter is the CSE-ID of the CSE processing this message. If multiple [cmdhLimits] resources match, select the one with the lowest value in the '*order'* attribute. Continue processing with step 2.4

2.3.5. Select the [cmdhLimits] resource containing the string 'default' in the list defined by the '*requestOrigin*' attribute in case of processing a message where no other matches were found.

2.4. Validate if '***ec***' parameter is within allowed range:
If the '***ec***' parameter of the message is not within the list defined by the 'limitsEventCategory' attribute of the selected [cmdhLimits] resource, mark CMDH message validation for this message as not successful and exit CMDH message validation.

2.5. Validate if '***rqet***' parameter is within allowed range:
If the '***rqet***' parameter is present in the message and if it is not within the range

804 defined by the 'limitsRequestExpTime' attribute of the selected [cmdhLimits]
805 resource, mark CMDH message validation for this message as not successful and
806 exit CMDH message validation.
807
808 2.6. Validate if '***rset***' parameter is within allowed range:
809 If the '***rset***' parameter is present in the message and if it is not within the range
810 defined by the 'limitsResultExpTime' attribute of the selected [cmdhLimits] resource,
811 mark CMDH message validation for this message as not successful and exit
812 CMDH message validation.
813
814 2.7. Validate if '***oet***' parameter is within allowed range:
815 If the '***oet***' parameter is present in the message and if it is not within the range
816 defined by the 'limitsOpExecTime' attribute of the selected [cmdhLimits] resource,
817 mark CMDH message validation for this message as not successful and exit
818 CMDH message validation.
819
820 2.8. Validate if '***rp***' parameter is within allowed range:
821 If the '***rp***' parameter is present in the message and if it is not within the range
822 defined by the 'limitsRespPersistence' attribute of the selected [cmdhLimits]
823 resource, mark CMDH message validation for this message as not successful and
824 exit CMDH message validation.
825
826 2.9. Validate if '***da***' parameter is within allowed range:
827 If the '***da***' parameter is present in the message and if it is not within the list of
828 allowed values defined by the 'limitsDelAggregation' attribute of the selected
829 [cmdhLimits] resource, mark CMDH message validation for this message as not
830 successful and exit CMDH message validation.
831
832 3. Check if message complies with network access rules and buffer limits:
833
834 3.1. Check if '***ec***' is '**immediate**':
835 If the '***ec***' parameter of the message is '**immediate**' bypass any checks on
836 buffering or access network usage rules. Mark the CMDH message validation for
837 this message as successful and end CMDH message validation.
838
839 3.2. Check if delivering the message is possible within the boundaries of access
840 network usage rules in CMDH policies:
841
842 3.2.1. Lookup all [cmdhNetworkAccessRules] child resources of the provisioned
843 active [cmdhPolicy] resource.
844
845 3.2.2. Among the all found [cmdhNetworkAccessRules] resources do the following
846 selection:
847
848 3.2.2.1. If present, select the [cmdhNetworkAccessRules] resource containing
849 the value of the '***ec***' parameter of the message in the list defined by the
850 '*applicableEventCategory*' attribute. If a match is found, continue
851 processing with step 3.2.3
852

3.2.2.2. Select the [cmdhNetworkAccessRules] resource that contains the string 'default' in the list defined by the '*applicableEventCategory*'.

3.2.3. Lookup all [cmdhNwAccessRule] child resources of the selected [cmdhNetworkAccessRules] resource

3.2.4. Among the all found [cmdhNwAccessRule] resources find at least one for which the <schedule> child resource 'allowedSchedule' is allowing usage of the corresponding target network consistent with the '***rqet***' parameter in case of a request message being processed or in line with the '***rset***' parameter in case of a response message being processed. If no matching [cmdhNwAccessRule] resource is found, mark CMDH validation for this message as not successful due to lack of scheduling opportunities and end CMDH message validation. Otherwise continue.

3.3. Check if delivering the message is possible within the boundaries of buffer usage rules in CMDH policies:

3.3.1. Lookup all [cmdhBuffer] child resources of the provisioned active [cmdhPolicy] resource.

3.3.2. Among the all found [cmdhBuffer] resources do the following selection:

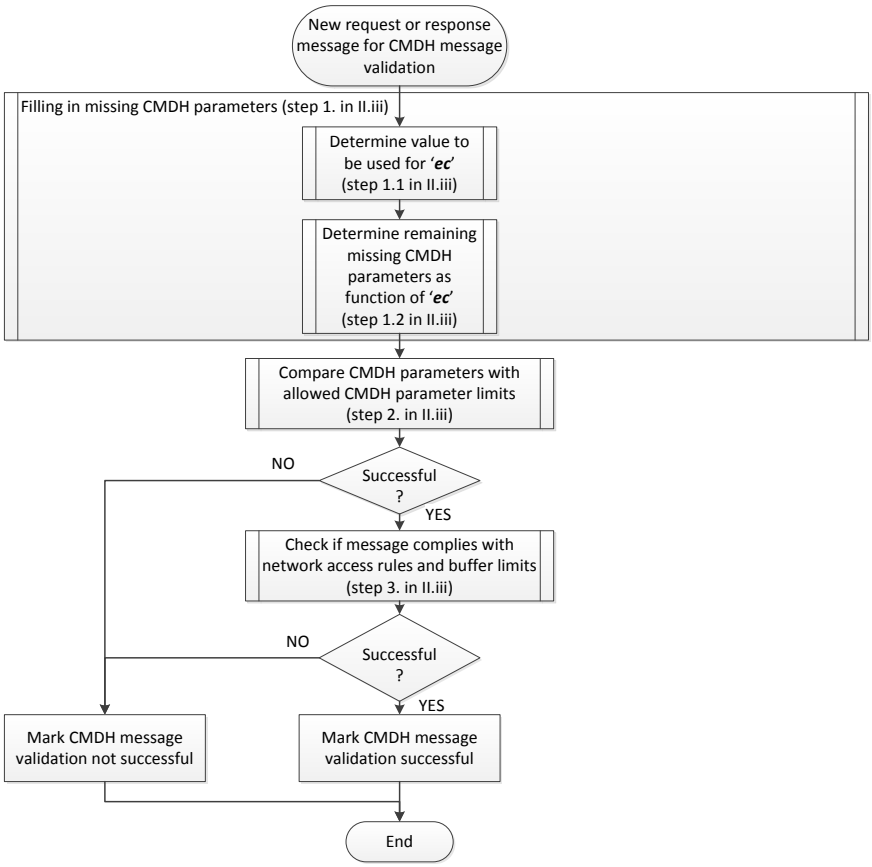
3.3.2.1. If present, select the [cmdhBuffer] resource containing the value of the '***ec***' parameter of the message in the list defined by the '*applicableEventCategory*' attribute. If a match is found, continue processing with step 3.3.3

3.3.2.2. Select the [cmdhBuffer] resource that contains the string 'default' in the list defined by the '*applicableEventCategory*'.

3.3.3. Check if the amount of memory needed to buffer the message being validated in addition to the already buffered messages matching with the same buffer usage policy in the selected [cmdhBuffer] resource would exhaust the limit defined by the 'maxBufferSize' attribute of the selected [cmdhBuffer] resource or if the available memory for CMDH forwarding on the receiver CSE would get exhausted even when purging buffered messages with lower storage priority.

3.3.3.1. If the check is negative, mark the CMDH message validation for the message being validated as successful, assign the storage priority defined in the 'storagePriority' attribute of the selected [cmdhBuffer] resource to the validated message, and end CMDH message validation

3.3.3.2. If the check is positive, mark the CMDH message validation for the message being validated as not successful and end CMDH message validation.



H.2.4. CMDH message forwarding procedure

The high-level sequence of processing steps for the CMDH message forwarding process is depicted in the flow chart below. Note that this flow chart only represents the reference flow for implementing a standard compliant behavior. Other standard compliant implementations may be possible as long as the events defined below will result in the same normative message exchanges via reference points.

Occurrence of the following events shall trigger processing in the CMDH message forwarding:

- One or more new message(s) get(s) queued up for CMDH message forwarding
- Any of the underlying networks becomes usable for message forwarding due to transition(s) in allowed schedule(s) or due to establishing of availability of connectivity (e.g. cable plugged-in, coverage established)
- Any of the underlying networks becomes unusable for message forwarding due to transition(s) in allowed schedule(s) or due to loss of availability of connectivity (e.g. cable unplugged, coverage lost)
- Any message buffered for CMDH forwarding expires

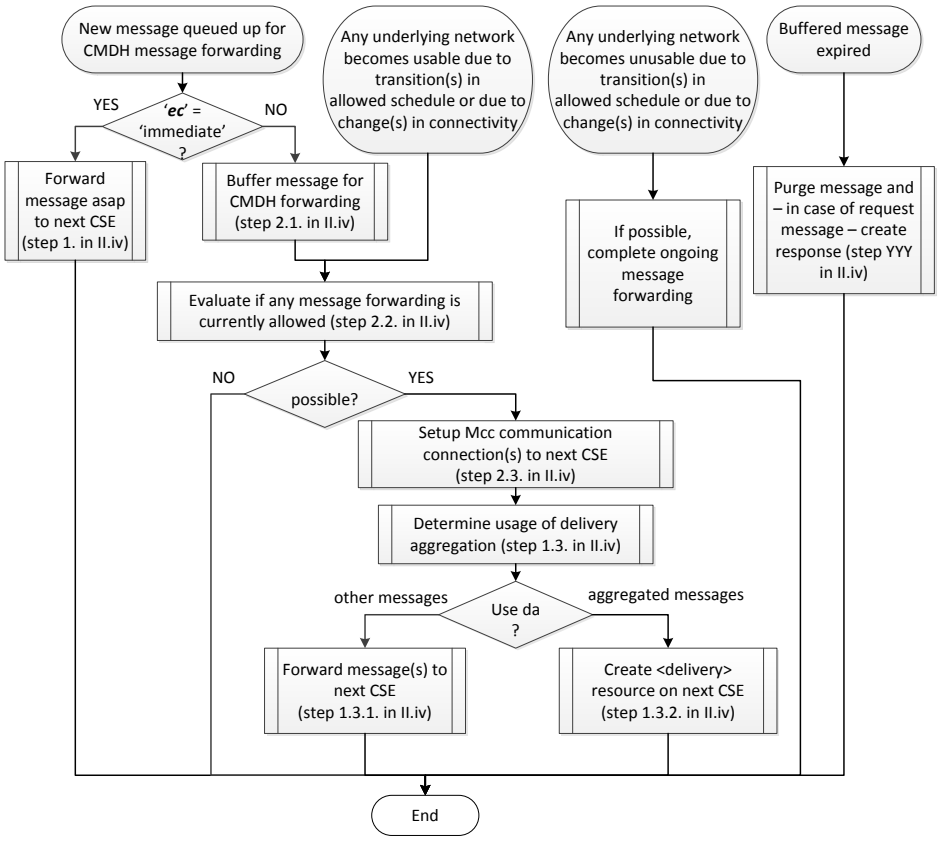


Figure H.2.4-1: (Title is TBD)

When a new message is getting queued up for CMDH message forwarding, carry out the following:

1. If the '***ec***' parameter of the messages has the value '**immediate**':
   Forward message as soon as possible to the next CSE. The processing in this situation is described by the flow chart in Figure 2 below.

   1.1. If a Mcc communication connection to the next CSE for forwarding the message is already established, continue with step 1.3.

   1.2. If no Mcc communication connection to the next CSE for forwarding the message is established pick one underlying network among all underlying networks that can provide communication to the next CSE and establish a Mcc communication connection to the next CSE in line with the rules outlined in clause H.2.5. . If establishment of a Mcc communication connection to the next

CSE was not successful before the message expires, continue with step 1.4.

1.3. Determine whether delivery aggregation or forwarding of the message itself shall be used:

1.3.1. If the message contains a '***da***' parameter set to the value 'ON', the Receiver CSE shall forward this message by creation of a <delivery> resource on the next CSE as outlined in clause 7.3.10. The receiver CSE can combine the forwarded message in the same <delivery> resource with other messages for which the '***da***' parameter set to 'ON' and which need to be forwarded to the same target CSE.

1.3.2. If the message is not forwarded using a <delivery> resource, the receiver CSE shall forward the message as is to the next CSE via the established Mcc communication connection.

1.4. If the message could not be forwarded successfully to the next CSE before it expired (e.g. due to repeated unsuccessful attempts to establish a Mcc communication connection or due to the lack of usable underlying networks), the receiver CSE shall carry out the following:

1.4.1. If the message was a response message, ignore the message. End this cycle of CMDH message forwarding and wait for new triggering events.

1.4.2. If the message was a request message:

1.4.2.1. If the request was a blocking request:
Send a unsuccessful response to the pending blocking request with a matching Request-ID and Originator indicating the reason for failure and close the blocking request. End this cycle of CMDH message forwarding and wait for new triggering events.

1.4.2.2. If the request was a non-blocking request:
Update the associated <request> resource with matching Request-ID and Originator using an unsuccessful response code indicating the reason for failure. If the non-blocking request was made in asynchronous mode, send a notification with the unsuccessful response to the notification target(s) of the request. End this cycle of CMDH message forwarding and wait for new triggering events.

1.5. Else, i.e. if the message was forwarded successfully to the next CSE:

1.5.1. If the message was a response and the Receiver CSE has an open blocking request context with a matching Request-ID and matching Originator, mark the open blocking request as closed, end this cycle of CMDH message forwarding and wait for new triggering events.

1.5.2. If the message was a request message:

1.5.2.1. If the request was a blocking request:
Keep the context of the pending blocking request with matching

Request-ID and matching Originator open and wait for an incoming response message with the same Request-ID and Originator. End this cycle of CMDH message forwarding and wait for new triggering events.

1.5.3. If the request was a non-blocking request:
Wait for a response to the forwarded request (e.g. response with acknowledgement or unsuccessful response). Update the associated <request> resource with the matching Request-ID and Originator using a response code that reflects the status of the forwarded request (e.g. accepted by next CSE, unsuccessful). If the next CSE responded with an unsuccessful response message and the request was in non-blocking asynchronous mode, send a notification request message to the Originator of the forwarded request containing the unsuccessful response of the next CSE. End this cycle of CMDH message forwarding and wait for new triggering events.

2. Else, i.e. when the '***ec***' parameter of the messages does not have the value '**immediate**':

2.1. Buffer the message to be forwarded in the CMDH forwarding buffer:
The processing in this situation is described by the flow chart in Figure 2 below.

2.1.1. If the message is a request message and the '***ec***' parameter of the messages has the value '**latest**':

2.1.1.1. If the request message is a notification triggered by a subscription:

2.1.1.1.1. Find any buffered request message that is a notification triggered by a subscription with the same subscription reference.

2.1.1.2. Else, i.e. if the request message is not a notification triggered by a subscription:

2.1.1.2.1. Find any buffered request message that has the same values in the ('***fr***', '***to***', '***op***') parameters as the message being processed

2.1.1.3. If any request message was found in steps 2.1.1.1.1 or 2.1.1.2.1, purge the found message from the CMDH forwarding buffer.

2.1.2. If there is not enough memory available to buffer the message being processed in the CMDH forwarding buffer:

2.1.2.1. Find any buffered messages with storage priority values lower than the one assigned to the message being processed.

2.1.2.2. If any messages are found:
Purge enough messages among the found messages so that the message being processed can be buffered in the CMDH forwarding buffer. Messages which entered the buffer later shall be purged first. In

case any request messages need to be purged, carry out the following:

2.1.2.2.1. In case of purging a non-blocking request messages:
Update the associated <request> resource with the same Request-ID as the purged request message with a status indicating unsuccessful completion. If the purged message was made in asynchronous mode, send a response to the notification target(s) of the pending non-blocking request

2.1.2.2.2. In case of purging a blocking request message:
Send a unsuccessful response to the open blocking request with the same Request-ID as in the purged request message and close the blocking request.

2.1.2.3. Due to the checking of sufficient memory in CMDH message forwarding buffer during CMDH message validation, there should be enough memory available to accommodate the message to be buffered at this point. If that is still not the case, then do the following:

2.1.2.3.1. In case the message to be buffered is a response message:
Ignore the message to be buffered. End this cycle of CMDH message forwarding and wait for new triggering events.

2.1.2.3.2. In case the message to be buffered is a non-blocking request message:
Update the associated <request> resource with the same Request-ID as the request message to be buffered with a status indicating unsuccessful completion. If the request message to be buffered was made in asynchronous mode, send a response to the notification target(s) of the pending non-blocking request. End this cycle of CMDH message forwarding and wait for new triggering events.

2.1.2.3.3. In case the message to be buffered is a blocking request message:
Respond with an unsuccessful response message to the open blocking request with the same Request-ID as in the request message to be buffered and close the blocking request. End this cycle of CMDH message forwarding and wait for new triggering events.

2.1.3. Store the message to be buffered with its assigned storage priority in the CMDH forwarding buffer. Include it in future evaluations for possible message forwarding.

2.2. Evaluate if any message forwarding is currently allowed:

2.2.1. For all buffered messages that are pending in CMDH message forwarding carry out the following evaluation steps:

2.2.1.1. Among all [cmdhNetworkAccessRules] child resources of the provisioned active [cmdhPolicy] resource do the following selection:

2.2.1.1.1. If present, select the [cmdhNetworkAccessRules] resource containing a value in the list defined by the '*applicableEventCategory*' attribute that is equal to the value of the '***ec***' parameter of the buffered message to be evaluated for forwarding. If a match is found, continue processing with step 2.2.1.2.

2.2.1.1.2. Select the [cmdhNetworkAccessRules] resource that contains the string 'default' in the list defined by the '*applicableEventCategory*'.

2.2.1.2. Lookup all [cmdhNwAccessRule] child resources of the selected [cmdhNetworkAccessRules] resource

2.2.1.3. If the attribute '*otherConditions' is* present in any of the found [cmdhNwAccessRule] resources, discard all [cmdhNwAccessRule] resources from the list of found items for which the conditions expressed by 'otherConditions' at time of evaluation of the message for forwarding are not met, respectively.

2.2.1.4. Among the all remaining found [cmdhNwAccessRule] resources find those for which
- the <schedule> child resource 'allowedSchedule' is currently allowing usage of the corresponding target network, and
- for which the corresponding target network could be used to reach the next CSE for forwarding the message under evaluation.

If any allowed target network was found, memorize the message under evaluation as an allowed message and the allowed target network(s) for the message under evaluation and continue with the next evaluation of buffered messages

2.2.2. When all buffered messages have been evaluated, remove from the memorized list of allowed messages and their allowed target networks those target networks where the amount of data to be forwarded – accumulated over all allowed messages of the same event category – is less than the amount of data indicated in the 'minReqVolume' attribute of the corresponding [cmdhNwAccessRule] resource.

2.2.3. Remove any messages from the list of allowed messages for forwarding if no allowed target network is left for that message after the previous step.

2.3. Process messages allowed for forwarding to the next CSE:
If any messages can be forwarded, i.e. if any evaluation of step 2.2 was positive, apply the following steps:

2.3.1. Reuse already established Mcc communication connections or – if needed – establish new Mcc communication connection(s) so that all the messages that are allowed to be forwarded to their next CSE can be forwarded. Some

messages may be allowed on the same target network. Follow the procedure outlined in clause H.2.5. for setting up a Mcc communication connection to another CSE via a particular target network. If no usable Mcc communication connection could be established for forwarding a particular allowed message before the message expires, execute step 1.4 in this clause above for that message.

1.

2.3.2. For all messages allowed for forwarding and for which Mcc communication connections are established, apply steps 1.3 through 1.5 in this clause above.

2.4. Else, i.e. currently no message forwarding is allowed:
End this cycle of CMDH message forwarding and wait for new triggering events.

When any of the underlying networks becomes usable for message forwarding due to transition(s) in allowed schedule(s) or due to establishing of availability of connectivity (e.g. cable plugged-in, coverage established), carry out the processing above in this clause starting with step 2.2.

When any of the underlying networks becomes unusable for message forwarding due to transition(s) in allowed schedule(s) or due to loss of availability of connectivity (e.g. cable unplugged, coverage lost), complete – if at all possible – any ongoing message forwarding procedures. End this cycle of CMDH message forwarding and wait for new triggering events.

When any message buffered for CMDH forwarding expires, carry out step 1.4 in this clause above. End this cycle of CMDH message forwarding and wait for new triggering events.

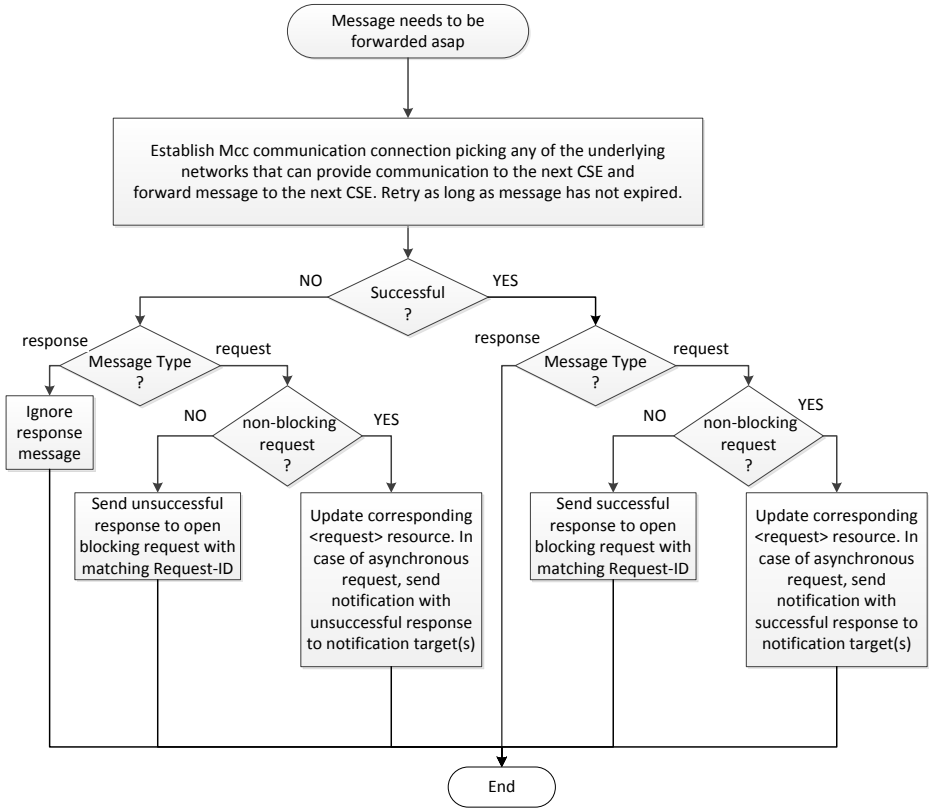


Figure 2 Forwarding of messages with 'ec' = 'immediate'.

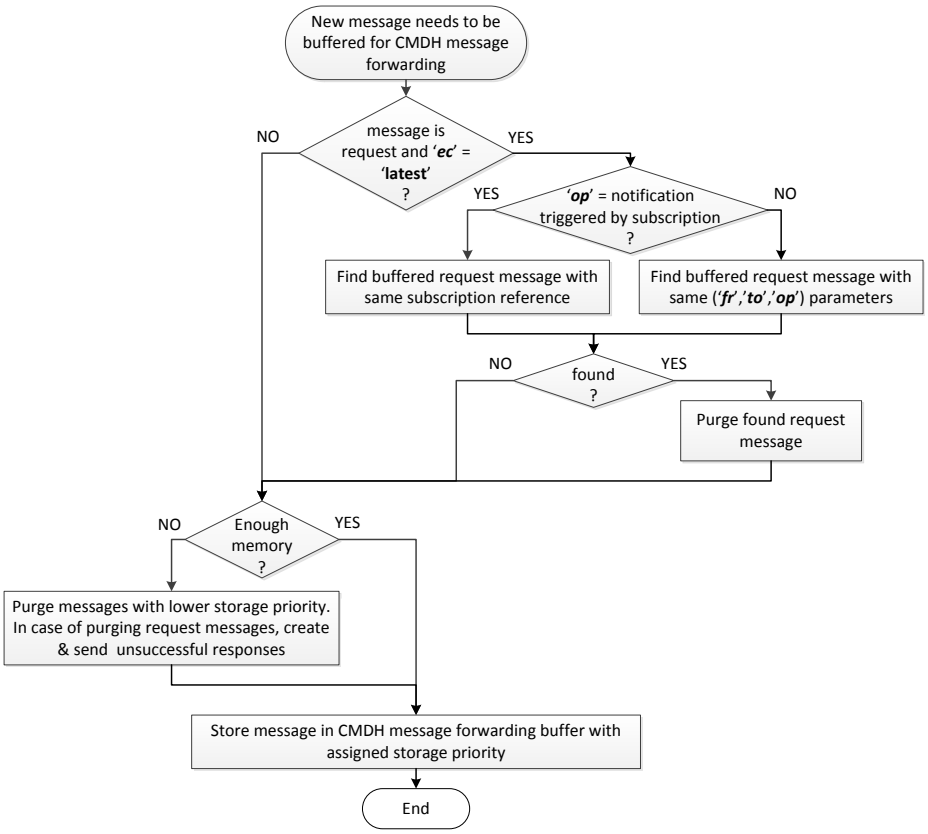


Figure 3 Buffering of messages for CMDH message forwarding.

H.2.5. Establishment of Mcc communication connection to another CSE

When a Mcc communication connection shall be established via a specific target network for forwarding a message of a specific event category indicated by the '***ec***' parameter of the message, the process of establishing the Mcc communication connection shall be governed by values contained in the 'backOffParameters' attribute of the [cmdhNwAccessRule] resource that was used to evaluate whether the message was allowed to be forwarded, as defined in step 2.2 in the procedure outlined in clause H.2.4. .

When connectivity via the selected target network to reach the next CSE has not already been established for other reasons, then the CSE that is trying to forward a message buffered for CMDH message forwarding shall establish a new Mcc communication connection via the selected target network for transporting oneM2M messages to the next CSE via a new Mcc instance. This communication connection shall be established following the procedures for authentication and security association using TLS or DTLS as defined in TS-0003 [[7]] taking into account provisioned security settings. The protocol mapping for transporting oneM2M specified messages via this instance of Mcc shall be selected according to the capabilities of the two end-points of the Mcc instance.

If establishing the Mcc communication connection via the selected target network fails, a new attempt to establish that communication connection shall only be made after waiting for a back-off time according to the value given in the 'back-off time' component of the 'backOffParameters' attribute of the effective [cmdhNwAccessRule] resource.

When establishing the Mcc communication connection via the selected target network still fails, for each subsequent new attempt to establish the Mcc communication connection without any successful attempts in-between, the back-off time shall be increased by the value given in the 'back-off time increment' component of the 'backOffParameters' attribute of the effective [cmdhNwAccessRule] resource.

The back-off time for waiting before making any new attempt to establish the Mcc communication connection via the selected target network shall not exceed the value given by the 'maximum back-off time' component of the 'backOffParameters' attribute of the effective [cmdhNwAccessRule] resource.

When the next CSE is hosted on a node for which a usable Mcc communication connection for forwarding a message to the next CSE can only be established by the next CSE itself, device triggering mechanisms as defined in TS-0001 shall be used.

In case the next CSE can only be reached via communication connections originating from the node that hosts the next CSE, while it is capable of processing incoming oneM2M messages, it is assumed that such a CSE establishes a polling channel as defined in TS-0001 in order to effectively receive unsolicited oneM2M messages.

1199

Annex I(infomative):
Template for one M2M resource type

7.3.x Resource Type <<resource name>>

7.3.x.1 Introduction

The few (2 statemanets are max.) will be copied from Architecture TS.

The detailed description can be found in clause x.x.x in Architecture TS [6].

Table 7.3.x.1-1: Data Type Definition of <<resource name>>

Data Type ID	File Name	Note
Actual Data Type ID	*CDT-<<resource name>>-v1_0_0-<<date of published>>.xsd*	*some note texts can be added here*

Table 7.3.x.1-2: Common Attributes on <<resource name>>

Attribute Name	Request Optionality				Default Value	Resource Specific Note
	C	R	U	D		
<<common attribute name1>>	*M,O,NP*	*M,O,NP*	*O*	*NP*		*Some Resource specific use of <<common attribute name 1>> as text.*
<<common attribute name 2>>			*O*	*NP*		*Another Resource specific use of <<common attribute name2>>*

Table 7.3.x.1-3: Resource Specific Attributes on <<resource name>>

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
<<common attribute name1>>	*M,O,NP*	*M,O,NP*	*O*	*NP*		*Some Resource specific use of <<common attribute name 1>> as text.*
<<common attribute name 2>>			*O*	*NP*		*Another Resource specific use of <<common attribute name2>>*

Table 7.3.x.1-4: Child resources of <<resource name>>

Child Resource Type	Child Resource Name	Multiplicity	Ref. to Resource Type Definition
<<resource type1>>	*[name](fixed)*	*Place 'multiplicity' at resource <<resource type>>*	*Place the reference to the Resource Type definition in Core Protocol TS*
Ex. <AE>	*[variable]*	*Ex. 0..n*	*Ex. Clause エラー! 参照元が見つかりません。*

7.3.x.2 <<resource name>> Resource Specific Procedure on CRUD Operations

This clause describes <<resource name>> resource specific behaviour for CRUD operations.

7.3.x.2.1 Create

The resource specific procedures shall be described here. Those can be mandatory attributes, expecting returning data types, and possible errors

7.3.x.2.2 Retrieve

The resource specific procedures shall be described here. Those can be mandatory attributes, expecting returning data types, and possible errors

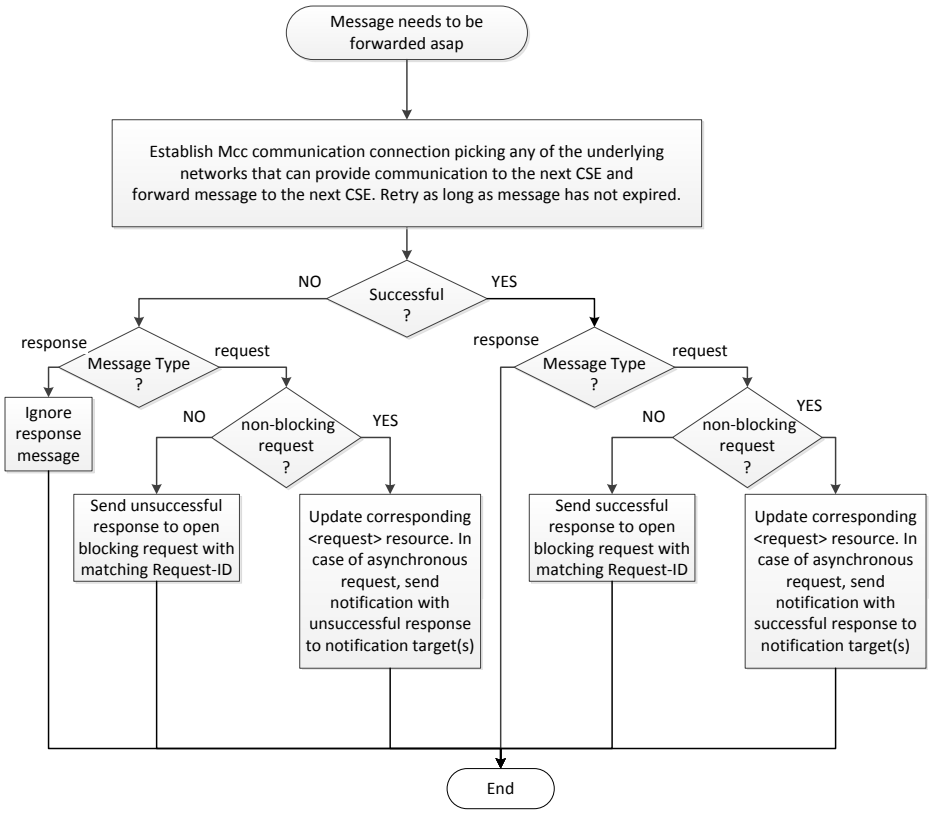
7.3.x.2.3 Update

The resource specific procedures shall be described here. Those can be mandatory attributes, expecting returning data types, and possible errors

7.3.x.2.4 Delete

The resource specific procedures shall be described here. Those can be mandatory attributes, expecting returning data types, and possible errors

List of tables and figures



History

Draft history (to be removed on publication)		
V.0.1.0	20 Jun 2013	Initial Version of TS
V.0.1.1	15 Aug 2013	Incorporate agreed contributions at TP#5 Ref: 2013-0020R01,2013-0026R02,2013-0028R02
V.0.1.2	12 Nov 2013	Incorporate agreed contrintuin at TP#7.1 Ref: 2013-0067R01
V.0.2.0	20 Dec 2013	Incorporate agreed contribution at TP#8 Ref: 2013-0092R01
V0.2.1	19 Feb 2014	Incorporate agreed contribution at PRO CC 8.x Ref: 2014-0106R01, 2014-0011R01
v0.3.0	5 May 2014	Incorporate agreed contribution at PRO #9 Ref: 2014-0031R01, 2014-0033R02
v0.3.1	25 May 2014	Incorporate agreed contributions at PRO CC 9.2 and 9.3 Ref: 2014-0117R02
v0.3.2	28 May 2014	Incorporate agreed contribution at PRO CC 9.4 Ref: 2014-0118R02
V0.4.0	22 Apr 2014	Incorporate agreed contributions at PRO #10 after 11 Apr 2014: Ref: PRO-2014-172R02(Template for oneM2M Resource) Ref:PRO-2014-151R01(common-operations)
V0.4.1	26 May 2014	Edit to use new template with line numbers
v0.4.2	30 May 2014	Incorporated agreed contributions at PRO #10.5 Ref:PRO-2014-202R01-CRUD_common_operations Ref:PRO-2014-205R01-Resource_Type_Definition_Template Ref:PRO-2014-199R02- Primitive_templates_and_procedure_outlines
V0.4.3	4 Jun 2014	Incorporated agreed contribuition at PRO #10.7 Ref: PRO-2014-195R03-Child_Resource_Conversion_Rules Added Editor's Notes to indicate name of leaders

V0.5.0	14 Jun 2014	Incorporated agreed contributions at PRO #11.0: Ref: PRO-2014-195R03- Child Resource Conversion Rules Ref: PRO-2014-216R02- Response Status Clause Ref: PRO-2014-217R03- Attribute Creation and Deletion Ref: PRO-2014-220R02- management-common-operations Ref: PRO-2014-229R03- Location Request on Mcn Reference Point Ref: PRO-2014-242- Correction to Table 7.2.1.1-2 in TS-0004 Ref: PRO-2014-219R02- Resource Discovery Procedure Ref: PRO-2014-224R01- design_principles_scalability Ref: PRO-2014-218R05- Announcement Procedures Ref: PRO-2014-249R03- Resource Type Definition Template Update Ref: PRO-2014-251R01- remoteCSE Resource Type Definition Ref: PRO-2014-236R01-Stage 3 text for Resource Type <statsConfig> Ref: PRO-2014-237R02-Stage 3 text for Resource Type <eventConfig> Ref: PRO-2014-238R02-Stage 3 text for Resource Type <statsCollect> Incorporated agreed contribuition at PRO #11 (by Nobu U) Ref: PRO-2014-0152R02-group-management-procedures Ref: PRO-2014-0194R04-CSEBase_Resource_Type_Definition Ref: PRO-2014-0221R03-mgmtObj-management-procedures Ref: PRO-2014-0228R01- LocationPolicy_Resource_Primitive Ref: PRO-2014-0230R03- mgmtCmd_input_TS0004 Ref: PRO-2014-0231R04-Request_resource_type_definition Ref: PRO-2014-0235R02- pollingChannel_procedures Incorporated PRO-2014-0239R03-Resource_Type_XSD_example in Annex with Editor's changes Incorporated agreed contributions (missing in R01) Ref: PRO-2014-223R01-node-resource-type-procedures Ref: PRO-2014-227R02-cmdh_policies_TS0004
v.0.5.1	17 Jun 2014	Incorporated pended Agreed contribution: Ref: PRO-2014-222R02- Primitive_templates_and_procedure_outlines_modification Removed resolved Editor's Notes and unused sub-clauseses.

V0.5.2	25 Jun 2014	Clean-up done by ***editHelp!*** e-mail: mailto:edithelp@etsi.org Incorporated agreed contribuition at PRO #11.1 (by Nobu U) Ref: PRO-2014-0252R02-XML_Schema_Further_Changes Ref: PRO-2014-0262R01-schedule Fixed reference link between section 1.1 and section 7.3.8 • PRO-2014-0228R01- LocationPolicy_Resource_Primitive Added missing annex • PRO-2014-229R03- Location Request on Mcn Reference Point Incorporated agreed contribution at PRO #11.2: Ref: PRO-2014-0274- TS0004 Abbreviations
V0.5.3	4 Jul 2014	Incorporated agreed contribution at PRO #11.3 (by Shingo): Ref: PRO-2014-0276R01-addtional-common-operation Ref: PRO-2014-0281R01-Delivery_resource_type_definition Ref: PRO-2014-0272R02-Messages_parameters_data_types_claus Ref: PRO-2014-0277- clause-re-arrangement PRO-2014-0287R021-TS-0004_Section_7.3_Cleanup PRO-2014-0261R01-generic_procedures

V0.5.4	28 Jul 2014	Incorporated agreed contributions before PRO #12 (by Shingo) Ref: Ref:PRO-2014-0268R04-Add_common_attributes_on_ResType_template Ref:PRO-2014-0272R02-Messages_parameters_data_types_clause Ref:PRO-2014-0286R03-Container_and_Container_Instance_ResType Ref:PRO-2014-0287R02-TS-0004_Section_7_3_Cleanup Ref:PRO-2014-0298R02-Generic_Procedure_Clean_Up Ref:PRO-2014-0304R03-data_types (this has NOT been incorporated.) Ref:PRO-2014-0308R06-subscription_definition_and_procedures Ref:PRO-2014-0320R02-Request-related_common_procedures Ref:PRO-2014-0324R02-notification_definition_and_procedures Ref:PRO-2014-0326R02-procedure-for-fanOutPoint Ref:PRO-2014-0328-Request_resource_type_definition_-_Update Ref:PRO-2014-0331R01-Operation_Applicability_Reference Ref:PRO-2014-0334R04- New_section_for_services_provided_by_the_underlying_network. Ref:PRO-2014-0335R02-Clarify_M2M-Ext-ID Ref:PRO-2014-0336R01-Clarify_Device_Triggering Ref:PRO-2014-0337R01-Notification_procedure_for_subscription Ref:PRO-2014-0342R01-Clarification_on_entrance_for_resource_announcement Ref:PRO-2014-0347R01-Announced_Resource_Procedures Ref:PRO-2014-0355R01-Corrections_of_locationPolicy_Resource Ref:PRO-2014-0373R01-Res_Def_Template_Update

V0.5.5	29 Jul 2014	Incorporated agreed contributions on July 29 at PRO #12 (by Nobu) PRO-2014-0309R03-Notify_Request_Re-targeting PRO-2014-0310R02-AE_Resource_Type_and_Procedure PRO-2014-0311R03-TS-0004_Cleanup PRO-2014-0312R01-accessControlPolicy_Resource_Type_and_Procedure PRO-2014-0356R02-Area-based_notification_Service_of_OMA_Location_API PRO-2014-0357R01- Correction_of_the_parameters_at_the_Device_Triggering_commands PRO-2014-0364-remoteCSE_XSD PRO-2014-0366R02-Extensibility_issues PRO-2014-0375R02-oneM2M_enumeration_types PRO-2014-0380R03-Correction_of_notification_procedure PRO-2014-0381R01-Resource_Discovery_Procedure PRO-2014-0383R01-Clarify_the_Trigger-Recipient-ID PRO-2014-0385R01-restructuring_notification_text PRO-2014-0389-Request_Applicability_on_Attributes
V0.5.6	30 Jul 2014	Incorporated agreed contributions on July 30 at PRO #12 (by Nobu) PRO-2014-0315R08-TS-0004_Annex_D_Cleanup PRO-2014-0327R07-resources-for-mgmtObj PRO-2014-0329R01-Delivery_resource_type_definition_-_Update PRO-2014-0340R02- Resource_definition_of_AreaNwkInfo_and_AreaNwkDeviceInfo PRO-2014-0343R04-Synchronization_of_announced_attribute PRO-2014-0345R02-devInfo_eventLog_Management_Resource_Procedures PRO-2014-0346R04-TS-0004-Mgmt_obj_common_operations_updates PRO-2014-0348R03-mgmtCmd_update_TS0004 PRO-2014-0367R04-Procedures_for_accessing_resources PRO-2014-0388-Enumeration_Data_Type_Definitions PRO-2014-0390-cleaning-for-the-new-template PRO-2014-0391R01-response-status-code-cleaning PRO-2014-0394R02-Cleaning_CSEBase_resource_for_the_new_template PRO-2014-0395R02-Cleaning_remoteCSE_resource_for_the_new_template PRO-2014-0397R01- 7_3_Cleanup_for_subscription_pollingChannel_pollingChannelURI PRO-2014-0398R01-Clean_Up_of_locationPolicy_Resource

V0.6.0	01 Aug 2014	Incorporated agreed contributions on July 31st at PRO #12 (by Shingo) PRO-2014-0314R02-MIME_type_for_oneM2M_resource_representation PRO-2014-0372R02-Status_Code_Cleanup PRO-2014-0374R03-CMDH_Procedures PRO-2014-0392R03-_schedule_resource_default_text PRO-2014-0399R01-_container_contentInstance_update PRO-2014-0401R02-clean_up_of_announced_resource_type PRO-2014-0403R01-Response_Status_Update PRO-2014-0404R01-authorizedNode_Resource_Type PRO-2014-0405R01-m2mServiceSubscriptionProfile_Resource_Type PRO-2014-0406-Management_common_operations_and_status_codes_updates And, add correction miisng implementation of PRO-2014-0304R03
V0.6.`	01 Aug 2014	Prepared for Initial Release. Same content of V0.6.0

1386