



## ONE M2M TECHNICAL SPECIFICATION

Document Number	TS-0004-Protocol_Core_Specification-V-0.7.0
Document Name:	Service Layer Protocol Core Specification
Date:	2014-Sep-09
Abstract:	The present document specifies the communication protocol(s) for oneM2M compliant Systems, M2M Applications, and/or other M2M Systems. The present document also specifies the common data formats, interfaces and message sequences to support reference points(s) defined by oneM2M.

This Specification is provided for future development work within oneM2M only. The Partners accept no liability for any use of this Specification.

The present document has not been subject to any approval process by the oneM2M Partners Type 1. Published oneM2M specifications and reports for implementation should be obtained via the oneM2M Partners' Publications Offices.

#### 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>

#### 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.

© 2013, 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.

---

## Contents

Contents	3
1 Scope	12
2 References	12
2.1 Normative references	12
2.2 Informative references	13
3 Definitions, symbols, abbreviations and acronyms	13
4 Conventions	15
5 Protocol Design Principles and Requirements	15
5.1 Introduction	15
5.2 API Design Guidelines	16
5.3 Primitives	16
5.3.1 Introduction	16
5.3.2 Primitives modelling	17
5.3.3 Primitive principles	18
5.4 Design Principles	18
5.4.1 Scalability	18
5.4.2 Extensibility	18
5.4.3 Efficiency	19
6 oneM2M Protocols/API Overview	19
6.1 Introduction	19
6.2 M2M Identifiers	19
6.3 Common Data Types	20
6.3.1 Simple Data Types incorporated from XML Schema	20
6.3.2 oneM2M Simple Data Types	22
6.3.3 oneM2M Enumerated Data Types	23
6.3.3.1 Introduction	23
6.3.3.2 Enumeration Type Definitions	24
6.3.3.2.1 m2m:resourceType	24
6.3.3.2.2 m2m:commaList	24
6.3.3.2.3 m2m:cseTypeID	25
6.3.3.2.4 m2m:locationSource	25
6.3.3.2.5 m2m:eventCatType	25
6.3.3.2.6 m2m:operation	26
6.3.3.2.7 m2m:responseType	26
6.3.3.2.8 m2m:resultContent	26
6.3.3.2.9 m2m:discResType	26
6.3.3.2.10 m2m:statusCode	27
6.3.3.2.11 m2m:requestStatus	27
6.3.3.2.12 m2m:operationResult	27
6.3.3.2.13 m2m:memberType	27
6.3.3.2.14 m2m:consistencyStrategy	28
6.3.3.2.15 m2m:cmdType	28
6.3.3.2.16 m2m:execModeType	28
6.3.3.2.17 m2m:execStateType	29
6.3.3.2.18 m2m:execResultType	29
6.3.3.2.19 m2m:pendingNotification	30
6.3.3.2.20 m2m:notificationContentType	30
6.3.3.2.21 m2m:resourceStatus	30
6.3.3.2.22 m2m:operation	31
6.3.3.2.23 m2m:attribute	31
6.3.3.2.24 m2m:status	31
6.3.3.2.25 m2m:batteryStatus	31
6.3.3.2.26 m2m:mgmtDefinition	32

6.3.3.2.27.	m2m:logTypeId	32
6.3.3.2.28.	m2m:logStatus	32
6.3.3.2.29.	m2m:listOfBoolean	33
6.3.4.	Complex Data Types	33
6.3.4.1.	m2m:filterCriteria	33
6.3.4.2.	m2m:eventCat	34
6.3.4.3.	m2m:deliveryMetaData	34
6.3.4.4.	m2m:aggregatedRequest	34
6.3.4.5.	m2m:metaInformation	34
6.3.4.6.	m2m:content	34
6.3.4.7.	m2m:batchNotify	34
6.3.4.8.	m2m:eventNotificationCriteria	34
6.3.4.9.	m2m:aggregatedNotification	35
6.3.4.10.	m2m:singleNotification	35
6.3.4.11.	m2m:notificationEvent	35
6.3.4.12.	m2m:operationMonitor	35
6.3.4.13.	m2m:externalID	36
6.3.4.14.	m2m:actionStatus	36
6.3.4.15.	m2m:responseStatus	36
6.3.4.16.	m2m:anyArgType	36
6.3.4.17.	m2m:anyArgListType	37
6.3.4.18.	m2m:resetArgsType	37
6.3.4.19.	m2m:rebootArgsType	37
6.3.4.20.	m2m:uploadArgsTypes	37
6.3.4.21.	m2m:downloadArgsType	37
6.3.4.22.	m2m:softwareInstallArgsType	38
6.3.4.23.	m2m:softwareUpdateArgsType	38
6.3.4.24.	m2m:softwareUninstallArgsType	38
6.3.4.25.	m2m:execReqArgsType	38
6.3.4.26.	m2m:execReqArgsListType	39
6.3.4.27.	m2m:mgmtLinkRef	39
6.3.4.28.	m2m:setOfAcrs	39
6.3.4.29.	m2m:	40
6.3.4.30.	m2m:	40
6.3.5.	Resource common attributes	40
6.4.	Message parameter data types	40
6.4.1.	Request message parameter data types	41
6.4.2.	Response message parameter data types	42
6.4.3.	resourceData type	42
6.4.3.1.	6.4.4.1 Description	42
6.4.3.2.	Reference	42
6.4.3.3.	Usage Example	43
6.5.	Resource Data Types	43
6.5.1.	regularResourceType	43
6.5.1.1.	Description	43
6.5.1.2.	Reference	43
6.5.1.3.	Usage Example	43
6.5.2.	announceableResourceType	44
6.5.2.1.	Description	44
6.5.2.2.	Reference	44
6.5.2.3.	Usage Example	44
6.5.3.	subordinateResourceType	44
6.5.3.1.	Description	44
6.5.3.2.	Reference	44
6.5.3.3.	Usage Example	44
6.6.	Response Status Codes	44
6.6.1.	Introduction	44
6.6.2.	RSC Framework Overview	45
6.6.3.	Current Response Status Codes	45
6.6.3.1.	Successful Responses	46
6.6.3.2.	Un-successful Response Codes	46
6.6.3.3.	Acknowledgement Responses	47

6.7.	MIME Types for oneM2M protocols.....	47
6.7.1.	application/vnd.onem2m-resource-data+xml .....	47
7	oneM2M Resource Types .....	49
7.1.	Introduction.....	49
7.2.	Prerequisites.....	49
7.2.1.	Primitive format and procedure outlines.....	49
7.2.1.1.	Primitive format.....	49
7.2.1.1.1.	Request primitive format.....	49
7.2.1.1.2.	Response primitive format.....	49
7.2.1.2.	Description of Generic Procedures.....	50
7.2.1.2.1.	Generic Resource Request Procedure for Originator.....	50
7.2.1.2.2.	Generic Request Procedure for Receiver.....	51
7.2.2.	Common operations .....	54
7.2.2.1.	Originator Actions .....	54
7.2.2.1.1.	Compose Request primitive.....	54
7.2.2.1.2.	Send a Request to the Receiver CSE.....	54
7.2.2.1.3.	Wait for Response primitive.....	54
7.2.2.1.4.	Retrieve the <request> resource.....	55
7.2.2.2.	Receiver CSE actions .....	55
7.2.2.2.1.	Check the validity of received request primitive.....	55
7.2.2.2.2.	Create Response Accepted.....	55
7.2.2.2.3.	Send Response primitive.....	55
7.2.2.2.4.	Create <request> resource locally.....	55
7.2.2.2.5.	Create a Successful Response (Acknowledgement).....	56
7.2.2.2.6.	Send Response primitive (Acknowledgement).....	56
7.2.2.2.7.	Update <request> resource.....	57
7.2.2.2.8.	Forwarding.....	57
7.2.2.3.	Hosting CSE actions.....	57
7.2.2.3.1.	Check existence of the addressed resource.....	57
7.2.2.3.2.	Check validity of resource representation for CREATE.....	57
7.2.2.3.3.	Check validity of resource representation for UPDATE.....	58
7.2.2.3.4.	Create the resource.....	59
7.2.2.3.5.	Retrieve the resource.....	59
7.2.2.3.6.	Update the resource.....	60
7.2.2.3.7.	Delete the resource.....	60
7.2.2.3.8.	Notify re-targeting.....	61
7.2.2.3.9.	Announce the resource or attribute.....	61
7.2.2.3.10.	De-announce the resource or attribute.....	62
7.2.2.3.11.	Create a successful Response.....	63
7.2.2.3.12.	Create an unsuccessful Response.....	63
7.2.2.3.13.	Resource Discovery Procedure.....	63
7.2.2.4.	Management Common Operations.....	63
7.2.2.4.1.	Identify the managed entity and the management protocol.....	63
7.2.2.4.2.	Locate the external management objects to be managed on the managed entity.....	64
7.2.2.4.3.	Establish a management session with the managed entity or management server.....	64
7.2.2.4.4.	Send the management request(s) to the managed entity corresponding to the received Request primitive.....	64
7.3.	Resource Type-specific procedures and definitions .....	65
7.3.1.	Resource Type accessControlPolicy .....	65
7.3.1.1.	Introduction.....	65
7.3.1.2.	accessControlPolicy Resource Specific Procedure on CRUD Operations.....	65
7.3.1.2.1.	Create.....	65
7.3.1.2.2.	Retrieve.....	66
7.3.1.2.3.	Update.....	66
7.3.1.2.4.	Delete.....	66
7.3.2.	Resource Type CSEBase .....	66
7.3.2.1.	Introduction.....	66
7.3.2.2.	Operations .....	67
7.3.2.2.1.	Create.....	67
7.3.2.2.2.	Retrieve.....	67
7.3.2.2.3.	Update.....	67

7.3.2.2.4.	Delete.....	68
7.3.3.	Resource Type remoteCSE.....	68
7.3.3.1.	Introduction.....	68
7.3.3.2.1.	Create.....	69
7.3.3.2.2.	Retrieve .....	70
7.3.3.2.3.	Update .....	70
7.3.3.2.4.	Delete.....	70
7.3.4.	Resource Type AE.....	70
7.3.4.1.	Introduction.....	70
7.3.4.2.	AE Resource Specific Procedure on CRUD Operations.....	71
7.3.4.2.1.	Create.....	71
7.3.4.2.2.	Retrieve .....	71
7.3.4.2.3.	Update .....	71
7.3.4.2.4.	Delete.....	71
7.3.4.2.5.	Notify.....	72
7.3.5.	Resource Type container.....	72
7.3.5.1.	Introduction.....	72
7.3.5.2.	<container> Resource Specific Procedure on CRUD Operations.....	73
7.3.5.2.1.	Create.....	73
7.3.5.2.2.	Retrieve .....	73
7.3.5.2.3.	Update .....	73
7.3.5.2.4.	Delete.....	73
7.3.6.	Resource Type contentInstance .....	74
7.3.6.1.	Introduction.....	74
7.3.6.2.	<contentInstance> Resource Specific Procedure on CRUD Operations .....	74
7.3.6.2.1.	Create.....	74
7.3.6.2.2.	Retrieve .....	75
7.3.6.2.3.	Update .....	75
7.3.6.2.4.	Delete.....	75
7.3.7.	Resource Type subscription.....	75
7.3.7.1.	Introduction.....	75
7.3.7.2.	<subscription> Resource Specific Procedure on CRUD Operations.....	76
7.3.7.2.1.	Create.....	76
7.3.7.2.2.	Retrieve .....	77
7.3.7.2.3.	Update .....	77
7.3.7.2.4.	No change from the generic procedures in clause 7.2.1.2.2.Delete .....	77
7.3.8.	Resource Type schedule.....	78
7.3.8.1.	Introduction.....	78
7.3.8.2.	Resource Specific Procedure on CRUD Operations .....	80
7.3.8.2.1.	Create.....	80
7.3.8.2.2.	Retrieve .....	80
7.3.8.2.3.	Update .....	80
7.3.8.2.4.	Delete.....	80
7.3.9.	Resource Type locationPolicy .....	81
7.3.9.1.	Introduction.....	81
7.3.9.2.	Operations .....	82
7.3.9.2.1.	Create.....	82
7.3.9.2.2.	Retrieve .....	83
7.3.9.2.3.	Update .....	83
7.3.9.2.4.	Delete.....	83
7.3.10.	Resource Type delivery.....	84
7.3.10.1.	Introduction.....	84
7.3.10.2.	Resource Specific Procedure on CRUD Operations .....	84
7.3.10.2.1.	Create.....	84
7.3.10.2.2.	Retrieve .....	85
7.3.10.2.3.	Update .....	85
7.3.10.2.4.	Delete.....	85
7.3.11.	Resource Type request .....	86
7.3.11.1.	Introduction.....	86
7.3.11.2.	Resource Specific Procedure on CRUD Operations .....	87
7.3.11.2.1.	Create.....	87
7.3.11.2.2.	Retrieve .....	88

7.3.11.2.3.	Update .....	88
7.3.11.2.4.	Delete.....	89
7.3.12.	Resource Type group .....	89
7.3.12.1.	Introduction.....	89
7.3.12.2.	Resource Specific Procedure on CRUD Operations .....	90
7.3.12.2.1.	Create.....	90
7.3.12.2.2.	Retrieve .....	90
7.3.12.2.3.	Update .....	90
7.3.12.2.4.	Delete.....	91
7.3.13.	Resource Type fanOutPoint.....	91
7.3.13.1.	Introduction.....	91
7.3.13.2.	fanOutPoint operations.....	91
7.3.13.2.1.	Validate the member types.....	91
7.3.13.2.2.	Sub-group creation for members residing on the same CSE .....	91
7.3.13.2.3.	Assign URI for aggregation of notification .....	91
7.3.13.2.4.	Fanout Request to each member.....	92
7.3.13.3.	<fanOutPoint> Resource Specific Procedure on CRUD Operations .....	92
7.3.13.3.1.	Create.....	92
7.3.13.4.1.	Update .....	93
7.3.13.4.2.	Delete.....	94
7.3.14.	Resource Type mgmtObj .....	94
7.3.14.1.	Introduction.....	94
7.3.14.2.	Resource Specific Procedure on CRUD Operations .....	95
7.3.14.2.1.	Create.....	95
7.3.14.2.2.	Retrieve .....	96
7.3.14.2.3.	Update .....	96
7.3.14.2.4.	Delete.....	96
7.3.15.	Resource Type mgmtCmd .....	97
7.3.15.1.	Introduction.....	97
7.3.15.2.	Resource Specific Procedures.....	99
7.3.15.2.1.	Create.....	99
7.3.15.2.2.	Retrieve .....	99
7.3.15.2.3.	Update .....	99
7.3.15.2.4.	Delete.....	100
7.3.16.	Resource Type execInstance.....	101
7.3.16.1.	Introduction.....	101
7.3.16.2.	Resource Specific Procedures.....	101
7.3.16.2.1.	Update (Cancel).....	101
7.3.16.2.2.	Retrieve .....	102
7.3.16.2.3.	Delete.....	102
7.3.17.	Resource Type node.....	103
7.3.17.1.	Introduction.....	103
7.3.17.2.	Resource Specific Procedure on CRUD Operations .....	103
7.3.17.2.1.	Create.....	103
7.3.17.2.2.	Retrieve .....	104
7.3.17.2.3.	Update .....	104
7.3.17.2.4.	Delete.....	104
7.3.18.	Resource Type m2mServiceSubscriptionProfile.....	104
7.3.18.1.	Introduction.....	104
7.3.18.2.	Operations .....	105
7.3.18.2.1.	Create.....	105
7.3.18.2.2.	Retrieve .....	105
7.3.18.2.3.	Update .....	105
7.3.18.2.4.	Delete.....	105
7.3.19.	Resource Type authorizedNode.....	105
7.3.19.1.	Introduction.....	105
7.3.19.2.	Operations .....	106
7.3.19.2.1.	Create.....	106
7.3.19.2.2.	Retrieve .....	106
7.3.19.2.3.	Update .....	106
7.3.19.2.4.	Delete.....	107
7.3.20.	Resource Type pollingChannel .....	107

7.3.20.1.	Introduction.....	107
7.3.20.2.	Operations .....	107
7.3.20.2.1.	Create.....	107
7.3.20.2.2.	Retrieve .....	107
7.3.20.2.3.	Update .....	108
7.3.20.2.4.	Delete.....	108
7.3.21.	Resource Type pollingChannelURI .....	108
7.3.21.1.	Introduction.....	108
7.3.21.2.	Operations .....	108
7.3.21.2.1.	Create.....	108
7.3.21.2.2.	Retrieve .....	108
7.3.21.2.3.	Update .....	109
7.3.21.2.4.	Delete.....	109
7.3.22.	Resource Type statsConfig .....	109
7.3.22.1.	Introduction.....	109
7.3.22.2.	<statsConfig> resource-specific procedure on CRUD operations .....	110
7.3.22.2.1.	Create.....	110
7.3.22.2.2.	Retrieve .....	110
7.3.22.2.3.	Update .....	110
7.3.22.2.4.	No change from the generic procedures in clause 7.2.1.2.2.Delete .....	110
7.3.23.	Resource Type eventConfig.....	110
7.3.23.1.	Introduction.....	110
7.3.23.2.	<eventConfig> resource-specific procedure on CRUD operations.....	111
7.3.23.2.1.	Create.....	111
7.3.23.2.2.	Retrieve .....	111
7.3.23.2.3.	Update .....	111
7.3.24.	Resource Type statsCollect.....	111
7.3.24.1.	Introduction.....	111
7.3.24.2.	<statsCollect> resource-specific procedure on CRUD operations.....	112
7.3.24.2.1.	Create.....	112
7.3.24.2.2.	Retrieve .....	112
7.3.24.2.3.	Update .....	112
7.3.24.2.4.	Delete.....	113
7.3.25.	Announced Resource Type.....	113
7.3.25.1.	Introduction.....	113
7.3.25.2.	Resource Specific Procedure on CRUD Operations .....	114
7.3.25.2.1.	Create.....	114
7.3.25.2.2.	Retrieve .....	114
7.3.25.2.3.	Update .....	114
7.3.25.2.4.	No change from the generic procedures in clause 7.2.1.2.2.Delete .....	114
7.4.	Notification definition and procedures .....	115
7.4.1.	Definition of Notification.....	115
7.4.1.1.	Introduction.....	115
7.4.2.	Notification Procedures.....	115
7.4.2.1.	Notification for Subscription .....	115
7.4.2.2.	Subscription Verification during Subscription Creation.....	117
7.4.2.3.	Notification for Subscription Deletion.....	118
7.4.2.4.	Notification for Asynchronous Non-blocking Request.....	118
7.4.2.5.	Notification for subscription via group.....	118
8	oneM2M Resource Types .....	119
8.1.	Introduction.....	119
8.2.	Short Names.....	119
8.2.1.	Introduction.....	119
8.2.2.	Primitive parameters .....	119
8.2.3.	Resource attributes .....	120
8.2.4.	Resource types.....	124
8.2.5.	Complex data types members.....	126
Annex A (normative):	Resource attributes .....	127
Annex B (normative):	Device Triggering .....	135
B.1.	Providing Device Triggering service by means of 3GPP networks.....	135



B.1.1.	Introduction.....	135
B.1.2.	Device Action Request command.....	135
B.1.3.	Device Action Answer command.....	135
B.1.4.	Device Notification Request command.....	135
B.1.5.	Device Notification Answer command.....	135
Annex C (informative): XML Examples.....		136
C.1.	XML Schema for container resource type.....	136
C.2.	Container resource that conforms to the Schema given above (see clause C.1).....	137
Annex D (Normative): <mgmtObj> Resource Specializations.....		138
D.1.	Introduction.....	138
D.2.	Resource [firmware].....	138
D.2.1.	Introduction.....	138
D.2.2.	Resource Specific Procedure on CRUD Operations.....	138
D.2.2.1.	Create.....	138
D.2.2.2.	Update.....	139
D.2.2.3.	Retrieve.....	139
D.2.2.4.	Delete.....	139
D.3.	Resource [software].....	139
D.3.1.	Introduction.....	139
D.3.2.	Resource Specific Procedure on CRUD Operations.....	140
D.3.2.1.	Create.....	140
D.3.2.2.	Update.....	140
D.3.2.3.	Retrieve.....	141
D.3.2.4.	Delete.....	141
D.4.	Resource [memory].....	141
D.4.1.	Introduction.....	141
D.4.2.	Resource Specific Procedure on CRUD Operations.....	141
D.4.2.1.	Create.....	141
D.4.2.2.	Update.....	142
D.4.2.3.	Retrieve.....	142
D.4.2.4.	Delete.....	142
D.5.	Resource [areaNwkInfo].....	142
D.5.1.	Introduction.....	142
D.5.2.	Resource Specific Procedure on CRUD Operations.....	143
D.5.2.1.	Create.....	143
D.5.2.2.	Update.....	143
D.5.2.3.	Retrieve.....	143
D.5.2.4.	Delete.....	143
D.6.	Resource [areaNwkDeviceInfo].....	143
D.6.1.	Introduction.....	143
D.6.2.	Resource Specific Procedure on CRUD Operations.....	144
D.6.2.1.	Create.....	144
D.6.2.2.	Update.....	144
D.6.2.3.	Retrieve.....	144
D.6.2.4.	Delete.....	145
D.7.	Resource [battery].....	145
D.7.1.	Introduction.....	145
D.7.2.	Resource Specific Procedure on CRUD Operations.....	145
D.7.2.1.	Create.....	145
D.7.2.2.	Update.....	145
D.7.2.3.	Retrieve.....	146
D.7.2.4.	Delete.....	146
D.8.	Resource [deviceInfo].....	146
D.8.1.	Introduction.....	146
D.8.2.	Resource Specific Procedure on CRUD Operations.....	147
D.8.2.1.	Create.....	147
D.8.2.2.	Update.....	147
D.8.2.3.	Retrieve.....	147
D.8.2.4.	Delete.....	147
D.9.	Resource [deviceCapability].....	147

D.9.1.	Introduction.....	147
D.9.2.	Resource Specific Procedure on CRUD Operations.....	148
D.9.2.1.	Create.....	148
D.9.2.2.	Update.....	148
D.9.2.3.	Retrieve.....	149
D.9.2.4.	Delete.....	149
D.10.	Resource [reboot].....	149
D.10.1.	Introduction.....	149
D.10.2.	Resource Specific Procedure on CRUD Operations.....	149
D.10.2.1.	Create.....	150
D.10.2.2.	Update.....	150
D.10.2.3.	Retrieve.....	150
D.10.2.4.	Delete.....	150
D.11.	Resource [eventLog].....	150
D.11.1.	Introduction.....	150
D.11.2.	Resource Specific Procedure on CRUD Operations.....	151
D.11.2.1.	Create.....	151
D.11.2.2.	Update.....	151
D.11.2.3.	Retrieve.....	151
D.11.2.4.	Delete.....	151
D.12.	Resource [cmdhPolicy].....	152
D.12.1.	Resource [activeCmdhPolicy].....	152
D.12.2.	Resource [cmdhDefaults].....	153
D.12.3.	Resource [cmdhDefEcValue].....	153
D.12.4.	Resource [cmdhEcDefParamValues].....	154
D.12.5.	[cmdhLimits] Resource.....	154
D.12.6.	Resource [cmdhNetworkAccessRules].....	155
D.12.7.	Resource [cmdhNwAccessRule].....	155
D.12.8.	Resource [cmdhBuffer].....	156
Annex E (informative) Procedures for accessing resources.....		157
E.1.	Accessing Resources in CSEs – Blocking Requests.....	157
E.2.	Accessing Resources in CSEs - non-Blocking Requests.....	157
E.2.1.	Synchronous Case.....	157
E.2.2.	Asynchronous Case.....	161
Annex F (infomative): Guidelines for one M2M resource type XSD.....		162
Annex G (Normative): Location Request.....		164
G.1.	Location Request by means of OMA-REST-NetAPI-TerminalLocation Interface.....	164
G.1.1.	Introduction.....	164
G.1.2.	Resource Structure of OMA NetAPI for Terminal Location.....	164
G.1.3.	Procedures for Terminal Location.....	167
G.1.3.1.	Request in a Single Query toward a Location Server.....	167
G.1.4.	Subscribe to Notifications for Periodic Location Updates.....	167
G.1.5.	Subscribe to Notifications for Area Updates.....	168
Annex H (Normative): CMDH Message Processing.....		170
H.2.1.	Applicability of CMDH processing.....	171
H.2.2.	Partitioning of CMDH processing.....	171
H.2.3.	CMDH message validation procedure.....	173
H.2.4.	CMDH message forwarding procedure.....	178
H.2.5.	Establishment of Mcc communication connection to another CSE.....	186
Annex I (infomative): Template for one M2M resource type.....		188
7.3.x	Resource Type <<resource name>>.....	188
7.3.x.1	Introduction.....	188
7.3.x.2	<<resource name>> Resource Specific Procedure on CRUD Operations.....	189
7.3.x.2.1	Create.....	189
7.3.x.2.2	Retrieve.....	189
7.3.x.2.3	Update.....	189
7.3.x.2.4	Delete.....	189
List of tables and figures.....		190

History .....195

---

---

# 1 Scope

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

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](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>  
<2<sup>nd</sup> symbol>       <2<sup>nd</sup> Explanation>  
<3<sup>rd</sup> symbol>       <3<sup>rd</sup> 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 management(see clause 7.2.2.4)

## 5.2. API Design Guidelines

The following are the guidelines for designing APIs:

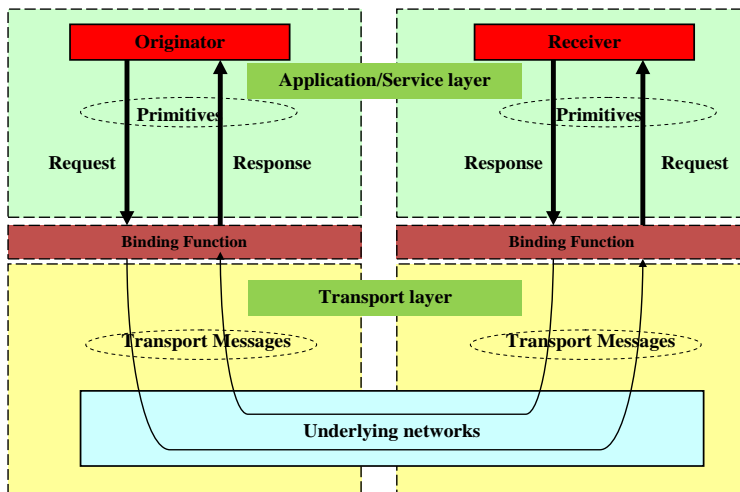
- 1) API shall follow the principle of RESTful architecture, as described in [i.2].
- 2) APIs shall define how to address resources and how to manipulate resources, in accordance with oneM2M TS-0001 [6]; the resource is identified by a Universal Resource Identifier (URI), [2].
- 3) Resource has a representation (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.

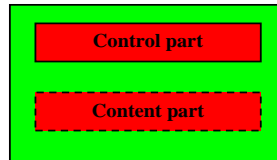
### 5.3.2. Primitives modelling

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 with Handling 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)	A globally unique ID as specified in [6]
App-ID	xs:string (see clause 6.3.2)	The identifier is specified in [6]
AE-ID	m2m:id (see clause 6.3.2)	A globally unique ID as specified in [6]
CSE-ID	m2m:id (see clause 6.3.2)	A globally unique ID as specified in [6]
M2M-Node-ID	TBD (see clause 6.3.2)	A globally unique ID as specified in [6]
M2M-Sub-ID	m2m:id (see clause 6.3.2)	A globally unique ID as specified in [6]
M2M-Request-ID	TBD (see clause 6.3.2)	A unique ID as specified in [6]
M2M-Ext-ID	M2m:externalId (see clause 6.3.4)	The identifier is specified in [6]
UNetwork-ID	m2m:id (see clause 6.3.2)	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 6.3.1-1: 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 $n \geq 0$ . 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. oneM2M Simple Data Types

Table 6.3.2-1 describes oneM2M-specific simple data type definitions.

**Table 6.3.2-1: 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 networkaccessIdentifier is a standard way of identifying users who request access to a network as specified at IETF RFC 4282 [18].
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:ipv4	IPv4 address string with optional CIDR suffix	10.125.0.0/16, 122.77.12.1	Required in m2m:acr
m2m:ipv6	IPv6 address string with optional CIDR suffix	::/0, Fadf:ddd0::/32, abcd:ffff:abb0:aaaa::/64	Required in m2m:acr

### 6.3.3. oneM2M Enumerated Data Types

#### 6.3.3.1. Introduction

The oneM2M Enumeration Types are defined as extension 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 6.3.3.1-1 shows the example of Enumeration Type Definition for m2m:enumFooType.

**Table 6.3.3.1-1: Example of oneM2M Enumeration Type Definition**

Value	Interpretation	Note
1	Interpretation-1	
2	Interpretation-2	
3	Interpretation-3	
See Clause x.x.x "title of clause"		

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"/>
    <xs:enumeration value="2"/>
    <xs:enumeration value="3"/>
  </xs:restriction>
</xs:simpleType>

```

**Figure 6.3.3.2.1-1: Example of XSD version of oneM2M Enumeration Type**

### 6.3.3.2. Enumeration Type Definitions

#### 6.3.3.2.1. m2m:resourceType

**Table 6.3.3.2.1-1: 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"		

#### 6.3.3.2.2. m2m:commaList

Used for *supportedResourceType* attribute of <CSEBase> resource.



**Table 6.3.3.2.2-1: 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"		

6.3.3.2.3. m2m:cseTypeID

Used for *cseType* attribute of <CSEBase> resource.

**Table 6.3.3.2.3-1: Interpretation of cseTypeID**

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

6.3.3.2.4. m2m:locationSource

Used for *locationSource* attribute of <locationPolicy> resource.

**Table 6.3.3.2.4-1: Interpretation of locationSource**

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

6.3.3.2.5. m2m:eventCatType

Used for *ec* parameter in request and *eventCat* attribute of <delivery> resource a5d cmdh policy resource types.

**Table 6.3.3.2.5-1: 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"		

### 6.3.3.2.6. m2m:operation

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

**Table 6.3.3.2.6-1: 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"		

### 6.3.3.2.7. m2m:responseType

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

**Table 6.3.3.2.7-1: Interpretation of responseType**

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

### 6.3.3.2.8. m2m:resultContent

Used for *rc* parameter in request.

**Table 6.3.3.2.8-1: 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"		

### 6.3.3.2.9. m2m:discResType

**Table 6.3.3.2.9-1: Interpretation of discResType**

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

#### 6.3.3.2.10. m2m:statusCode

**Table 6.3.3.2.10-1: Interpretation of statusCode**

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

#### 6.3.3.2.11. m2m:requestStatus

Used for *requestStatus* attribute in <request> resource.

**Table 6.3.3.2.11-1: Interpretation of requestStatus**

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

#### 6.3.3.2.12. m2m:operationResult

Used for *operationResult* attribute in <request> resource

**Table 6.3.3.2.12-1: Interpretation of operationResult**

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

#### 6.3.3.2.13. m2m:memberType

Used for *memberType* attribute in <member> resource.

**Table 6.3.3.2.13-1: 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 "Resource Type group"		

6.3.3.2.14. m2m:consistencyStrategy

Used for *consistencyStrategy* attribute in <group> resource.

**Table 6.3.3.2.14-1: Interpretation of consistencyStrategy**

Value	Interpretation	Note
1		
2		
3		
See Clause 7.3.12 "Resource Type group"		

6.3.3.2.15. m2m:cmdType

Used for *cmdType* attribute in <mgmtCmd> resource.

**Table 6.3.3.2.15-1: 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"		

6.3.3.2.16. m2m:execModeType

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

**Table 6.3.3.2.16-1: 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 execlnstance"		

6.3.3.2.17. m2m:execStateType

Used for *execStatusType* attribute in <execInstance> resource.

**Table 6.3.3.2.17-1: Interpretation of execStateType**

Value	Interpretation	Note
1	INITIATED	
2	STARTED	
3	FINISHED	
4	CANCELLING	
5	CANCELLED	
6	STATUS_NON_CANCELLABLE	
See Clause 7.3.16 "Resource Type execlnstance"		

6.3.3.2.18. m2m:execResultType

Used for *execStatusType* attribute in <execInstance> resource.

**Table 6.3.3.2.18-1: Interpretation of execResultType**

Value	Interpretation	Note
1	STATUS_REQUEST_UNSUPPORTED	
2	STATUS_REQUEST_DENIED	
3	STATUS_CANCELLATION_DENIED	
4	STATUS_INTERNAL_ERROR	
5	STATUS_INVALID_ARGUMENTS	
6	STATUS_RESOURCES_EXCEEDED	
7	STATUS_FILE_TRANSFER_FAILED	
8	STATUS_FILE_TRANSFER_SERVER_AUTHENTICATION_FAILURE	
9	STATUS_UNSUPPORTED_PROTOCOL	
10	STATUS_UPLOAD_FAILED	
11	STATUS_FILE_TRANSFER_FAILED_MULTICAST_GROUP_UNABLE_JOIN	
12	STATUS_FILE_TRANSFER_FAILED_SERVER_CONTACT_FAILED	
13	STATUS_FILE_TRANSFER_FAILED_FILE_ACCESS_FAILED	
14	STATUS_FILE_TRANSFER_FAILED_DOWNLOAD_INCOMPLETE	
15	STATUS_FILE_TRANSFER_FAILED_FILE_CORRUPTED	
16	STATUS_FILE_TRANSFER_FILE_AUTHENTICATION_FAILURE	
17	STATUS_FILE_TRANSFER_FAILED	
18	STATUS_FILE_TRANSFER_SERVER_AUTHENTICATION_FAILURE	
19	STATUS_FILE_TRANSFER_WINDOW_EXCEEDED	
20	STATUS_INVALID_UUID_FORMAT	
21	STATUS_UNKNOWN_EXECUTION_ENVIRONMENT	
22	STATUS_DISABLED_EXECUTION_ENVIRONMENT	
23	STATUS_EXECUTION_ENVIRONMENT_MISMATCH	
24	STATUS_DUPLICATE_DEPLOYMENT_UNIT	
25	STATUS_SYSTEM_RESOURCES_EXCEEDED	
26	STATUS_UNKNOWN_DEPLOYMENT_UNIT	
27	STATUS_INVALID_DEPLOYMENT_UNIT_STATE	
28	STATUS_INVALID_DEPLOYMENT_UNIT_UPDATE_DOWNGRADE_DISALLOWED	
29	STATUS_INVALID_DEPLOYMENT_UNIT_UPDATE_UPGRADE_DISALLOWED	
30	STATUS_INVALID_DEPLOYMENT_UNIT_UPDATE_VERSION_EXISTS	
See Clause 7.3.16 "Resource Type mgmtCmd"		

6.3.3.2.19. m2m:pendingNotification

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

**Table 6.3.3.2.19-1: Interpretation of pendingNotification**

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

6.3.3.2.20. m2m:notificationContentType

**Table 6.3.3.2.20-1: Interpretation of notificationContentType**

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

6.3.3.2.21. m2m:resourceStatus

This is used for eventNotificationCriteria.

**Table 6.3.3.2.21-1: Interpretation of resourceStatus**

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

6.3.3.2.22. m2m:operation

This is used for operationMonitor.

**Table 6.3.3.2.22-1: Interpretation of operation**

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

6.3.3.2.23. m2m:attribute

**Table 6.3.3.2.23-1: Interpretation of attribute**

Value	Interpretation	Note
1		
2		
3		
4		

6.3.3.2.24. m2m:status

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

**Table エラー! 指定したスタイルは使われていません。 -1: Interpretation of status**

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

6.3.3.2.25. m2m:batteryStatus

This is used for [battery] resource.

**Table 6.3.3.2.25-1: 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		

6.3.3.2.26. m2m:mgmtDefinition

This is used for mgmtObj resource.

**Table 6.3.3.2.26-1: 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		

6.3.3.2.27. m2m:logTypeId

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

**Table 6.3.3.2.27-1: Interpretation of logTypeId**

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

6.3.3.2.28. m2m:logStatus

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



**Table 6.3.3.2.28-1: 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.

6.3.3.2.29. m2m:listOfBoolean

Used to define the value space of the attribute "limitsDelAggregation" by imposing restrictions on the value space of xs:boolean.

**Table 6.3.3.2.29-1: Interpretation of listOfBoolean**

Value	Interpretation	Note
1	true	
2	false	
3	True and false	
See Clause D.1.5 "[cmdhLimits] Recourse"		

6.3.4. Complex Data Types

The present clause defines structured information for specific use in oneM2M protocol.

**Editor's Note:** explanation of table notations and example should be provided later.

6.3.4.1. m2m:filterCriteria

Used for *fc* (filter criteria) parameter in a request.

**Table 6.3.4.1-1: Type Definition of m2m:filterCriteria**

Element Path	Element Data Type	Multiplicity	Note
createdBefore	xs:dateTime		
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		

#### 6.3.4.2. m2m:eventCat

Used for *ec* parameter in request and eventCat attribute of <delivery> resource.

**Table 6.3.4.2-1: Type Definition of m2m:eventCat**

Element Path	Element Data Type	Multiplicity	Note
eventCatType	m2m:eventCatType		
eventCatNo	xs:nonNegativeInteger		

#### 6.3.4.3. m2m:deliveryMetaData

Used for *deliveryMetaData* attribute in <delivery> resource.

TBD

#### 6.3.4.4. m2m:aggregatedRequest

Used for *aggregatedRequest* attribute in <delivery> resource.

TBD

#### 6.3.4.5. m2m:metaInformation

Used for *metaInformation* attribute in <request> resource.

TBD

#### 6.3.4.6. m2m:content

Used for *cn* parameter in request/response and content attribute in <request> resource.

TBD

#### 6.3.4.7. m2m:batchNotify

Used for *batchNotify* attribute in <subscription> resource.

**Table 6.3.4.7-1: Type Definition of m2m:batchNotify**

Element Path	Element Data Type	Multiplicity	Note
number	xs:nonNegativeInteger		
duration	xs:duration		

#### 6.3.4.8. m2m:eventNotificationCriteria

Used for *eventNotificationCriteria* of a <subscription> resource.

**Table 6.3.4.8-1: Type Definition of m2m:eventNotificationCriteria**

Element Path	Element Data Type	Multiplicity	Note
createdBefore	xs:dateTime		
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		

#### 6.3.4.9. m2m:aggregatedNotification

**Table 6.3.4.9-1: Type Definition of m2m:aggregatedNotification**

Element Path	Element Data Type	Multiplicity	Note
notification	m2m:singleNotification	...	

#### 6.3.4.10. m2m:singleNotification

**Table 6.3.4.10-1: Type Definition of m2m:singleNotification**

Element Path	Element Data Type	Multiplicity	Note
notificationEvent	m2m:notificationEvent	...	
verificationRequest	xs:boolean		
subscriptionDeletion	xs:boolean		
subscriptionReference	xs:anyURI		
creator	m2m:id		
notificationForwardingURI	xs:anyURI		

#### 6.3.4.11. m2m:notificationEvent

Used for representation of resource at notification event.

**Table 6.3.4.11-1: Type Definition of m2m:notificationEvent**

Element Path	Element Data Type	Multiplicity	Note
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 for representation of resource at notification event.
resourceStatus	m2m:resourceStatus		
operationMonitor	m2m:operationMonitor		

#### 6.3.4.12. m2m:operationMonitor

**Table 6.3.4.12-1: Type Definition of m2m:operationMonitor**

Element Path	Element Data Type	Multiplicity	Note
operation	m2m:operation		
originator	m2m:id		

#### 6.3.4.13. m2m:externalID

**Table 6.3.4.13-1: Type Definition of m2m:externalID**

Element Path	Element Data Type	Multiplicity	Note
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].

#### 6.3.4.14. m2m:actionStatus

**Table 6.3.4.14-1: Type Definition of m2m:actionStatus**

Element Path	Element Data Type	Multiplicity	Note
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

#### 6.3.4.15. m2m:responseStatus

**Table 6.3.4.15-1: Type Definition of m2m:responseStatus**

Element Path	Element Data Type	Multiplicity	Note
code	m2m:statusCode		
description	xs:string		

#### 6.3.4.16. m2m:anyArgType

**Table 6.3.4.16-1: Type Definition of m2m:**

Element Path	Element Data Type	Multiplicity	Note
name	xs:string		
type	xs:anyType		

#### 6.3.4.17. m2m:anyArgListType

**Table 6.3.4.17-1: Type Definition of m2m:anyArgListType**

Element Path	Element Data Type	Multiplicity	Note
anyArg(TBD)	m2m:anyArgType	...	

*Editors's Note: original spec was anonymous structure. notation for anonymous structure is FFS.*

#### 6.3.4.18. m2m:resetArgsType

**Table 6.3.4.18-1: Type Definition of m2m:resetArgsType**

Element Path	Element Data Type	Multiplicity	Note
anyArg	m2m:anyArgType		

#### 6.3.4.19. m2m:rebootArgsType

**Table 6.3.4.19-1: Type Definition of m2m:rebootArgsType**

Element Path	Element Data Type	Multiplicity	Note
anyArg	m2m:anyArgType		

#### 6.3.4.20. m2m:uploadArgsTypes

**Table 6.3.4.20-1: Type Definition of m2m:uploadArgsType**

Element Path	Element Data Type	Multiplicity	Note
fileType	xs:string		
URL	xs:anyURI		
username	xs:string		
password	xs:string		

#### 6.3.4.21. m2m:downloadArgsType

**Table 6.3.4.21-1: Type Definition of m2m:downloadArgsType**

Element Path	Element Data Type	Multiplicity	Note
fileType	xs:string		
URL	xs:anyURI		
username	xs:string		
password	xs:string		
filesize	xs:positiveInteger		
targetFile	xs:string		
delaySeconds	xs:positiveInteger		
successURL	xs:anyURI		
startTime	xs:dateTime		
completeTime	xs:dateTime		

#### 6.3.4.22. m2m:softwareInstallArgsType

**Table 6.3.4.22-1: Type Definition of m2m:softwareInstallArgsType**

Element Path	Element Data Type	Multiplicity	Note
URL	xs:anyURI		
UUID	xs:string		
username	xs:string		
password	xs:string		
executionEnvRef	xs:string		

#### 6.3.4.23. m2m:softwareUpdateArgsType

**Table 6.3.4.23-1: Type Definition of m2m:softwareUpdateArgsType**

Element Path	Element Data Type	Multiplicity	Note
UUID	xs:string		
version	xs:string		
URL	xs:anyURI		
username	xs:string		
password	xs:string		
executionEnvRef	xs:string		

#### 6.3.4.24. m2m:softwareUninstallArgsType

**Table 6.3.4.24-1: Type Definition of m2m:softwareUninstallArgsType**

Element Path	Element Data Type	Multiplicity	Note
UUID	xs:string		
version	xs:string		
executionEnvRef	xs:string		

#### 6.3.4.25. m2m:execReqArgsType

**Table 6.3.4.25-1: Type Definition of m2m:execReqArgsType**

Element Path	Element Data Type	Multiplicity	Note
reset	m2m:resetArgsType		
reboot	m2m:rebootArgsType		
upload	m2m:downloadArgsType		
download	m2m:downloadArgsType		
softwareInstall	m2m:softwareInstallArgsType		
softwareUpdate	m2m:softwareUpdateType		
softwareUninstall	m2m:softwareUninstallArgsType		
anyArg	m2m:anyArgListType		

#### 6.3.4.26. m2m:execReqArgsListType

**Table 6.3.4.26-1: Type Definition of m2m:execReqArgsListType**

Element Path	Element Data Type	Multiplicity	Note
execReqArgs	m2m:execReqArgsType	...	

*Editor's Note: the lists in the table above (execReqArgsListType, anyArgListType) need to be reformatted to include the multiplicity.*

#### 6.3.4.27. m2m:mgmtLinkRef

**Table 6.3.4.27-1: Type Definition of m2m:mgmtLinkRef**

Element Path	Element Data Type	Multiplicity	Note
(base content)	xs:anyURI		URI (of type xs:anyURI) with name and type attributes.
name (XML attribute)	xs:string		The name attribute represents the name of the referenced resource instance.
type (XML attribute)	m2m:mgmtDefinition		The type attribute is restricted to the allowed specializations of resource type <mgmtObj>

*Editor's Note: notation for XML attribute is FFS.*

#### 6.3.4.28. m2m:setOfAcrs

**Table 6.3.4.28-1: Type Definition of m2m:setOfAcrs**

Element Path	Element Data Type	Multiplicity	Note
accessControlRules	anonymous		Data type of privileges and selfPrivileges attributes
accessControlOriginators	anonymous		
accessControlOperations	anonymous		
accessControlContexts	anonymous		

*Editor's Note: definition of anonymous data type is FFS.*

#### 6.3.4.29. m2m:

**Table 6.3.4.29-1: Type Definition of m2m:**

Element Path	Element Data Type	Multiplicity	Note

#### 6.3.4.30. m2m:

**Table 6.3.4.30-1: Type Definition of m2m:**

Element Path	Element Data Type	Multiplicity	Note

### 6.3.5. 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 6.3.5-1.

**Table 6.3.5-1: 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 <b>defaultDelAggregation</b> 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 <b>stateTag</b> attribute)
stateTagBigger	xs:integer		(Same type as common <b>stateTag</b> attribute)
expireBefore	xs:dateTime		
expireAfter	xs:dateTime		
Labels	m2m:labels		
resourceType	m2m:resourceType		
sizeAbove	xs:int		(Same type as <b>contentType</b> attribute of the <contentInstance> resource)
sizeBelow	xs:int		(Same type as <b>contentType</b> 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 <ul style="list-style-type: none"> <li>• native oneM2M codes</li> <li>• mapped security codes</li> <li>• mapped device management codes</li> <li>• mapped transport codes</li> <li>• mapped interworking codes</li> <li>• mapped vendor extensions</li> </ul>	2000 - bbbb
	Reserved for future <ul style="list-style-type: none"> <li>• native oneM2M codes</li> <li>• mapped security codes</li> <li>• mapped device management codes</li> <li>• mapped transport codes</li> <li>• mapped Interworking codes</li> <li>• mapped vendor extensions</li> </ul>	bbbb+1 - 7999
Acknowledgements	Current <ul style="list-style-type: none"> <li>• native oneM2M codes</li> <li>• mapped security codes</li> <li>• mapped device management codes</li> <li>• mapped transport codes</li> <li>• mapped interworking codes</li> <li>• mapped vendor extensions</li> </ul>	8000 - cccc
	Reserved for future <ul style="list-style-type: none"> <li>• native oneM2M codes</li> <li>• mapped security codes</li> <li>• mapped device management codes</li> <li>• mapped transport codes</li> <li>• mapped interworking codes</li> <li>• mapped vendor extensions</li> </ul>	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		
access denied			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.

This MIME-Type is also mandated to specify the parameter 'ty' with the resource type id of the resource.

The content of this MIME-Type shall be XML serialized formed M2M resource.

Ex) Content-Type: application/vnd.onem2m-resource-data+xml; ty=3

See clause 6.3.4 Complex Data Types for XSD definition of m2m:resourceData

### 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 6.3.4 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 6.3.4 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 parameters/attributes 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)**

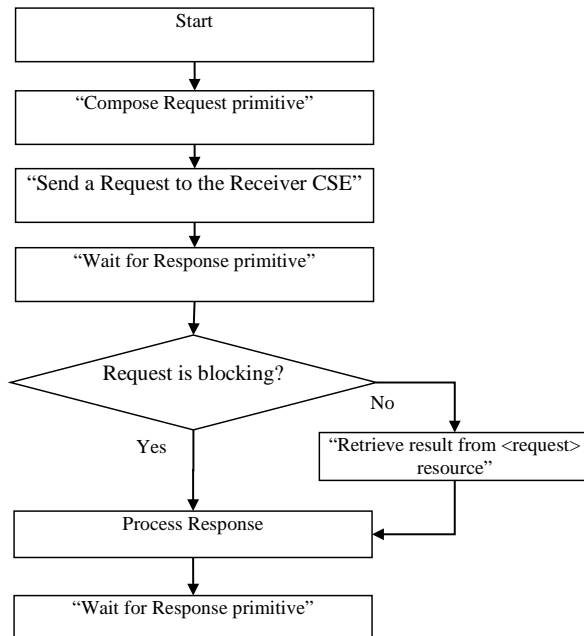
Primitive parameter	Successful DELETE		Successful NOTIFY		Unsuccessful I	
	M/O		M/O		M/O	
	Mca	Mcc	Mca	Mcc	Mca	Mcc
primitiveType (=RESPONSE)	M	M	M	M	M	M
Response Status Code	M	M	M	M	M	M
Request Identifier	M	M	M	M	M	M
Content	C	C	NP	NP	C	C
To	C	C	C	C	C	C
From	C	C	C	C	C	C
Originating Timestamp	O	O	O	O	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.

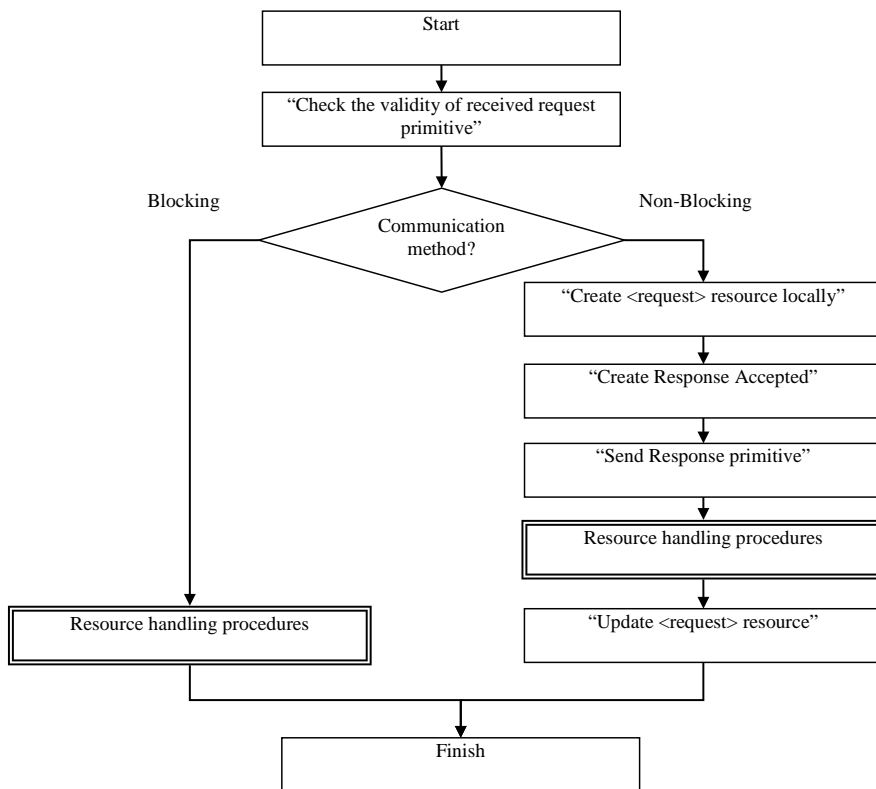
Orig-4.0 "Request is blocking?": This step shall be operated after getting the Response primitive from step Oring-3.0 "Wait for Response primitive". In this step, the Originator checks whether the request was blocking. If the request was blocking, it goes to step Orig-6.0 "Process Response" (TRUE branch). 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 in clause 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 7.2.2.4 for details.

Recv-4.0 “Create Response Accepted”: Please refer to clause 7.2.2.2 for details.

Recv-5.0 “Send Response Primitive”: Please refer to clause 7.2.2.2 for details.

Recv-6.0 “Resource handling procedure”: Please refer to Figure 7.2.1.2.2-2 for details.

Recv-7.0 “Update <request> resource”: Please refer to clause 7.2.2.7 for details. This step is only valid when the request is non-blocking.

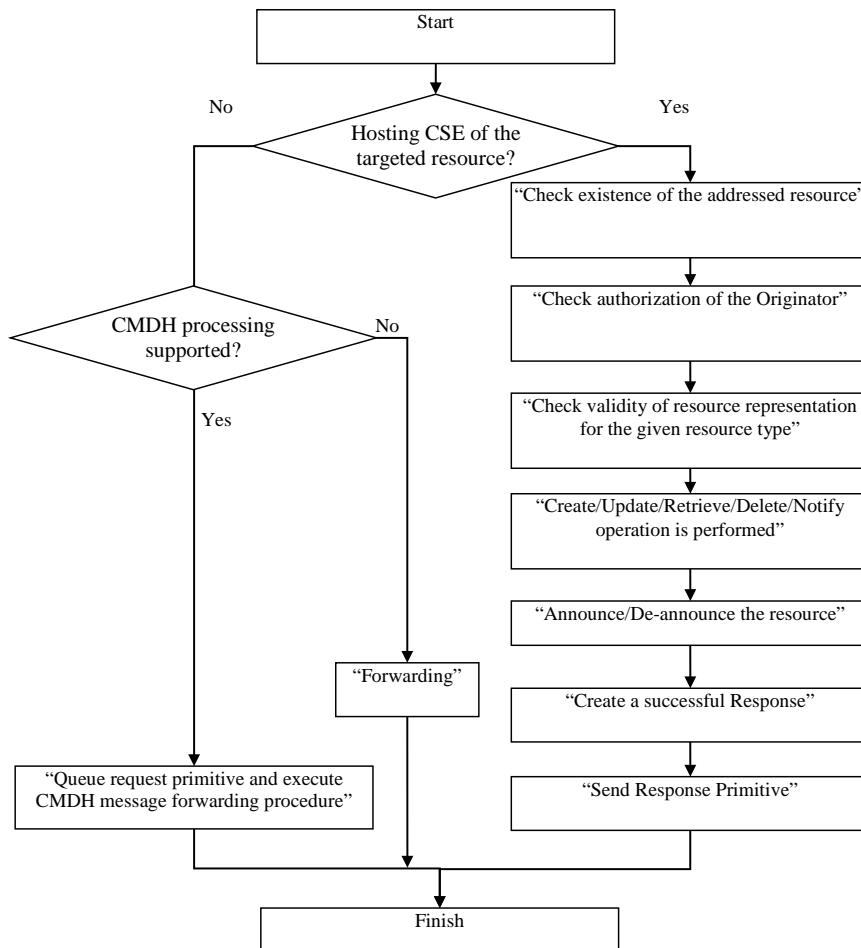


Figure 7.2.1.2.2-2: Resource handling procedure

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

### 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.

**Editor’s Note:** Is this “to” parameter ?

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.

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

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 parameters 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. Receiver 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 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 <b>rqet</b> , <b>rset</b> , <b>oet</b> and <b>rp</b> parameters associated with the original request. If the value consistent with the <b>rqet</b> , <b>rset</b> , <b>oet</b> and <b>rp</b> 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 ' <b>privileges</b> ' attribute <ul style="list-style-type: none"> <li>Allow RUD operations to the Hosting CSE</li> <li>Allow RD operations to the Originator, i.e. the value of the parameter <b>fr</b> in the associated non-blocking request</li> </ul> In the ' <b>selfPrivileges</b> ' attribute <ol style="list-style-type: none"> <li>Allow U operations the parent &lt;accessControlPolicy&gt; resource to the Originator, i.e. the value of the parameter <b>fr</b> in the associated non-blocking request</li> </ol>
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 according 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 <b>op</b> in the associated non-blocking request.
target	The value of the parameter <b>to</b> in the associated non-blocking request.
originator	The value of the parameter <b>fr</b> in the associated non-blocking request.
requestID	The value of the parameter <b>ri</b> 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 <b>cn</b> , 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 (Acknowledgement)

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 ( <b>op</b> )	The value of the parameter <b>to</b> in the associated non-blocking request.
statusResponseCode	"Locally accepted" – 800x
requestIdentifier ( <b>ri</b> )	The value of the parameter <b>ri</b> in the associated non-blocking request.
originatingTimestamp ( <b>ot</b> )	Timestamp when this message was built
Content ( <b>cn</b> )	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 <i>rc</i> 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.

**Editor's Note: both statements should be corrected as "to" parameter .**

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".

**Editor's Note: attribute -> parameter ?**

#### 7.2.2.3.2. Check validity of resource representation for CREATE

The handling below shall apply 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 request. The 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 with the 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 accessControlOperations) 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
accessControlPolicy	CDT-accessControlPolicy-V1_0.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:setOfAcrs	
selfPrivileges	M	O	O	NP	m2m:setOfAcrs	

**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
remoteCSE	CDT-remoteCSE-v1_0_0.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		
accessControlPolicyIDs	O	O	O	NP		accessControlPolicyIDs
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 7.3.3.1-3: Resource specific attributes on <remoteCSE>**

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	m2m:pOAList	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: 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. <remoteCSE> Resource Specific Procedure on 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
container	CDT-container-V1_0_0.xsd	

**Table 7.3.5.1-2: Common Attributes on <container>**

Attribute Name	Request Optionality				Default Value	Resource Specific Note
	C	R	U	D		
<i>resourceType</i>	NP	O	NP	NP		
<i>resourceID</i>	NP	O	NP	NP		
<i>parentID</i>	NP	O	NP	NP		
<i>accessControlPolicyIDs</i>	O	O	O	NP		
<i>creationTime</i>	NP	O	NP	NP		
<i>expirationTime</i>	O	O	O	NP		
<i>lastModifiedTime</i>	NP	O	NP	NP		
<i>stateTag</i>	NP	O	NP	NP		
<i>labels</i>	O	O	NP	NP		
<i>announceTo</i>	O	O	O	NP		
<i>announcedAttribute</i>	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		
<i>creator</i>	NP	O	NP	NP	m2m:id	
<i>maxNrOfInstances</i>	O	O	O	NP	xs:nonNegativeInteger	
<i>maxByteSize</i>	O	O	O	NP	xs:nonNegativeInteger	
<i>maxInstanceAge</i>	O	O	O	NP	xs:nonNegativeInteger	
<i>currentNrOfInstances</i>	NP	O	NP	NP	xs:nonNegativeInteger	
<i>currentByteSize</i>	NP	O	NP	NP	xs:nonNegativeInteger	
<i>latest</i>	NP	O	NP	NP	xs:string	
<i>locationID</i>	O	O	O	NP	xs:anyURI	
<i>ontologyRef</i>	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

### 7.3.5.2. <container> Resource Specific Procedure on CRUD Operations

This clause describes container resource specific behaviour for CRUD operations.

#### 7.3.5.2.1. 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.

#### 7.3.5.2.2. 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.

#### 7.3.5.2.3. 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.

#### 7.3.5.2.4. 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

### 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
contentInstance	CDT-contentInstance-v1_0_0.xsd	

**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		
creationTime	NP	O	NP	NP		
lastModifiedTime	NP	O	NP	NP		
stateTag	NP	O	NP	NP		
labels	O	O	NP	NP		
announceTo	O	O	NP	NP		
announcedAttribute	NP	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

#### 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

*Originator:*

The <contentInstance> resource shall not be Updated 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".

#### 7.3.6.2.4. 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
subscription	CDT-subscription-v1_0_0.xsd	

**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. <subscription> Resource Specific Procedure on 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:*

No change from the generic procedures in clause 7.2.1.2.2.

#### 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 <scheduleAnn>

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

*Originator:*

No change from the generic procedures in clause 7.2.1.2.1.

If <schedule> is created then *scheduleElement* (L) shall be created.

*Receiver:*

No change from the generic procedures in clause 7.2.1.2.2.

#### 7.3.8.2.2. 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.

#### 7.3.8.2.3. 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.

#### 7.3.8.2.4. Delete

*Originator:*

No change from the generic procedures in clause 7.2.1.2.1.

If <schedule> is deleted then *scheduleElement* (L) shall be deleted.

*Receiver:*

No change from the generic procedures in clause 7.2.1.2.2.



## 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 <locationPolicy> 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) Rcv-6.6
- 8) Rcv-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	<a href="#">CDT-request-v1_0_0-&lt;&lt;date of published&gt;&gt;.xsd</a>	

**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 Hosng 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 attribute 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 representing 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 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".



#### 7.3.11.2.4. 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 a *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.

**Editor's Note:** primitive 'attribute' should be replaced with 'parameter' ?

- 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 to 7.3.12.2.1

- 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 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) 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.

The procedures are defined for management when external management technologies are used. When service layer management are performed, generic procedures defined in 7.2.1.2 shall comply for resource creation, update, retrieval and deletion. Procedures additional to resource manipulations to perform the management are further defined in Annex D.

### 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 エラー! 参照元が見つかりません。 and table エラー! 参照元が見つかりません。 . 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
mgmtCmd	CDT-mgmtCmd-v1_0_0.xsd	

**Table 7.3.15.1-2: Common Attributes on <mgmtCmd>**

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	NP	NP		
creationTime	NP	O	NP	NP		
expirationTime	O	O	O	NP		
lastModifiedTime	NP	O	NP	NP		
labels	O	O	O	NP		

**Table 7.3.15.1-3: Resource Specific Attributes on <mgmtCmd>**

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
<i>description</i>	O	O	O	NP	xs:string	size: 256 No default
<i>cmdType</i>	M	O	O	NP	m2m:cmdType	RESET, REBOOT, UPLOAD, DOWNLOAD, SOFTWAREINSTALL, SOFTWAREUPDATE, SOFTWAREUNINSTALL No default
<i>execReqType</i>	O	O	O	NP	m2m:execReqArgs ListType	A list of entries which are dependent on cmdType: If cmdType=RESET, execReqArgsList=resetArgsType. If cmdType=REBOOT, execReqArgsList=rebootArgsType. If cmdType=UPLOAD, execReqArgsList=uploadArgsType. If cmdType=DOWNLOAD, execReqArgsList=downloadArgsType. If cmdType=SOFTWAREINSTALL, execReqArgsList=softwareInstallArgsType. If cmdType=SOFTWAREUPDATE, execReqArgsList=softwareUpdateArgsType. If cmdType=SOFTWAREUNINSTALL, execReqArgsList=softwareUninstallArgsType. No default
<i>execEnable</i>	O	O	O	NP	xs:anyURI	No default
<i>execTarget</i>	M	O	O	NP	m2m:nodeID	No default
<i>execMode</i>	M	O	O	NP	m2m:execModeType	IMMEDIATEONCE, IMMEDIATE REPEAT, RANDOMONCE, RANDOMREPEAT Default=IMMEDIATEONCE
<i>execFrequency</i>	O	O	O	NP	xs:duration	No default
<i>execDelay</i>	O	O	O	NP	xs:duration	Default=0
<i>execNumber</i>	O	O	O	NP	xs:nonNegativeInteger	Default=1

**Table 7.3.15.1-4: Child resources of <mgmtCmd>**

Child Resource Type	Child Resource Name	Multiplicity	Ref. to Resource Type Definition
<subscription>	[variable]	0..n	clause 7.3.7
<execInstance>	[variable]	1	clause 7.3.16

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 `IMMEDIATE REPEAT`, `<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 `RANDOM ONCE`, `<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 `RANDOM REPEAT`, `<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 ***resourceType***, ***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 unsuccessful 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:nodelD	Not applicable		
execMode	m2m:execModeType	IMMEDIATEONCE	IMMEDIATEONCE, IMMEDIATE REPEAT, 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 there 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
node	CDT-node-v1_0_0.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	NP	NP		
resourceID	NP	O	NP	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	m2m:nodeID	
hostedCSEID	O	O	NP	NP	m2m:id	

**Table 7.3.17.1-4: Child resources of <node>**

Child Resource Type	Child Resource Name	Multiplicity	Ref. to Resource Type Definition
<mgmtObj>	[variable]	0..n	7.3.14, Annex D
<subscription>	[variable]	0..n	7.3.7

### 7.3.17.2. Resource Specific Procedure on 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	Name	Multiplicity	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
statsConfig	CDT-statsConfig-v1_0_0.xsd	

**Table 7.3.22.1-2: Common Attributes on <statsConfig>**

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 7.3.22.1-3 Resource specific attributes on <statsConfig>**

Attribute Name	Request Optionality				Data Type	Default Value and Constraints
	C	R	U	D		
creator	NP	O	NP	NP	m2m:ID	

**Table 7.3.22.1-4: Child resources of <statsConfig>**

Child Resource Type	Child Resource Name	Multiplicity	Ref. to Resource Type Definition
<eventConfig>	[variable]	0..n	7.3.23
<subscription>	[variable]	0..n	7.3.7

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

### 7.3.22.2.1. Create

*Originator:*

No change from the generic procedure in clause 7.2.1.2.1

*Receiver:*

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

Resource-specific operation before Recv-6.2:

- 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

*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.

### 7.3.22.2.3. Update

*Originator:*

No change from the generic procedures in clause 7.2.1.2.1.

*Receiver:*

### 7.3.22.2.4. 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..

## 7.3.23. 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
<i>Actual Data Type ID</i>	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	<i>CDT-<a href="#">&lt;&lt;resource name&gt;&gt;-v1_0-<a href="#">&lt;&lt;date of published&gt;&gt;.xsd</a></a></i>	

**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.

If *ec* parameter is set to 'latest' in the notification for subscription, the transit CSE as well as Originator caches the most recent Notify request. That is, if a new Notify request is received by the CSE with a subscription reference that has already been buffered for a pending Notify request, the newer Notify request will replace the buffered older Notify request.

**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 2.1 to 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 *ec* parameter 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 and it shall have the *ec* set to 'latest'. 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..**

#### 7.4.2.5. Notification for subscription via group

Whenever the subscribed to resources' modification causes a notification sending procedure indicated in clauses 7.4.2.1 and the subscription relationship is established through group resource, the following procedure shall be performed for the notification sending.

The **Member hosting CSE** shall perform the steps defined in 7.4.2.1.

The **Group hosting CSE** shall perform the following steps in order:

Validate if the notification is sent from its own member resources when it gets a notification at the notificationURI. The group hosting CSE shall respond a response with response status code “access denied” if the validation is not successful.

Upon successful validation, the group hosting CSE shall collect the notifications to the same subscriber according to the *notificationForwardingURI* of each notification. The group hosting CSE shall aggregate them into aggregatedNotification which shall be included in notification. The timing of aggregation is done as per the group hosting CSE's local policy which is out of scope.

Send the aggregated notification to the *notificationURI* according to the *notificationForwardingURI* in the notification. In the case the group hosting CSE is the member of another group hosting CSE through which the subscription is created, the notification shall be sent according to the mapping of the *notificationURI* of the two group hosting CSEs. When aggregating the notifications, the group hosting CSE may utilize the *rqet* in the notification to determine the time the aggregated notifications to be sent.

“Wait for Response primitive”.

Upon receiving the response, the group hosting CSE shall send the response separately to each individual member hosting CSEs to respond their corresponding notify request.

The group hosting CSE may also stop aggregating the notifications depending on its own policy. The group hosting CSE shall not stop aggregating the notifications before the expirationTime of the corresponding subscription expires.

The **Subscriber** shall perform the following steps in order:

- 1) Extract each notification from the aggregated notification;
- 2) Treat the notification as it is sent from the original subscribed-to resource;
- 3) “Create a successful response”
- 4) “Send the Response primitive”

---

## 8 oneM2M Resource Types

### 8.1. Introduction

(TBD)

### 8.2. Short Names

#### 8.2.1. Introduction

XML and JSON representations require the explicit encoding of the names of primitive parameters, resource attributes, (in the case of XML) resource types and complex data types members. Whenever a protocol binding transfers such a name over a oneM2M reference point, it shall use a shortened form of that name, rather than the full name that is used elsewhere in this and other oneM2M specifications. Short names enable payload reduction on involved telecommunication interfaces.

The mapping between the full names and their shortened form is given in the clauses that follow.

#### 8.2.2. Primitive parameters

In protocol bindings primitive parameter names shall be translated into short names of Table 8.2.2-1.

**Table 8.2.2-1: Primitive parameter short names**

Parameter Name	Occurs in	Short Name
<i>Primitive Type</i>	Request, Response	pt
<i>Operation</i>	Request, Response	op
<i>To</i>	Request, Response	to
<i>From</i>	Request, Response	fr
<i>Request Identifier</i>	Request, Response	rqi
<i>Resource Type</i>	Request	ty
<i>Name</i>	Request	nm
<i>Content</i>	Request, Response	cn
<i>Originating Timestamp</i>	Response	ot
<i>Request Expiration Timestamp</i>	Request	rget
<i>Result Expiration Timestamp</i>	Response	rset
<i>Operation Execution Time</i>	Request	oet
<i>Response Type</i>	Request	rt
<i>Result Persistence</i>	Request	rp
<i>Result Content</i>	Request	rsc
<i>Event Category</i>	Request	ec
<i>Delivery Aggregation</i>	Request	da
<i>Group Request Identifier</i>	Request	gid
<i>Filter Criteria</i>	Request	fc
<i>Discovery Result Type</i>	Request	drt
<i>Response Code</i>	Response	rc
<i>Status Code</i>	Response	sc
<i>Pending Requests</i>	Response	pr

### 8.2.3. Resource attributes

In protocol bindings resource attributes names shall be translated into short names of Table 8.3-1.

**Table: Resource attribute short names (1/5)**

Attribute Name	Occurs in	Short Name
accessControlPolicyIDs	All except management resources from firmware	acpi
announcedAttribute	All except delivery, request, management resources from firmware	aa
announceTo	All except delivery, request, management resources from firmware	at
creationTime	All except management resources from firmware	ct
expirationTime	All except management resources from firmware	et
lastModifiedTime	All except management resources from firmware	lt
parentID	All except management resources from firmware	pi
resourceID	All except management resources from firmware	ri
stateTag	All except management resources from firmware	st
privileges	accessControlPolicy	pv
selfPrivileges	accessControlPolicy	pvs
App-ID	AE	api
AE-ID	AE	aei
pointOfAccess	AE, CSEBase, remoteCSE	pa
ontologyRef	AE, container, contentInstance	or
nodeLink	container, CSEBase, remoteCSE	nl
creator	container, eventConfig, group, pollingChannel, statsConfig, subscription	cr
maxNrOfInstances	container	mni
maxByteSize	container	mbs
maxInstanceAge	container	mia
currentNrOfInstances	container	cni
currentByteSize	container	cbs
latest	container	la
locationID	container	li
typeOfContent	contentInstance	toc
contentSize	contentInstance	cs
cseType	CSEBase, remoteCSE	cst
CSE-ID	CSEBase, remoteCSE	csi
supportedResourceType	CSEBase	srt
notificationCongestionPolicy	CSEBase	ncp
source	delivery	sr
target	delivery	tg
lifespan	delivery	ls
eventCat	delivery	ec*
deliveryMetaData	delivery	dmd
aggregatedRequest	delivery	arq
eventID	eventConfig	evi
eventType	eventConfig	evt
privileges	accessControlPolicy	pv
selfPrivileges	accessControlPolicy	pvs
App-ID	AE	api
AE-ID	AE	aei
pointOfAccess	AE, CSEBase, remoteCSE	pa
ontologyRef	AE, container, contentInstance	or
nodeLink	container, CSEBase, remoteCSE	nl
creator	container, eventConfig, group, pollingChannel, statsConfig, subscription	cr
maxNrOfInstances	container	mni
maxByteSize	container	mbs
maxInstanceAge	container	mia
currentNrOfInstances	container	cni



**Table: Resource attribute short names (2/5)**

Attribute Name	Occurs in	Short Name
currentByteSize	container	cbs
latest	container	la
locationID	container	li
typeOfContent	contentInstance	toc
contentSize	contentInstance	cs
cseType	CSEBase, remoteCSE	cst
CSE-ID	CSEBase, remoteCSE	csi
supportedResourceType	CSEBase	srt
notificationCongestionPolicy	CSEBase	ncp
source	delivery	sr
target	delivery	tg
lifespan	delivery	ls
eventCat	delivery	ec*
deliveryMetaData	delivery	dmd
aggregatedRequest	delivery	arq
eventID	eventConfig	evi
eventType	eventConfig	evt
evenStart	eventConfig	evs
eventEnd	eventConfig	eve
transactionType	eventConfig	tt
dataSize	eventConfig	ds
execStatus	execInstance	exs
execResult	execInstance	exr
execDisable	execInstance	exd
execTarget	execInstance, mgmtCmd	ext
execMode	execInstance, mgmtCmd	exm
execFrequency	execInstance, mgmtCmd	exf
execDelay	execInstance, mgmtCmd	exy
execNumber	execInstance, mgmtCmd	exn
execReqArgs	execInstance, mgmtCmd	extra
execEnable	mgmtCmd	exe
memberType	group	mt
currentNrOfMembers	group	cnm
maxNrOfMembers	group	mnm
membersList	group	ml
membersAccessControlPolicyIDs	group	macpi
memberTypeValidated	group	mtv
consistencyStrategy	group	csy
groupName	group	gn
locationSource	locationPolicy	los
locationUpdatePeriod	locationPolicy	lou
locationTargetId	locationPolicy	lot
locationServer	locationPolicy	lor
locationContainerID	locationPolicy	loi
locationContainerName	locationPolicy	lon
serviceRoles	m2mServiceSubscriptionProfile	svr
description	mgmtCmd, mgmtObj, all management resources from firmware	dc
cmdType	mgmtCmd	cmt
mgmtDefinition	mgmtObj, all management resources from firmware	mgd
objectIDs	mgmtObj	obis

**Table: Resource attribute short names (3/5)**

Attribute Name	Occurs in	Short Name
objectPaths	mgmtObj	obps
objectAttribute	mgmtObj	oba
nodeID	node	ni
hostedCSEID	node	hci
CSEBase	remoteCSE	cb
M2M-Ext-ID	remoteCSE	mei
Trigger-Recipient-ID	remoteCSE	tri
requestReachability	remoteCSE	rr
originator	request	og
metaInformation	request	mi
requestStatus	request	rs
operationResult	request	ol
scheduleElement	schedule	se
deviceIdIdentifier	serviceSubscribedNode	di
statsCollectID	statsCollect	sci
collectingEntityID	statsCollect	cei
collectedEntityID	statsCollect	cdi
status	statsCollect, areaNwkDeviceInfo	ss
statModel	statsCollect	sm
collectPeriod	statsCollect	cp
eventNotificationCriteria	subscription	enc
expirationCounter	subscription	exc
notificationURI	subscription	nu
notificationForwardingURI	subscription	nfu
batchNotify	subscription	bn
rateLimit	subscription	rl
preSubscriptionNotify	subscription	psn
pendingNotification	subscription	pn
notificationSprivitoragePriority	subscription	nsp
latestNotify	subscription	ln
notificationContentType	subscription	nct
notificationEventCat	subscription	nec
subscriberURI	subscription	su
objectID	All management resources from firmware	obi
objectPath	All management resources from firmware	obp
version	firmware, software	vr
URL	firmware, software	ur
update	firmware	ud
updateStatus	firmware	uds
install	software	in
uninstall	software	un
installStatus	software	ins
activate	software	act
objectPaths	mgmtObj	obps
objectAttribute	mgmtObj	oba
nodeID	node	ni
deactivate	software	dea
activateStatus	Software, areaNwkInfo	acts
memAvailable	memory	mma
memTotal	memory	mmt

**Table: Resource attribute short names (4/5)**

Attribute Name	Occurs in	Short Name
areaNwkType	areaNwkInfo	ant
listOfDevices	areaNwkInfo	ldv
devId	areaNwkDeviceInfo	dvd
devType	areaNwkDeviceInfo	dvt
areaNwkId	areaNwkDeviceInfo	awi
sleepInterval	areaNwkDeviceInfo	sli
sleepDuration	areaNwkDeviceInfo	sld
listOfNeighbors	areaNwkDeviceInfo	lnh
batteryLevel	battery	btl
batteryStatus	battery	bts
deviceLabel	deviceInfo	dlb
manufacturer	deviceInfo	man
model	deviceInfo	mod
deviceType	deviceInfo	dtv
fwVersion	deviceInfo	fwv
swVersion	deviceInfo	swv
hwVersion	deviceInfo	hwv
capabilityName	deviceCapability	can
attached	deviceCapability	att
capabilityActionStatus	deviceCapability	cas
enable	deviceCapability	ena
disable	deviceCapability	dis
reboot	reboot	rbo
factoryReset	reboot	far
logTypeId	eventLog	lgt
logData	eventLog	lgd
logStatus	eventLog	lgs
logStart	eventLog	lga
logStop	eventLog	lgo
name	cmdhPolicy	cmn
mgmtLink	cmdhPolicy, activeCmdhPolicy, cmdhDefaults, cmdhNetworkAccessRules, cmdhNwAccessRule	cmk
order	cmdhDefEcValue, cmdhLimits	od
defEcValue	cmdhDefEcValue	dev
requestOrigin	cmdhDefEcValue, cmdhLimits	ror
requestContext	cmdhDefEcValue, cmdhLimits	rct
requestContextNotification	cmdhDefEcValue, cmdhLimits	rcn
requestCharacteristics	cmdhDefEcValue, cmdhLimits	rch
applicableEventCategories	cmdhEcDefParamValues, cmdhNetworkAccessRules, cmdhBuffer	aec
defaultRequestExpTime	cmdhEcDefParamValues	dqet
defaultResultExpTime	cmdhEcDefParamValues	dset
defaultOpExecTime	cmdhEcDefParamValues	doet
defaultRespPersistence	cmdhEcDefParamValues	drp
defaultDelAggregation	cmdhEcDefParamValues	dda
limitsEventCategory	cmdhLimits	lec
limitsRequestExpTime	cmdhLimits	lqet
limitsResultExpTime	cmdhLimits	lset
limitsOpExecTime	cmdhLimits	loet
limitsRespPersistence	cmdhLimits	lrp
limitsDelAggregation	cmdhLimits	lda
targetNetwork	cmdhNwAccessRule	ttn

**Table: Resource attribute short names (5/5)**

<b>Attribute Name</b>	<b>Occurs in</b>	<b>Short Name</b>
minReqVolume	cmdhNwAccessRule	mrv
backOffParameters	cmdhNwAccessRule	bop
otherConditions	cmdhNwAccessRule	ohc
maxBufferSize	cmdhBuffer	mbfs
storagePriority	cmdhBuffer	sgp

Note: marked short names have been already assigned in primitive Table 8.2.2-1.

#### 8.2.4. Resource types

In protocol bindings resource type names shall be translated into short names of Table 8.2.4-1.

**Table 8.2.4-1: Resource type short names**

Resource Type Name	Short Name
accessControlPolicy	acp
AE	ae
container	cnt
contentInstance	cin
CSEBase	csb
delivery	dlv
eventConfig	evcg
execInstance	exin
fanOutPoint	fopt
group	grp
locationPolicy	lcp
m2mServiceSubscriptionProfile	mssp
mgmtCmd	mgc
mgmtObj	mgo
node	nod
pollingChannelURI	pcu
remoteCSE	csr
request	req
schedule	sch
serviceSubscribedNode	svsn
statsCollect	stcl
statsConfig	stcg
subscription	sub
firmware	fwr
software	swr
memory	mem
areaNwkInfo	ani
areaNwkDeviceInfo	andi
battery	bat
deviceInfo	dvi
deviceCapability	dvc
reboot	rbt
eventLog	evl
cmdhPolicy	cmp
activeCmdhPolicy	acmp
cmdhDefaults	cmdf
cmdhDefEcValue	cmdv
cmdhEcDefParamValues	cmpv
cmdhLimits	cml
cmdhNetworkAccessRules	cmnr
cmdhNwAccessRule	cmwr
cmdhBuffer	cmbf

## 8.2.5. Complex data types members

In protocol bindings complex data types member names shall be translated into short names of Table 8.2.5-1.

**Table 8.2.5-1: Complex data types members short names**

Parameter Name	Occurs in	Short Name
createdBefore	filterCriteria, eventNotificationCriteria	crb
createdAfter	filterCriteria, eventNotificationCriteria	cra
modifiedSince	filterCriteria, eventNotificationCriteria	ms
unmodifiedSince	filterCriteria, eventNotificationCriteria	us
stateTagSmaller	filterCriteria, eventNotificationCriteria	sts
stateTagBigger	filterCriteria, eventNotificationCriteria	stb
expireBefore	filterCriteria, eventNotificationCriteria	exb
expireAfter	filterCriteria, eventNotificationCriteria	exa
labels	filterCriteria, eventNotificationCriteria	lbl
resourceType	filterCriteria	rty
sizeAbove	filterCriteria, eventNotificationCriteria	sza
sizeBelow	filterCriteria, eventNotificationCriteria	szb
contentType	filterCriteria	cty
limit	filterCriteria	lim
attribute	filterCriteria, eventNotificationCriteria	atr
resourceStatus	eventNotificationCriteria, notificationEvent	rss
operationMonitor	eventNotificationCriteria, notificationEvent	om
filterUsage	filterCriteria	fu
eventCatType	eventCat	ect
eventCatNo	eventCat	ecn
number	batchNotify	num
duration	batchNotify	dur
singleNotification	aggregatedNotification	sgn
notificationEvent	singleNotification	nev
verificationRequest	singleNotification	vrq
subscriptionDeletion	singleNotification	sud
subscriptionReference	singleNotification	sur
creator	singleNotification	cr*
notificationForwardingURI	singleNotification	nfu*
operation	operationMonitor	opr
originator	operationMonitor	org
accessId	externalID	aci
MSISDN	externalID	msd
action	actionStatus	acn
status	actionStatus	sus
childResource	All except execInstance, announced resource, management resources from firmware	ch

NOTE: \* marked short names have been already assigned in attribute Table 8.2.3-1.

---

## 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 the following information in the table:

- Resource Type: 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					
<areaNwkInfo>	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>	logTypeld					
<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		AnyJRIList			
<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	rctx				
	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:nonNegative Integer	0		

---

## 3 Annex B(normative): 4 Device Triggering

### 5 B.1. Providing Device Triggering service by means of 3GPP 6 networks

#### 7 B.1.1. Introduction

8 3GPP Underlying Network has defined a dedicated interface for requesting device triggering. The normative references  
9 for applicable interfaces are as follows: 3GPP TS 23.682 [15]. The specification for the interface Tsp is described in  
10 3GPP TS 29.368 [16]. Tsp interface uses Diameter Base Protocol as specified in IETF RFC 3588 [13], in order to use  
11 such an interface the CSE shall act as a Diameter client as described in IETF RFC 6733 [14].

12 Editors Note: IETF RFC 3588 Reference needs to be checked to determine that it is current.

13 Before the CSE initiates the device triggering, the CSE and MTC-IWF shall execute the procedures once as specified in  
14 3GPP TS29.368 [16].

#### 15 B.1.2. Device Action Request command

16 When a CSE needs to issue a device triggering request to the MTC-IWF, the CSE shall send a Device-Action-Request  
17 (DAR) command (for detail, see TS 29.368 [16]). The following list provides the parameters mapping between the  
18 oneM2M and 3GPP.

19 Either External-Id or MSISDN: the CSE maps it to the M2M-External-ID, see clause 6.3.2.3.

20 SCS identifier: the CSE maps it to the CSE-ID, see clause 6.3.2.1.

21 Application Port Identifier: the CSE maps it to Trigger-Recipient-ID, see clause 6.2.

#### 22 B.1.3. Device Action Answer command

23 As a result of device triggering request to MTC-IWF, the CSE receives a Device-Action-Answer (DAA) command (for  
24 detail, see TS 29.368 [16]).

#### 25 B.1.4. Device Notification Request command

26 As a report of the result for device triggering delivery by 3GPP network, the CSE receives a Device-Notification-  
27 Request (DNR) command (for detail, see TS 29.368 [16]).

#### 28 B.1.5. Device Notification Answer command

29 As a result of device notification request to MTC-IWF, the CSE sends a Device-Notification-Answer (DNA) command  
30 (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, 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"

```



```

96         minOccurs="0" maxOccurs="unbounded" />
97     </xs:sequence>
98 </xs:extension>
99 </xs:complexContent>
100 </xs:complexType>
101 </xs:element>
102 </xs:schema>
103

```

NOTE: The XML Schema documents need to contain some legal information at the top. Please use the statements shown in this example for now - though 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)

```

109 <?xml version="1.0" encoding="UTF-8"?>
110 <m2m:container xmlns:m2m="http://www.onem2m.org/xml/protocols"
111     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
112     xsi:schemaLocation=
113     "http://www.onem2m.org/xml/protocols CDT-container-v1_0_0-20140609.xsd "
114     name="12xx">
115     <parentID>//IN-CSEID.m2m.myoperator.org/96734</parentID>
116     <accessControlPolicyIDs>//IN-CSEID.m2m.myoperator.org/93405</accessControlPolicyIDs>
117     <creationTime>2013-12-31T12:00:00</creationTime>
118     <expirationTime>2013-12-31T12:30:00</expirationTime>
119     <lastModifiedTime>2013-12-31T12:00:00</lastModifiedTime>
120     <stateTag>0</stateTag>
121     <labels>label1 label2</labels>
122
123     <maxNrOfInstances>5</maxNrOfInstances>
124     <maxByteSize>104857600</maxByteSize>
125     <maxInstanceAge>3600</maxInstanceAge>
126     <currentNrOfInstances>2</currentNrOfInstances>
127     <currentByteSize>6</currentByteSize>
128     <latest>//IN-CSEID.m2m.myoperator.org/96739</latest>
129     <locationID>//IN-CSEID.m2m.myoperator.org/1112</locationID>
130     <ontologyRef>http://tempuri.org/ontologies/xyz</ontologyRef>
131
132     <childResource name="instance1234" type="instance">//IN-CSEID/1722</childResource>
133     <childResource name="instance1235" type="instance">//IN-CSEID/34722</childResource>
134     <childResource name="1923" type="subscription">//IN-CSEID/2323</childResource>
135
136 </m2m:container>
137

```

138

139

## Annex D(Normative): <mgmtObj> Resource Specializations

140

### D.1. Introduction

141

142

143

144

145

146

147

The annex is to define the structure and procedure for each <mgmtObj> resource specializations. The following resource specializations shall be created on the IN-CSE when the management request is performed using external management protocols. The IN-CSE further interacts with the management server to perform management requests towards the managed entity. If the management request is performed solely over the M2M Service Layer, the following resources are created on the managed entity if the managed entity is equipped with a CSE. If the managed entities are non-oneM2M Nodes, the resources are created on the MN-CSE of the managed entity. The details can be referenced to TS0001[6].

148

### D.2. Resource [firmware]

149

#### D.2.1. Introduction

150

The detailed description can be found in clause D.2 of Architecture TS-0001 [6].

151

**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	

152

153

**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	

154

155

#### D.2.2. Resource Specific Procedure on CRUD Operations

156

157

158

When management is performed using external management technologies, procedures defined in 7.3.14.2 <mgmtObj> specific procedures shall be used. The following clauses define additional procedures besides generic procedure defined in 7.2.1.2.

159

##### D.2.2.1. Create

160

###### *Originator:*

161

No change from the generic procedures in clause 7.2.1.2.1.

162

###### *Receiver:*

163

Primitive specific step after generic procedure defined in clause 7.2.1.2.2.

164 May start to download the firmware image from the location indicated by attribute URL in the firmware resource.

#### 165 D.2.2.2.Update

166 **Originator:**

167 No change from the generic procedures in clause 7.2.1.2.1.

168 **Receiver:**

169 Primitive specific operation additional to Recv-6.5 “Create/Update/Retrieve/Delete/Notify operation is performed”:

170 When the attribute *update* of the firmware resource is updated to TRUE, use the downloaded firmware image to update  
171 the current using firmware. The Receiver may need to update the *fwVersion* attribute of the deviceInfo resource if  
172 needed.

#### 173 D.2.2.3.Retrieve

174 **Originator:**

175 No change from the generic procedures in clause 7.2.1.2.1.

176 **Receiver:**

177 No change from the generic procedures in clause 7.2.1.2.2.

#### 178 D.2.2.4.Delete

179 **Originator:**

180 No change from the generic procedures in clause 7.2.1.2.1.

181 **Receiver:**

182 Primitive specific step after generic procedure defined in clause 7.2.1.2.2:

183 Delete the downloaded firmware image locally.

### 184 D.3. Resource [software]

#### 185 D.3.1. Introduction

186 The detailed description can be found in clause D.3 of Architecture TS-0001 [6].

187 **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	

188

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	

190

### 191 D.3.2. Resource Specific Procedure on CRUD Operations

192 When management is performed using external management technologies, procedures defined in 7.3.14.2 <mgmtObj>  
 193 specific procedures shall be used. The following clauses define additional procedures besides generic procedure defined  
 194 in 7.2.1.2.

#### 195 D.3.2.1. Create

##### 196 **Originator:**

197 No change from the generic procedures in clause 7.2.1.2.1.

##### 198 **Receiver:**

199 No change from the generic procedures in clause 7.2.1.2.2.

200 May start to download the software package from the location indicated by attribute *URL* in the software resource.

#### 201 D.3.2.2. Update

##### 202 **Originator:**

203 No change from the generic procedures in clause 7.2.1.2.1.

##### 204 **Receiver:**

205 Primitive specific operation additional to Recv-6.5 "Create/Update/Retrieve/Delete/Notify operation is performed":

206 When the attribute *install* of the software resource is updated to TRUE, install the software package downloaded from  
 207 the address indicated by attribute *URL* of the software resource.

208 When the attribute *uninstall* of the software resource is updated to TRUE, uninstall the corresponding software of the  
 209 software resource.

210 When the attribute *activate* of the software resource is updated to TRUE, activate the corresponding software of the  
 211 software resource.

212 When the attribute *deactivate* of the software resource is updated to TRUE, deactivate the corresponding software of the  
 213 software resource.

214 The Receiver may need to update the *swVersion* attribute of the deviceInfo resource if needed.

215

216 D.3.2.3.Retrieve

217 **Originator:**

218 No change from the generic procedures in clause 7.2.1.2.1.

219 **Receiver:**

220 No change from the generic procedures in clause 7.2.1.2.2.

221 D.3.2.4.Delete

222 **Originator:**

223 No change from the generic procedures in clause 7.2.1.2.1.

224 **Receiver:**

225 Primitive specific step after generic procedure defined in clause 7.2.1.2.2.

226 Delete the downloaded software package locally.

227

228 D.4. Resource [memory]

229 D.4.1. Introduction

230 The detailed description can be found in clause D.4 of Architecture TS-0001 [6].

231

**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	

232

233

**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.

234

235 D.4.2. Resource Specific Procedure on CRUD Operations

236 When management is performed using external management technologies, procedures defined in 7.3.14.2 <mgmtObj>  
237 specific procedures shall be used. The following clauses define additional procedures besides generic procedure defined  
238 in 7.2.1.2.

239 D.4.2.1.Create

240 **.Originator:**

241 No change from the generic procedures in clause 7.2.1.2.1.

242 **Receiver:**  
 243 No change from the generic procedures in clause 7.2.1.2.2.

244 **D.4.2.2.Update**

245 **Originator:**  
 246 No change from the generic procedures in clause 7.2.1.2.1.

247 **Receiver:**  
 248 No change from the generic procedures in clause 7.2.1.2.2.

249 **D.4.2.3.Retrieve**

250 **Originator:**  
 251 No change from the generic procedures in clause 7.2.1.2.1.

252 **Receiver:**  
 253 No change from the generic procedures in clause 7.2.1.2.2.

254 **D.4.2.4.Delete**

255 **Originator:**  
 256 No change from the generic procedures in clause 7.2.1.2.1.

257 **Receiver:**  
 258 No change from the generic procedures in clause 7.2.1.2.2.

259 **D.5. Resource [areaNwkInfo]**

260 **D.5.1. Introduction**

261 The detailed description can be found in clause D.5 of Architecture TS-0001 [6].

262 **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	

264 **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	

265

266 **D.5.2. Resource Specific Procedure on CRUD Operations**

267 When management is performed using external management technologies, procedures defined in 7.3.14.2 <mgmtObj>  
268 specific procedures shall be used. The following clauses define additional procedures besides generic procedure defined  
269 in 7.2.1.2.

270 **D.5.2.1. Create**

271 **.Originator:**

272 No change from the generic procedures in clause 7.2.1.2.1.

273 **Receiver:**

274 No change from the generic procedures in clause 7.2.1.2.2.

275 **D.5.2.2. Update**

276 **Originator:**

277 No change from the generic procedures in clause 7.2.1.2.1.

278 **Receiver:**

279 No change from the generic procedures in clause 7.2.1.2.2.

280 **D.5.2.3. Retrieve**

281 **Originator:**

282 No change from the generic procedures in clause 7.2.1.2.1.

283 **Receiver:**

284 No change from the generic procedures in clause 7.2.1.2.2.

285 **D.5.2.4. Delete**

286 **Originator:**

287 No change from the generic procedures in clause 7.2.1.2.1.

288 **Receiver:**

289 No change from the generic procedures in clause 7.2.1.2.2.

290

291 **D.6. Resource [areaNwkDeviceInfo]**

292 **D.6.1. Introduction**

293 The detailed description can be found in clause D.6 of Architecture TS-0001 [6].

294 **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	

295

296

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	

297

## 298 D.6.2. Resource Specific Procedure on CRUD Operations

299 When management is performed using external management technologies, procedures defined in 7.3.14.2 <mgmtObj>  
300 specific procedures shall be used. The following clauses define additional procedures besides generic procedure defined  
301 in 7.2.1.2.

### 302 D.6.2.1. Create

#### 303 *Originator:*

304 No change from the generic procedures in clause 7.2.1.2.1.

#### 305 *Receiver:*

306 No change from the generic procedures in clause 7.2.1.2.2.

### 307 D.6.2.2. Update

#### 308 *Originator:*

309 No change from the generic procedures in clause 7.2.1.2.1.

#### 310 *Receiver:*

311 Primitive specific operation additional to Recv-6.5 "Create/Update/Retrieve/Delete/Notify operation is performed":

312 When the attribute *listOfNeighbors* of the [areaNwkDeviceInfo] is updated, the receiver shall modify the corresponding  
313 connection relationship among devices in the M2M Area Network by sending signals to non-oneM2M Nodes which is  
314 out of scope of oneM2M. According to the response from the non-oneM2M nodes of the modify signal, the receiver  
315 shall corresponding update the [areaNwkDeviceInfo] resource which may include the update of the *listOfNeighbors* and  
316 the *devType* attribute. The modify may include change of the attach point of the device or removal from the area  
317 network.

### 318 D.6.2.3. Retrieve

#### 319 *Originator:*

320 No change from the generic procedures in clause 7.2.1.2.1.

#### 321 *Receiver:*

322 No change from the generic procedures in clause 7.2.1.2.2.



323

#### 324 D.6.2.4.Delete

325 **Originator:**

326 No change from the generic procedures in clause 7.2.1.2.1.

327 **Receiver:**

328 No change from the generic procedures in clause 7.2.1.2.2.

329

### 330 D.7. Resource [battery]

#### 331 D.7.1. Introduction

332 The detailed description can be found in clause D.7 of Architecture TS-0001 [6].

333 **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	

334

335 **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	

336

#### 337 D.7.2. Resource Specific Procedure on CRUD Operations

338 When management is performed using external management technologies, procedures defined in 7.3.14.2 <mgmtObj>  
339 specific procedures shall be used. The following clauses define additional procedures besides generic procedure defined  
340 in 7.2.1.2.

##### 341 D.7.2.1.Create

342 **.Originator:**

343 No change from the generic procedures in clause 7.2.1.2.1.

344 **Receiver:**

345 No change from the generic procedures in clause 7.2.1.2.2.

##### 346 D.7.2.2.Update

347 **Originator:**

348 No change from the generic procedures in clause 7.2.1.2.1.

349 **Receiver:**  
 350 No change from the generic procedures in clause 7.2.1.2.2.

351 **D.7.2.3.Retrieve**

352 **Originator:**  
 353 No change from the generic procedures in clause 7.2.1.2.1.

354 **Receiver:**  
 355 No change from the generic procedures in clause 7.2.1.2.2.

356 **D.7.2.4.Delete**

357 **Originator:**  
 358 No change from the generic procedures in clause 7.2.1.2.1.

359 **Receiver:**  
 360 No change from the generic procedures in clause 7.2.1.2.2.

361

362 **D.8. Resource [deviceInfo]**

363 **D.8.1. Introduction**

364 The Resource [deviceInfo] is used to provide information regarding the device.  
 365 The detailed description can be found in clause D.8 of Architecture TS-0001 [6].

366 **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	

367

368 **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	

369

370 **D.8.2. Resource Specific Procedure on CRUD Operations**

371 When management is performed using external management technologies, procedures defined in 7.3.14.2 <mgmtObj>  
372 specific procedures shall be used. The following clauses define additional procedures besides generic procedure defined  
373 in 7.2.1.2.

374 **D.8.2.1. Create**

375 **Originator:**

376 No change from the generic procedures in clause 7.2.1.2.1.

377 **Receiver:**

378 No change from the generic procedures in clause 7.2.1.2.2.

379 **D.8.2.2. Update**

380 **Originator:**

381 No change from the generic procedures in clause 7.2.1.2.1.

382 **Receiver:**

383 No change from the generic procedures in clause 7.2.1.2.2.

384 **D.8.2.3. Retrieve**

385 **Originator:**

386 No change from the generic procedures in clause 7.2.1.2.1.

387 **Receiver:**

388 No change from the generic procedures in clause 7.2.1.2.2.

389 **D.8.2.4. Delete**

390 **Originator:**

391 No change from the generic procedures in clause 7.2.1.2.1.

392 **Receiver:**

393 No change from the generic procedures in clause 7.2.1.2.2.

394

395

396 **D.9. Resource [deviceCapability]**

397 **D.9.1. Introduction**

398 The Resource [deviceCapability] is used to provide information regarding the device.

399 The detailed description can be found in clause D.9 of Architecture TS-0001 [6].

400 **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	

401  
402 **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 detached 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	<ul style="list-style-type: none"> <li>true: the device capability is enabled</li> <li>false: the device capability is disabled</li> </ul>
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

403  
404 **D.9.2. Resource Specific Procedure on CRUD Operations**

405 When management is performed using external management technologies, procedures defined in 7.3.14.2 <mgmtObj>  
406 specific procedures shall be used. The following clauses define additional procedures besides generic procedure defined  
407 in 7.2.1.2.

408 **D.9.2.1. Create**

409 **Originator:**

410 No change from the generic procedures in clause 7.2.1.2.1.

411 **Receiver:**

412 No change from the generic procedures in clause 7.2.1.2.2.

413 **D.9.2.2. Update**

414 **Originator:**

415 No change from the generic procedures in clause 7.2.1.2.1.

416 **Receiver:**

417 Primitive specific operation additional to Recv-6.5 "Create/Update/Retrieve/Delete/Notify operation is performed":

418 When the attribute *enable* of the *deviceCapability* resource is updated to TRUE, enable the device capability of the  
419 *deviceCapability* resource.

420 When the attribute *disable* of the *deviceCapability* resource is updated to TRUE, disable the device capability of the  
421 *deviceCapability* resource.

422

### D.9.2.3. Retrieve

423

**Originator:**

424

No change from the generic procedures in clause 7.2.1.2.1.

425

**Receiver:**

426

No change from the generic procedures in clause 7.2.1.2.2.

427

428

### D.9.2.4. Delete

429

**Originator:**

430

No change from the generic procedures in clause 7.2.1.2.1.

431

**Receiver:**

432

No change from the generic procedures in clause 7.2.1.2.2.

433

434

## D.10. Resource [reboot]

435

### D.10.1. Introduction

436

The Resource [reboot] is used to provide information regarding the device.

437

The detailed description can be found in clause D.10 of Architecture TS-0001 [6].

438

439

**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	

440

441

**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

442

### D.10.2. Resource Specific Procedure on CRUD Operations

443

When management is performed using external management technologies, procedures defined in 7.3.14.2 <mgmtObj> specific procedures shall be used. The following clauses define additional procedures besides generic procedure defined in 7.2.1.2.

444

445

446

447 D.10.2.1. Create

448 **Originator:**

449 No change from the generic procedures in clause 7.2.1.2.1.

450 **Receiver:**

451 No change from the generic procedures in clause 7.2.1.2.2.

452 D.10.2.2. Update

453 **Originator:**

454 No change from the generic procedures in clause 7.2.1.2.1.

455 **Receiver:**

456 Primitive specific operation additional to Recv-6.5 "Create/Update/Retrieve/Delete/Notify operation is performed":

457 When the attribute *reboot* of the [reboot] resource is updated to TRUE, reboot the corresponding node.

458 When the attribute *factoryReset* of the [reboot]resource is updated to TRUE, factoryReset the corresponding node.

459 D.10.2.3. Retrieve

460 **Originator:**

461 No change from the generic procedures in clause 7.2.1.2.1.

462 **Receiver:**

463 No change from the generic procedures in clause 7.2.1.2.2.

464 D.10.2.4. Delete

465 **Originator:**

466 No change from the generic procedures in clause 7.2.1.2.1.

467 **Receiver:**

468 No change from the generic procedures in clause 7.2.1.2.2.

469

470

471 D.11. Resource [eventLog]

472 D.11.1. Introduction

473 The Resource [eventLog] is used to provide information regarding the device.

474 The detailed description can be found in clause D.11 of Architecture TS-0001 [6].

475 **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	

476

477

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

478

## 479 D.11.2. Resource Specific Procedure on CRUD Operations

480 When management is performed using external management technologies, procedures defined in 7.3.14.2 <mgmtObj>  
 481 specific procedures shall be used. The following clauses define additional procedures besides generic procedure defined  
 482 in 7.2.1.2.

### 483 D.11.2.1. Create

#### 484 *Originator:*

485 No change from the generic procedures in clause 7.2.1.2.1.

#### 486 *Receiver:*

487 No change from the generic procedures in clause 7.2.1.2.2.

### 488 D.11.2.2. Update

#### 489 *Originator:*

490 No change from the generic procedures in clause 7.2.1.2.1.

#### 491 *Receiver:*

492 Primitive specific operation additional to Recv-6.5 "Create/Update/Retrieve/Delete/Notify operation is performed":

493 When the attribute *logStart* of the [eventLog] resource is updated to TRUE, start the logging.

494 When the attribute *logStop* of the [eventLog]resource is updated to TRUE, stop the logging.

### 495 D.11.2.3. Retrieve

#### 496 *Originator:*

497 No change from the generic procedures in clause 7.2.1.2.1.

#### 498 *Receiver:*

499 No change from the generic procedures in clause 7.2.1.2.2.

500

### 501 D.11.2.4. Delete

#### 502 *Originator:*

503 No change from the generic procedures in clause 7.2.1.2.1.

504 **Receiver:**

505 No change from the generic procedures in clause 7.2.1.2.2.

506

507

## 508 D.12. Resource [cmdhPolicy]

509 The Resource [cmdhPolicy] represents a set of rules defining which CMDH parameters will be used by default when a  
510 request issued by a local originator contains the ec (event category) parameter but not all other CMDH parameters.

511 The detailed description can be found in clause D.12 of Architecture TS-0001 [6].

512 **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	

513 Note that the optional <subscription> child resources are not used for CMDH policies.

514 **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)

515 The Resource Specific Procedure on CRUD Operations as specified in clause 7.3.14 for the generic <mgmtObj>  
516 resource type apply.

517

### 518 D.12.1. Resource [activeCmdhPolicy]

519 The resource [activeCmdhPolicy] provides a link to the currently active set of CMDH policies.

520 The detailed description can be found in clause D.12.1 of Architecture TS-0001 [6].

521 **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	

522



523

**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

524

### 525 D.12.2. Resource [cmdhDefaults]

526 The resource [cmdhDefaults] defines default CMDH policy values. The detailed description can be found in clause  
527 D.12.2 of Architecture TS-0001 [6].

528

**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	

529

530

**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)

531

### 532 D.12.3. Resource [cmdhDefEcValue]

533 The resource [cmdhDefEcValue] represents a value for the ec (event category) parameter of an incoming request. The  
534 detailed description can be found in clause D.12.3 of Architecture TS-0001 [6].

535

**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	

536

537

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

538

539

#### D.12.4. Resource [cmdhEcDefParamValues]

540

541

542

543

The resource [cmdhEcDefParamValues] represents a specific set of default values for the CMDH related parameters rqt (request expiration timestamp), rset (result expiration timestamp), oet (operational execution time), rp (response persistence) and da (delivery aggregation) that are applicable for a given ec (event category) if these parameters are not specified in the request. The detailed description can be found in clause D.12.4 of Architecture TS-0001 [6].

544

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	

545

546

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

547

548

#### D.12.5. [cmdhLimits] Resource

549

550

The Resource [cmdhLimits] represents limits for CMDH related parameter values. The detailed description can be found in clause D.12.5 of Architecture TS-0001 [6].

551

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	

552

553

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

554

### 555 D.12.6. Resource [cmdhNetworkAccessRules]

556 The resource [cmdhNetworkAccessRules] defines the usage of underlying networks for forwarding information to other  
 557 CSEs during processing of CMDH-related requests in a CSE. The detailed description can be found in clause D.12.6 of  
 558 Architecture TS-0001 [6].

559

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	

560

561

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)

562

### 563 D.12.7. Resource [cmdhNwAccessRule]

564 The resource [cmdhNwAccessRule] defines limits in usage of specific underlying networks for forwarding information  
 565 to other CSEs during processing of CMDH-related requests. The detailed description can be found in clause D.12.7 of  
 566 Architecture TS-0001 [6].

567

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	

568

569

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

570

### D.12.8. Resource [cmdhBuffer]

571

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].

572

573

574

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	

575

576

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.

577

578

## 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:

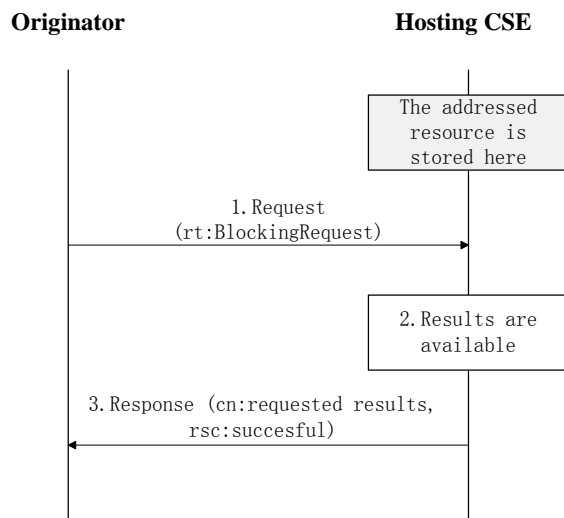


Figure E.1-1: Blocking access to resource

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

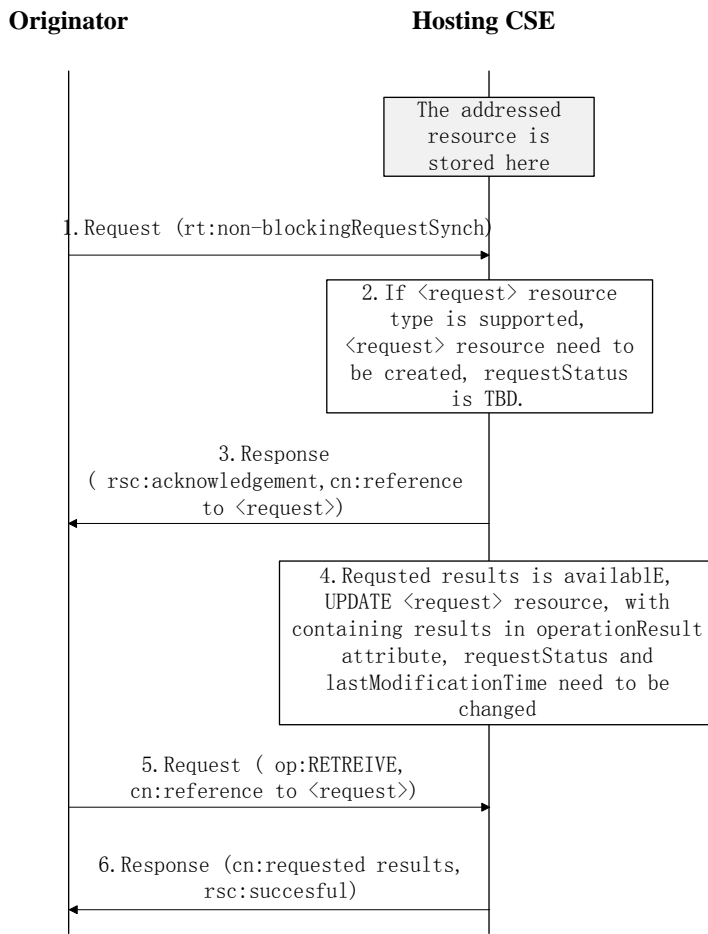
The Originator demands a non-Blocking Communication, with the **Response Type** parameter of the Request setting to 'nonBlockingRequestSynch', eg, the Hosting CSE responds after acceptance with an Acknowledgement confirming, then it will further process the Request. The Hosting CSE of the Request s needs to locally create a <request> resource pertaining to the Request received and repond with an acknowledge Response with the reference of the created

602 <request> resource as the cn of the Response. Then the Receiver needs to continue forward the Request to the next CSE  
 603 if the Hosting CSE is not the Hosting CSE of the addressed resource. Or the Hosting CSE needs to start handling the  
 604 Request if the Hosting CSE is the Hosting CSE of the addressed resource.

605 The Originator if the Request may retrieve the <request> resource afterwards to inspect the final result of the Request if  
 606 it is available.

607 The interaction employing non-blocking mode needs to execute the following steps in order:

608



609

610 **Figure E.2.1-1: non-Blocking accesse to resource in synchronous mode (no hop)**

611

- 612 1. The originator sends a request for accessing resources, the **Response Type** parameter of request is set to  
 613 'nonblockingRequestSynch'.
- 614 2. In case of the Hosting CSE supports the <request> resource type, it will create an instance of <request> resource,  
 615 the **Response Status Code** parameter of response needs to be set to acknowledgement, the value is TBD, and a  
 616 reference to <request> resource is provided in the *content*. Please refer to Table 7.3.11.1-2 for other attributes.

- 617 3. Hosting CSE sends a response to the Originator, the **Response Status Code** parameter of response needs to be set to  
618 acknowledgement, the value is TBD , and a reference to <request> resource is provided in the *content*.
- 619 4. After the requested operation has finished, Hosting CSE will UPDATE the <request> resource, the requested results  
620 needs to be contained in the **operationResult** attribute, the values of *requestStatus* and *lastModifiedTime* needs to  
621 be changed.
- 622 5. Originator requests to RETREIVE the original requested results by addressing the <request> resource.
- 623 6. Hosting CSE responds to Originator, the response contains the requested results in *resource content*, and the  
624 **Response Status Code** parameter of response needs to be set to successful, the value is TBD.
- 625 A variation of synchronous case is depicted in the following clauses. In this variation it is assumed that the addressed  
626 resource is not stored in the Registrar CSE, then the Registrar CSE needs to be a Transit CSE to forward the request to  
627 the hosting CSE.
- 628 The interaction needs to execute the following steps in order:
- 629
- 630

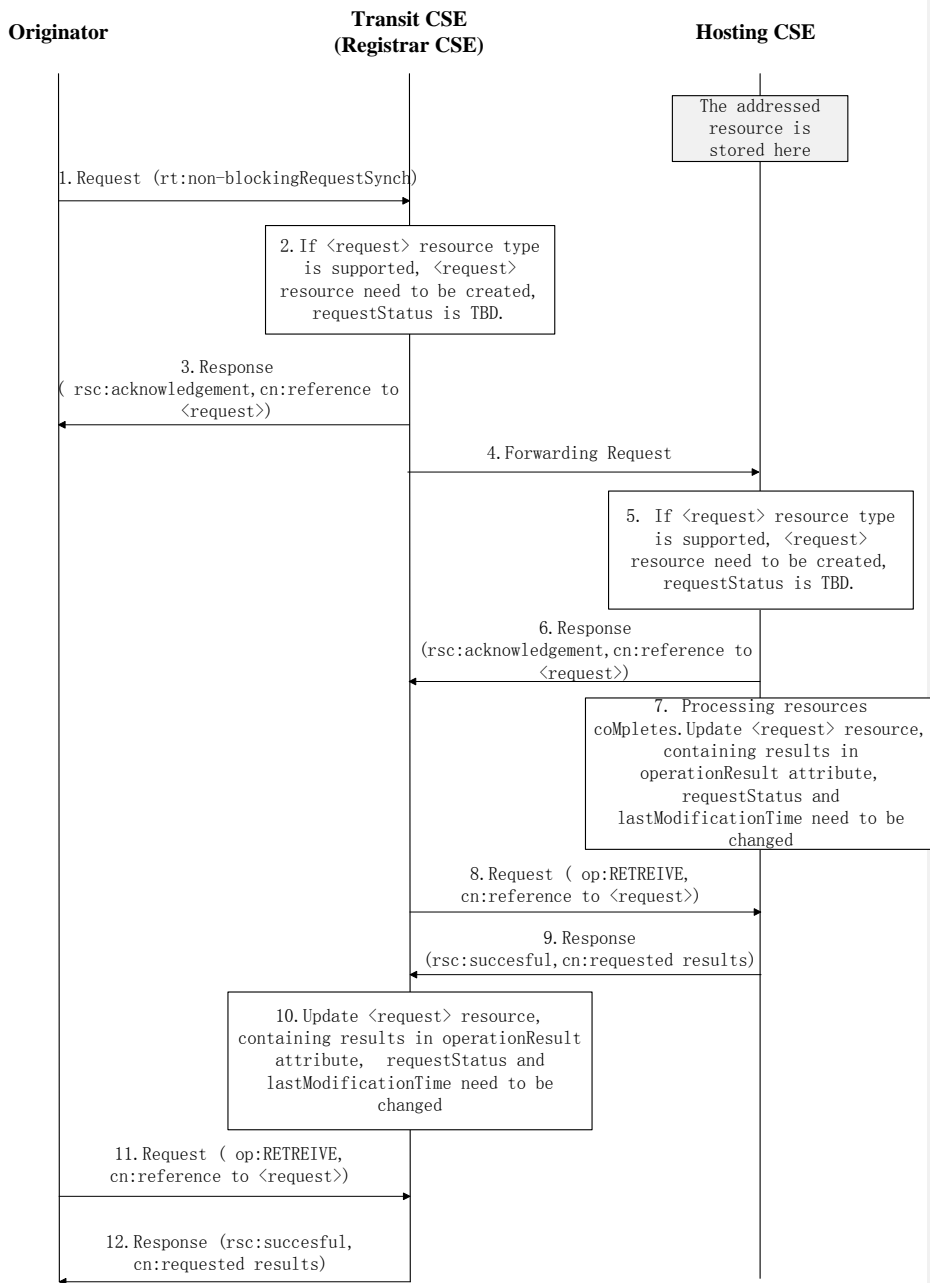


Figure E.2.1-2: non-Blocking accesse to resource in synchronous mode (one hop)

631  
632

633



- 634 1. The originator sends a request for accessing resources to Registrar CSE (Transit CSE, not Hosting CSE), the  
635 **Response Type** parameter of request is set to 'nonblockingRequestSynch'.
- 636 2. In case of the Transit CSE supports the <request> resource type, it will create an instance of <request> resource. The  
637 *requestStatus* needs to be set, the value is TBD. Please refer to Table 7.3.11.1-2 for other attributes.
- 638 3. Transit CSE sends a response to the Originator, the **Response Status Code** parameter of response needs to be set to  
639 acknowledgement, the value is TBD, and a reference to <request> resource is provided in the *content*.
- 640 4. Transit CSE forwards the original request to Hosting CSE.
- 641 5. In case of the Hosting CSE supports the <request> resource type, it will create an instance of <request> resource.  
642 The *requestStatus* needs to be set, the value is TBD. Please refer to Table 7.3.11.1-2 for other attributes.
- 643 6. Hosting CSE sends a response to the Originator, the **Response Status Code** parameter of response needs to be set to  
644 acknowledgement, the value is TBD, and a reference to <request> resource is provided in the *content*.
- 645 7. Hosting CSE processes the resource according to the requested operation, when the operation completes, Hosting  
646 CSE updates the <request> resource, the results needs to be contained in the **operationResult** attribute, and the  
647 values of *requestStatus* and *lastModifiedTime* needs to be changed.
- 648 8. Transit CSE requests to RETREIVE the original requested results by addressing the <request> resource.
- 649 9. Hosting CSE send a response to the Transit CSE. The requested result needs to be contained in the *content* of  
650 request.
- 651 10. Transit CSE updates the <request> resource, the results needs to be contained in the **operationResult** attribute, and  
652 the values of *requestStatus* and *lastModifiedTime* needs to be changed.
- 653 11. Originator requests to RETREIVE the original requested results by addressing the <request> resource.
- 654 12. Hosting CSE responds to Originator, the response contains the requested results in *resource content*, and the  
655 **Response Status Code** parameter of response needs to be set to successful, the value is TBD.

## 656 E.2.2. Asynchronous Case

657

658

659

660

## Annex F (informative): Guidelines for one M2M resource type XSD

This Annex contains rules to be followed when creating XML Schemas Definition (XSD files to represent the oneM2M resources. The XSD files 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 XSD 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 XSD file should include a schema element with following namespace declaration:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.onem2m.org/xml/protocols"
  xmlns:m2m="http://www.onem2m.org/xml/protocols"
  elementFormDefault="unqualified" attributeFormDefault="unqualified" >
```

This defines the prefix xs: for the XML Schema namespace, a target namespace <http://www.onem2m.org/xml/protocols>, and the prefix m2m: as equivalent for the target namespace. Locally declared elements and attributes shall be unqualified (elementFormDefault and attributeFormDefault declarations are not strictly required since “unqualified” is the default value setting).

- 2) Each Resource XSD file 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 as an XML instance) contains an m2m: (or equivalent) namespace prefix. It shall not contribute anything to the m2m: namespace other than this root element.
- 3) 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.
- 4) Each resource attribute of the Resource Type in accordance with [6] 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 instance representation of the resource.
- 5) 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.
- 6) Each Resource attribute shall be declared to use one of the following data types:
  - a. A data type listed in clause 6.3.1 or エラー! 参照元が見つかりません。 .
  - b. A list of one of the data types listed in clause 6.3.1 or エラー! 参照元が見つかりません。 . If the list type is not already included in エラー! 参照元が見つかりません。 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 エラー! 参照元が見つかりません。 . This may be added to clause エラー! 参照元が見つかりません。 , or defined inside the XSD file for the resource, but in the latter case it must be defined as an anonymous type in the attribute declaration itself.

706 d. An anonymous complex type defined as part of the attribute declaration (inside the XSD file for the  
707 resource). The complex type should only be composed out of the types listed in clause 6.3.1 or エラ  
708 ー! 参照元が見つかりません。 .

- 709 7) If a data type is used by more than one attribute (either in the same resource or in two different resources) it  
710 must be included in エラー! 参照元が見つかりません。 , and referenced by each attribute that uses it.  
711 Options 6b, 6c, 6d should only be used in cases where the type is only used by one attribute.
- 712 8) With the exception of CSEBase, all Resource types will extend one of the XML complex  
713 types regularResourceType , subordinateResourceType or announceableResourceType defined in the file CDT-  
714 commonTypes-v1\_0\_0.xsd.
- 715 9) The resource-specific attributes and child resources shall appear as a sequence of elements in the XSD file,  
716 with their order being determined by the order shown in the tables in clause 9.6 of [6].
- 717 10) Each XSD file shall include an XML comment that contains a oneM2M Copyright Notification Notice of  
718 Disclaimer & Limitation of Liability, and a change history. The change history is to be filled in only after the  
719 initial release.
- 720 11) To enable distinction between element names used for resource attributes and their data types in the m2m:  
721 namespace, it shall be avoided to use identical names. It is recommended to use the text suffix 'Type' in data  
722 type names.

723 Example:

724 `<xs:element name="status" type="m2m:statusType />`  
725

---

## 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.

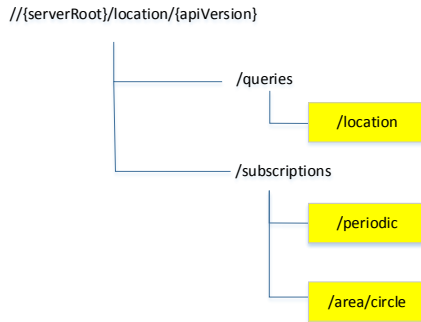
Additionally, in order to track the terminal's movement in relation to the geographic area (circle), crossing in and out (more detail usage is defined in the annex E of TS-0003) it is also proposed to use a service of the Terminal Location API:

- 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].

761



762

763

**Figure G.1.2-1: Resource Structure defined by NetAPI for Terminal Location**

764

765

766

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.

767

**Table G.1.2-3: Applicable NetAPI for Terminal Location**

Capability	URL Base URL:	Resource Type	Operations			
			C	R	U	D
Terminal location	/location	<i>TerminalLocation</i>	no	return current location of the terminal	no	no
Periodic location notification subscriptions	/periodic	<i>PeriodicNotificationSubscription</i> (used for CREATE)	create new subscription	return all subscriptions	no	No
Area notification subscription	/area/circle	<i>CircleNotificationSubscription</i> (used for CREATE)	create a new subscription	return all subscriptions	No	no

768

769

770

771

772

773

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].

774 **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	<i>locationTargetID</i> in the <locationPolicy> resource type
locationRetrievalStatus	common:RetrievalStatus	Status of retrieval for this terminal address.	<i>locationStatus</i> in the <locationPolicy> resource type
currentLocation	LocationInfo	Location of terminal.	<i>Content</i> in the <contentInstance> resource type

775  
776 **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	<i>locationTargetID</i> in the <locationPolicy> resource type
frequency	xsd:int	Maximum frequency (in seconds) of notifications (can also be considered minimum time between notifications) per subscription.	<i>locationUpdatePeriod</i> 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.	<i>locationUpdatePeriod</i> in the <locationPolicy> resource type

777  
778 **Table 0-6: Resource Type Definition – CircleNotificationSubscription**

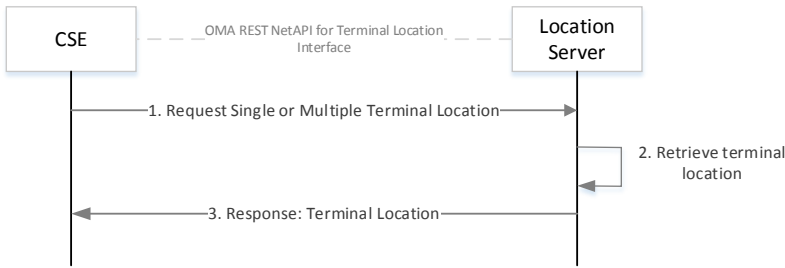
Attributes	OMA NetAPI Defined Type	Description	Relevant Attribute defined by oneM2M
Latitude	xsd:float	Latitude of center point.	<i>accessControlLocationRegion</i> in the <accessControlPolicy> resource type
longitude	xsd:float	Longitude of center point.	<i>accessControlLocationRegion</i> in the <accessControlPolicy> resource type
Radius	xsd:float	Radius of circle around center point in meters.	<i>accessControlLocationRegion</i> in the <accessControlPolicy> resource type
checkImmediate	xsd:boolean	Check location immediately after establishing subscription.	

780 G.1.3. Procedures for Terminal Location

781 G.1.3.1. Request in a Single Query toward a Location Server

782 This procedure shows how to request and return location for a M2M Node.

783



784

785 **Figure G.1.3.1-1: Single Query Toward Location Server**

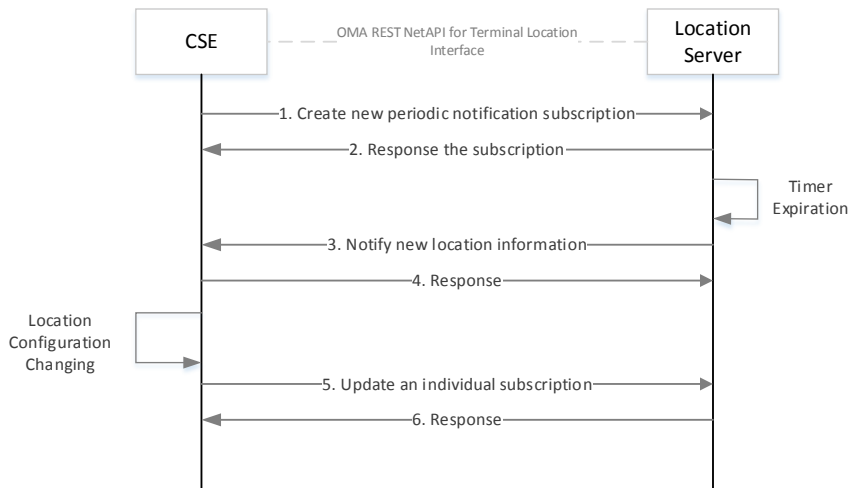
786 1. A Hosting CSE requests location for a single terminal (Node) by means of OMA REST NetAPI for terminal  
787 location API. This request message shall contain terminal address and Request URL with the address of  
788 Location Server using RETRIEVE operation.  
789 In this step, the *TerminalLocation* resource type described in Table G.1.2-3 shall be used with RETRIEVE  
790 operation.

791 NOTE: GET operation shall be used for this RETRIEVE operation.

- 792 2. The Location Server shall retrieve the location information of the terminal.
  - 793 3. After the successful retrieve, the Hosting CSE receives the location information.
- 794

795 G.1.4. Subscribe to Notifications for Periodic Location Updates

796 This procedure shows how to control subscriptions for periodic notifications about terminal location.



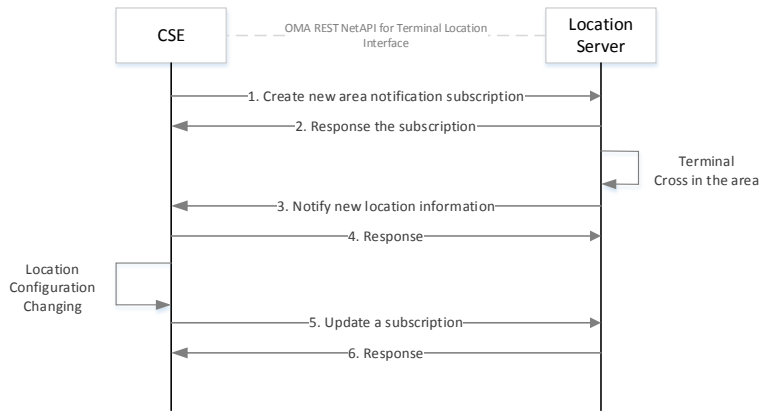
**Figure G.1.4-1: Subscribe to Notification for Periodic Location Updates**

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.





819  
820 **Figure G.1.5-1: Subscribe to Notification for Area Updates**

- 821 1. A Hosting CSE shall create a new area notification subscription to track the terminal's movement in relation  
822 to the geographical area (circle), crossing in and out. In this step, the *CircleNotificationSubscription*  
823 type described in the table-G.1-3 shall be used with CREATE operation.  
824 NOTE: POST operation shall be used for this CREATE operation.

- 825 2. After the successful creation of subscription, the Hosting CSE shall receive the response.

- 826 3. When the target terminal crosses in or out the specified area (circle), the location server shall notify the  
827 application of current location information.  
828 In this step, the notification message shall be used as NOTIFY operation.

829 NOTE: Alternatively, the hosting CSE obtains the notifications using a Notification Channel [i.4].

830 NOTE: POST operation shall be used for this NOTIFY operation

- 831 4. After the successful receiver of notification, the Hosting CSE shall send a response to the location server.

- 832 5. Based upon the location configuration change by the Hosting CSE, it updates an individual subscription for  
833 area location notification.  
834 In this step, the *CircleNotificationSubscription* resource type described in the table-G.1-3 shall be used with  
835 UPDATE operation.

836 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

**Editor's Note: short name for request primitive parameters is not consistent with clause 7.2.1.2.1. Need to be harmonized.**

CMDH-related response message parameters are:

- *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 reached 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 objects defined for legacy device management technologies shall contain the specified default values as described in the [cmdhPolicy] 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

888 context information of the CSE defined in *'requestContext'* attribute of the found [cmdhLimits] as well as subscription  
889 to this [cmdhLimits] resource for all AEs corresponding to the AE-ID or an App-ID appearing in the *'requestOrigin'*  
890 attribute. The subscription(s) shall be established when the [cmdhPolicy] is provisioned or re-provisioned and any of  
891 found [cmdhLimits] child resource has the attribute *'requestContextNotification'* that is set to true. Hence, both this  
892 policy establishment and changes of the context information and the [cmdhLimits] resource shall be notified to the  
893 respective AEs and the notification shall contain the limits for CMDH related parameter values defined in  
894 [cmdhLimits], context information and subscription reference ID. After this, the AEs received the notification shall  
895 send only allowed *'ec'* messages if *'ec'* is specified by the AEs.

## 896 H.2. CMDH Processing: Processing request or response 897 messages requiring the Receiver CSE to forward information to 898 another CSE

### 899 H.2.1. Applicability of CMDH processing

900 If a request or response message that is targeting an entity or a resource in the *'to'* parameter that is not among any of

- 901 • the receiver CSE itself,
- 902 • an AE registered with the receiver CSE,
- 903 • a resource hosted on the receiver CSE,

904 and if the message is not a response message with an acknowledgement response code, the receiver CSE of that  
905 message needs to forward the message to another CSE via CMDH processing, see also the description in Clause 7.2.1.2.  
906 *Description of Generic Procedures* of this TS. For forwarding a message to the target CSE indicated by the *'to'*  
907 parameter of the message, the receiver CSE shall determine to which CSE the message needs to be forwarded next. In  
908 the following clauses this CSE is referred to as the *'next CSE'*. CMDH processing shall be carried out as described in  
909 the following clauses.  
910

### 911 H.2.2. Partitioning of CMDH processing

912 The CMDH processing consists of two parts:

913

- 914 A. CMDH message validation: This includes message parameter pre-processing, deciding on  
915 acceptance for transporting the message, and buffering of messages.  
916 This procedure defines how incoming request or response messages that need to be  
917 forwarded to other CSE(s) shall be pre-processed, how a decision on acceptance of the  
918 message for forwarding to another CSE shall be derived and how the messages shall be  
919 queued up before the actual forwarding can happen. Details of CMDH validation are defined  
920 in clause H.2.3. .  
921
- 922 B. CMDH message forwarding: This includes selecting buffered messages and communication  
923 path for forwarding the message to another CSE.  
924 This procedure defines how to select among the messages buffered for forwarding to other  
925 CSEs the ones that need to be transported at a certain time and how to select an appropriate  
926 communication path for transporting the message(s). Details of CMDH message forwarding  
927 are defined in Annex H.2.4. .  
928

929 CMDH message validation (Part A) will be carried out for each incoming new message for which CMDH processing is  
930 applicable.

931 If CMDH message validation is successful, the received message shall be queued up for the CMDH message  
932 forwarding process (Part B) including the associated *'storagePriority'* value as defined in the applicable [cmdhBuffer]  
933 resource (see details in the CMDH message validation procedure).

934 If the queued message was a request message and it was done in non-blocking mode then:

- 935 ◆ if the Receiver CSE supports the <request> resource type, it shall create a <request> resource representing the  
936 pending non-blocking request
- 937 ◆ the Receiver CSE shall send an acknowledgement response message to the entity that sent the request message  
938 directly via Mca or Mcc to the receiver CSE indicating the acceptance of the request
- 939 ◆ if the receiver CSE supports the <request> resource type it shall provide a reference to the created <request>  
940 resource in the *cn* parameter of the response.

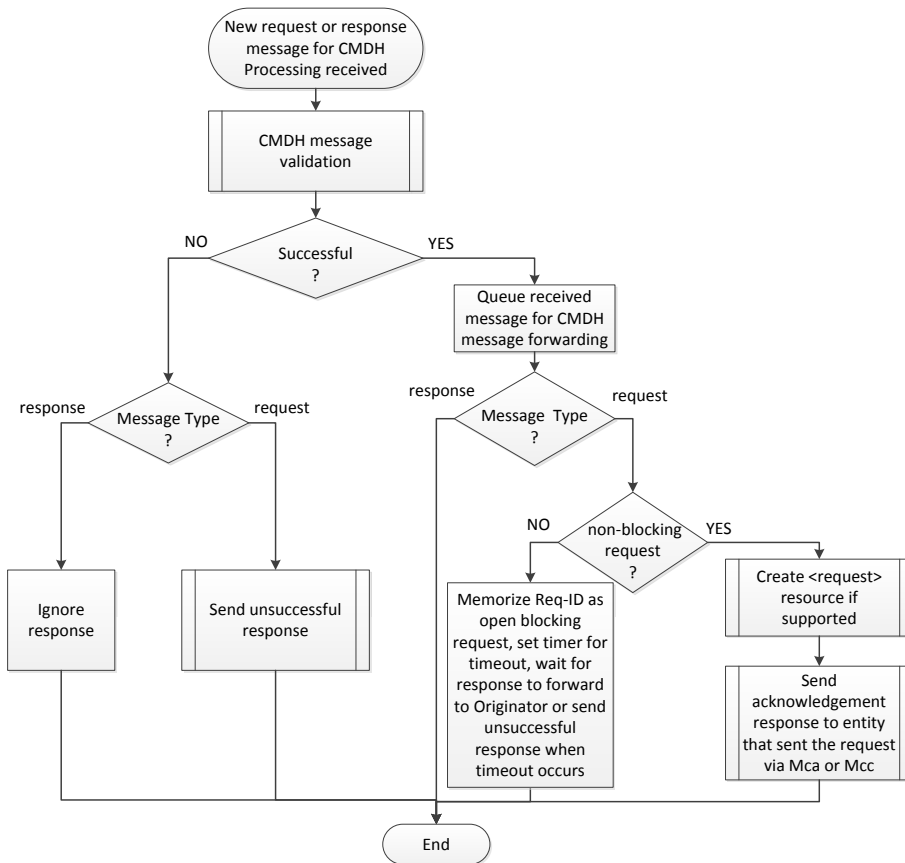
941 After successful forwarding of such a request message, any incoming response message matching with the Request-ID  
942 and the Originator in the <request> resource shall be parsed to update the corresponding attributes of the <request>  
943 resource. In case a non-blocking synchronous request was forwarded successfully and a response with  
944 acknowledgement was received, it is the responsibility of the CSE that forwarded the message to periodically poll the  
945 status of the <request> resource created on the next CSE and update the locally created <request> resource accordingly.  
946 When the locally created <request> resource expires the hosting CSE can remove it. Details on <request> resource  
947 specific procedures for polling results are defined in clause 7.2.2.1.

948 If the queued message was a request message and it was done in blocking mode then memorize the open blocking  
949 request by storing its Request-ID and Originator and set a timer for a timeout until which a matching response message  
950 with the same Request-ID and Originator shall be received by the CSE processing this message. If no matching  
951 response is received when the timeout expires, the receiver CSE shall send a response message to the entity that sent the  
952 request to the Receiver CSE indicating unsuccessful processing of the request, unless the Receiver CSE and the  
953 Originator are the same. If Receiver CSE and Originator are the same, the Originator can decide internally whether to  
954 retry forwarding of the message.

955 If CMDH message validation is not successful, then the received message shall either get ignored – in case the received  
956 message is a response message – or a new unsuccessful response message shall be sent back to the entity that sent the  
957 message to the Receiver CSE – in case the received message is a request message and the Originator is not the  
958 Receiver CSE. If Receiver CSE and Originator are the same, the Originator can decide internally whether to create a  
959 new request message.

960 The CMDH message forwarding process (Part B) will handle all queued up messages that shall be forwarded to another  
961 CSE. This process shall always be carried out when messages are pending for forwarding to another CSE.

962 The flow of CMDH processing is depicted in the flowchart below:



963

### 964 H.2.3. CMDH message validation procedure

965

966 In CMDH message validation, pre-processing of CMDH related parameters of a message for which CMDH-processing  
 967 applies, deriving the decision on acceptance of a message and the buffering of that messages shall be carried out in line  
 968 with the following steps. A summary of this processing is depicted in the flow chart at the end of this clause.  
 969

#### 970 1. Filling in missing CMDH-related parameters:

971

##### 972 1.1. Determine the value that shall be used for the 'ec' parameter of the processed message

973

974 1.1.1.If the message contains an 'ec' parameter: Use the value of the 'ec' parameter provided in the  
 975 message.

976

977 1.1.2.If the message does not contain an 'ec' parameter:

978

979 1.1.2.1. Lookup all [cmdhDefEcValue] child resources of the [cmdhDefaults] resource that  
 980 is a child resource of the provisioned active [cmdhPolicy] resource.

981

982 1.1.2.2. If the message is a request message and any of the attributes '*requestContext*',  
983 and '*requestCharacteristics*' are present in the found [cmdhDefEcValue] resources,  
984 discard all [cmdhDefEcValue] resources from the list of found items for which the  
985 context conditions or the request characteristics at time of processing the request  
986 message are not met, respectively.

987 1.1.2.3. Among the remaining found [cmdhDefEcValue] resources do the following  
988 selection:  
989

990 1.1.2.3.1. If present, select the [cmdhDefEcValue] resource containing the AE-ID in the list  
991 defined by the '*requestOrigin*' attribute which matches with the '*fr*' parameter in  
992 case of a request message or with the '*to*' parameter in case of a response  
993 message. If multiple [cmdhDefEcValue] resources match, select the one with the  
994 lowest value in the '*order*' attribute. Continue processing with step 1.1.2.4  
995

996 1.1.2.3.2. If present, select the [cmdhDefEcValue] resource containing the App-ID in the list  
997 defined by the '*requestOrigin*' attribute which matches with the '*fr*' parameter in  
998 case of a request message or with the '*to*' parameter in case of a response  
999 message. If multiple [cmdhDefEcValue] resources match, select the one with the  
1000 lowest value in the '*order*' attribute. Continue processing with step 1.1.2.4  
1001

1002 1.1.2.3.3. If present, select the [cmdhDefEcValue] resource containing the string 'localAE' in  
1003 the list defined by the '*requestOrigin*' attribute in case of processing a message  
1004 where the '*fr*' parameter is the AE-ID of an AE registered with the CSE  
1005 processing this message. If multiple [cmdhDefEcValue] resources match, select the  
1006 one with the lowest value in the '*order*' attribute. Continue processing with step  
1007 1.1.2.4  
1008

1009 1.1.2.3.4. If present, select the [cmdhDefEcValue] resource containing the string 'thisCSE' in  
1010 the list defined by the '*requestOrigin*' attribute in case of processing a message  
1011 where the '*fr*' parameter is the CSE-ID of the CSE processing this message. If  
1012 multiple [cmdhDefEcValue] resources match, select the one with the lowest value  
1013 in the '*order*' attribute. Continue processing with step 1.1.2.4  
1014

1015 1.1.2.3.5. Select the [cmdhDefEcValue] resource containing the string 'default' in the list  
1016 defined by the '*requestOrigin*' attribute in case of processing a message where  
1017 no other matches were found.  
1018

1019 1.1.2.4. If a [cmdhDefEcValue] resource has been selected in steps 1.1.2.3.1 through  
1020 1.1.2.3.4: Use the value of the '*defEcValue*' attribute of the selected  
1021 [cmdhDefEcValue] resource as the value for the '*ec*' parameter of the message. Else use  
1022 the default value of '**bestEffort**' for the '*ec*' parameter of the message.  
1023  
1024

1025 1.2. Filling in values that shall be used for the remaining CMDH-related parameters of  
1026 messages  
1027

1028 1.2.1. If the message contains any of the CMDH-related parameters '*rqet*', '*rset*', '*oet*', '*rp*': The  
1029 provided values of the respective parameters in the message shall be used. No filling in is  
1030 needed for those parameters. If any of the parameters '*rqet*', '*rset*', '*oet*', '*rp*' present in the  
1031 message is represented with a duration, the receiving CSE shall translate the values of those  
1032 parameters into absolute times by adding the duration to the originating timestamp in the '*ot*'  
1033 parameter of the message. This '*ot*' parameter is an optional message parameter and in case it  
1034 is not present in a message, it shall be filled in by the first receiving CSE of a message using

1035 the time when the message was received.

1036  
1037 1.2.2.If the message parameter '**ec**' has a value of '**bestEffort**', use the following values for any  
1038 missing CMDH-related parameters: For a request message use '**rqet**' = 'infinite', '**rset**' =  
1039 'infinite', '**oet**' = 'now', '**rp**' = 'none', '**da**' = ON. For a response message use '**rset**' =  
1040 'infinite', '**da**' = ON. Continue with step 2.

1041  
1042 1.2.3.If the message parameter '**ec**' has a value of '**immediate**', do not fill in any remaining  
1043 missing CMDH-related parameters and continue with step 2.

1044  
1045 1.2.4. For any of the missing CMDH-related parameters fill in values as follows:

1046  
1047 1.2.4.1. Lookup all [cmdhEcDefParamValues] child resources of the [cmdhDefaults]  
1048 resource that is a child resource of the provisioned active [cmdhPolicy] resource.

1049  
1050 1.2.4.2. Among the found [cmdhEcDefParamValues] resources do the following  
1051 selection:

1052  
1053 1.2.4.2.1.If present, select the [cmdhEcDefParamValues] resource containing the value of  
1054 the '**ec**' parameter of the message in the list defined by the  
1055 '**applicableEventCategory**' attribute. If a match is found, continue processing  
1056 with step 1.2.4.3

1057  
1058 1.2.4.2.2.Select the [cmdhEcDefParamValues] resource that contains the string 'default' in  
1059 the list defined by the '**applicableEventCategory**'.

1060  
1061 1.2.4.3. Use the following attributes of the selected [cmdhEcDefParamValues] resource to fill  
1062 in any missing CMDH-related message parameters: Fill in the value of the attribute  
1063 '**defaultRequestExpTime**' for the parameter '**rqet**' if it is missing. Fill in the value  
1064 of the attribute '**defaultResultExpTime**' for the parameter '**rset**' if it is missing.  
1065 Fill in the value of the attribute '**defaultOpExecTime**' for the parameter '**oet**' if it is  
1066 missing. Fill in the value of the attribute '**defaultRespPersistence**' for the parameter  
1067 '**rp**' if it is missing. Fill in the value of the attribute '**defaultDelAggregation**' for the  
1068 parameter '**da**' if it is missing.

1069  
1070 2. Compare CMDH parameters with allowed CMDH parameter limits:

1071 Check if CMDH-related parameters effective for the message are with allowed limits.

1072  
1073 2.1. Lookup all [cmdhLimits] child resources of the provisioned active [cmdhPolicy] resource.

1074  
1075 2.2. If the message is a request message and any of the attributes '**requestContext**', and  
1076 '**requestCharacteristics**' are present in the found [cmdhLimits] resources, discard all  
1077 [cmdhLimits] resources from the list of found items for which the context conditions or the  
1078 request characteristics at time of processing the request message are not met, respectively.

1079  
1080 2.3. Among the remaining found [cmdhLimits] resources do the following selection:

1081  
1082 2.3.1.If present, select the [cmdhLimits] resource(s) containing the AE-ID in the list defined by the  
1083 '**requestOrigin**' attribute which matches with the '**fr**' parameter in case of a request  
1084 message or with the '**to**' parameter in case of a response message. If multiple  
1085 [cmdhLimits] resources match, select the one with the lowest value in the '**order**' attribute.

1086 Continue processing with step 2.4

1087  
1088 2.3.2.If present, select the [cmdhLimits] resource(s) containing the App-ID in the list defined by the  
1089 'requestOrigin' attribute which matches with the 'fr' parameter in case of a request  
1090 message or with the 'to' parameter in case of a response message. If multiple  
1091 [cmdhLimits] resources match, select the one with the lowest value in the 'order' attribute.  
1092 Continue processing with step 2.4

1093  
1094 2.3.3.If present, select the [cmdhLimits] resource(s) containing the string 'localAE' in the list  
1095 defined by the 'requestOrigin' attribute in case of processing a message where the 'fr'  
1096 parameter is the AE-ID of an AE registered with the CSE processing this message. If  
1097 multiple [cmdhLimits] resources match, select the one with the lowest value in the 'order'  
1098 attribute. Continue processing with step 1.1.2.4

1099  
1100 2.3.4.If present, select the [cmdhLimits] resource(s) containing the string 'thisCSE' in the list  
1101 defined by the 'requestOrigin' attribute in case of processing a message where the 'fr'  
1102 parameter is the CSE-ID of the CSE processing this message. If multiple [cmdhLimits]  
1103 resources match, select the one with the lowest value in the 'order' attribute. Continue  
1104 processing with step 2.4

1105  
1106 2.3.5.Select the [cmdhLimits] resource containing the string 'default' in the list defined by the  
1107 'requestOrigin' attribute in case of processing a message where no other matches were  
1108 found.

1109  
1110 2.4. Validate if 'ec' parameter is within allowed range:  
1111 If the 'ec' parameter of the message is not within the list defined by the 'limitsEventCategory'  
1112 attribute of the selected [cmdhLimits] resource, mark CMDH message validation for this message  
1113 as not successful and exit CMDH message validation.

1114  
1115 2.5. Validate if 'rget' parameter is within allowed range:  
1116 If the 'rget' parameter is present in the message and if it is not within the range defined by the  
1117 'limitsRequestExpTime' attribute of the selected [cmdhLimits] resource, mark CMDH message  
1118 validation for this message as not successful and exit CMDH message validation.

1119  
1120 2.6. Validate if 'rset' parameter is within allowed range:  
1121 If the 'rset' parameter is present in the message and if it is not within the range defined by the  
1122 'limitsResultExpTime' attribute of the selected [cmdhLimits] resource, mark CMDH message  
1123 validation for this message as not successful and exit CMDH message validation.

1124  
1125 2.7. Validate if 'oet' parameter is within allowed range:  
1126 If the 'oet' parameter is present in the message and if it is not within the range defined by the  
1127 'limitsOpExecTime' attribute of the selected [cmdhLimits] resource, mark CMDH message  
1128 validation for this message as not successful and exit CMDH message validation.

1129  
1130 2.8. Validate if 'rp' parameter is within allowed range:  
1131 If the 'rp' parameter is present in the message and if it is not within the range defined by the  
1132 'limitsRespPersistence' attribute of the selected [cmdhLimits] resource, mark CMDH message  
1133 validation for this message as not successful and exit CMDH message validation.

1134  
1135 2.9. Validate if 'da' parameter is within allowed range:  
1136 If the 'da' parameter is present in the message and if it is not within the list of allowed values  
1137 defined by the 'limitsDelAggregation' attribute of the selected [cmdhLimits] resource, mark CMDH  
1138 message validation for this message as not successful and exit CMDH message validation.

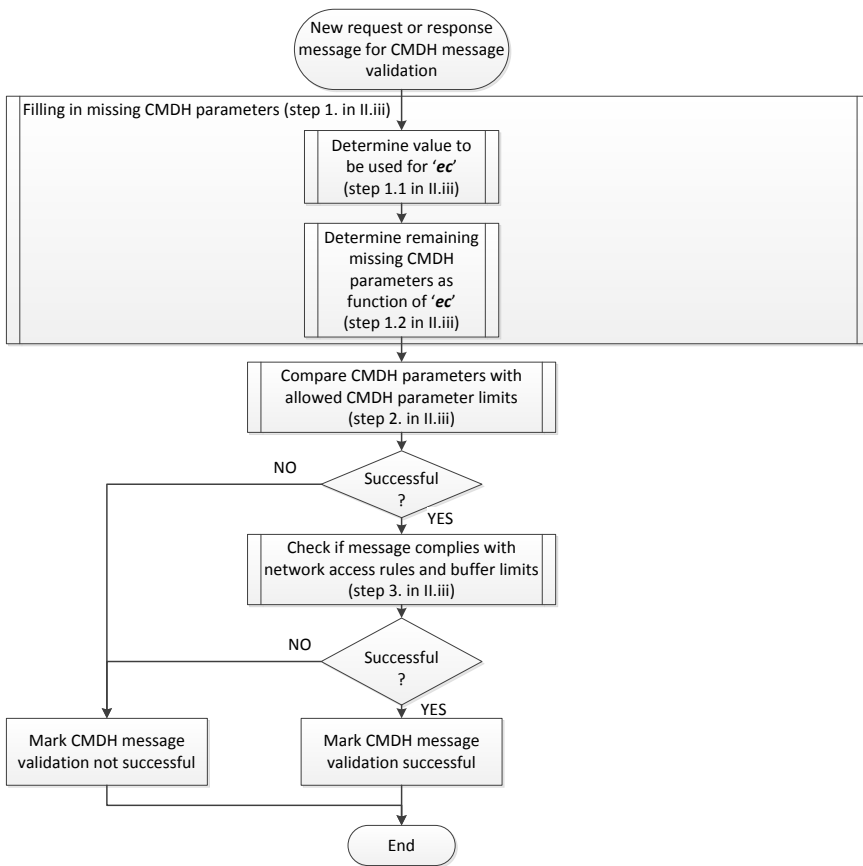


1140 3. Check if message complies with network access rules and buffer limits:  
1141  
1142 3.1. Check if 'ec' is **'immediate'**:  
1143 If the 'ec' parameter of the message is **'immediate'** bypass any checks on buffering or access  
1144 network usage rules. Mark the CMDH message validation for this message as successful and end  
1145 CMDH message validation.  
1146  
1147 3.2. Check if delivering the message is possible within the boundaries of access network usage  
1148 rules in CMDH policies:  
1149  
1150 3.2.1. Lookup all [cmdhNetworkAccessRules] child resources of the provisioned active  
1151 [cmdhPolicy] resource.  
1152  
1153 3.2.2. Among the all found [cmdhNetworkAccessRules] resources do the following  
1154 selection:  
1155  
1156 3.2.2.1. If present, select the [cmdhNetworkAccessRules] resource containing the value of the  
1157 'ec' parameter of the message in the list defined by the 'applicableEventCategory'  
1158 attribute. If a match is found, continue processing with step 3.2.3  
1159  
1160 3.2.2.2. Select the [cmdhNetworkAccessRules] resource that contains the string 'default' in  
1161 the list defined by the 'applicableEventCategory'.  
1162  
1163 3.2.3. Lookup all [cmdhNwAccessRule] child resources of the selected [cmdhNetworkAccessRules]  
1164 resource  
1165  
1166 3.2.4. Among the all found [cmdhNwAccessRule] resources find at least one for which the  
1167 <schedule> child resource 'allowedSchedule' is allowing usage of the corresponding target  
1168 network consistent with the 'rreq' parameter in case of a request message being processed or  
1169 in line with the 'rset' parameter in case of a response message being processed. If no matching  
1170 [cmdhNwAccessRule] resource is found, mark CMDH validation for this message as not  
1171 successful due to lack of scheduling opportunities and end CMDH message validation.  
1172 Otherwise continue.  
1173  
1174 3.3. Check if delivering the message is possible within the boundaries of buffer usage rules in  
1175 CMDH policies:  
1176  
1177 3.3.1. Lookup all [cmdhBuffer] child resources of the provisioned active [cmdhPolicy] resource.  
1178  
1179 3.3.2. Among the all found [cmdhBuffer] resources do the following selection:  
1180  
1181 3.3.2.1. If present, select the [cmdhBuffer] resource containing the value of the 'ec' parameter  
1182 of the message in the list defined by the 'applicableEventCategory' attribute. If a  
1183 match is found, continue processing with step 3.3.3  
1184 3.3.2.2. Select the [cmdhBuffer] resource that contains the string 'default' in the list defined  
1185 by the 'applicableEventCategory'.  
1186  
1187 3.3.3. Check if the amount of memory needed to buffer the message being validated in  
1188 addition to the already buffered messages matching with the same buffer usage policy  
1189 in the selected [cmdhBuffer] resource would exhaust the limit defined by the  
1190 'maxBufferSize' attribute of the selected [cmdhBuffer] resource or if the available  
1191 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.



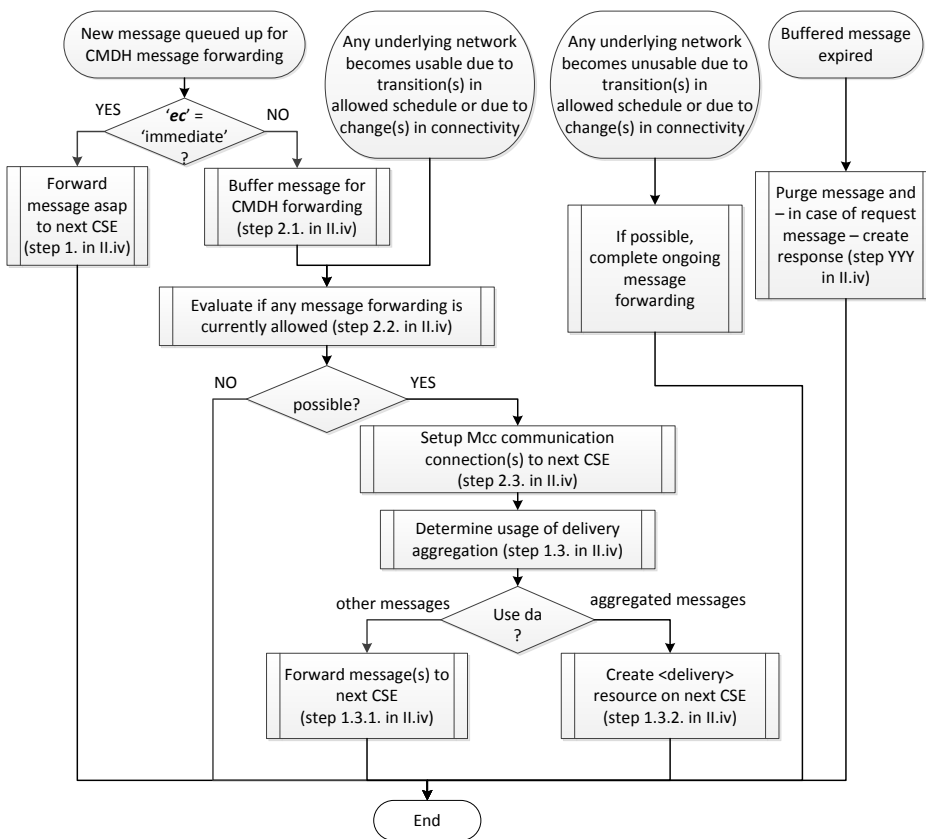
#### H.2.4. CMDH message forwarding procedure

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:

1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218

- 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



1219  
1220  
1221  
1222  
1223

Figure H.2.4-1: (Title is TBD)

When a new message is getting queued up for CMDH message forwarding, carry out the following:

1224 **If the 'ec' parameter of the messages has the value 'immediate':**  
1225 **Forward message as soon as possible to the next CSE. The processing in this situation is described by**  
1226 **the flow chart in**

1227 1. Figure H.2.4-2 below.

- 1228
- 1229 1.1. If a Mcc communication connection to the next CSE for forwarding the message is  
1230 already established, continue with step 1.3.
- 1231
- 1232 1.2. If no Mcc communication connection to the next CSE for forwarding the message is  
1233 established pick one underlying network among all underlying networks that can  
1234 provide communication to the next CSE and establish a Mcc communication connection  
1235 to the next CSE in line with the rules outlined in clause H.2.5. . If establishment of a  
1236 Mcc communication connection to the next CSE was not successful before the message  
1237 expires, continue with step 1.4.
- 1238
- 1239 1.3. Determine whether delivery aggregation or forwarding of the message itself shall be  
1240 used:
- 1241
- 1242 1.3.1. If the message contains a '*da*' parameter set to the value 'ON', the Receiver CSE shall  
1243 forward this message by creation of a <delivery> resource on the next CSE as outlined in  
1244 clause 7.3.10. The receiver CSE can combine the forwarded message in the same  
1245 <delivery> resource with other messages for which the '*da*' parameter set to 'ON' and  
1246 which need to be forwarded to the same target CSE.
- 1247
- 1248 1.3.2. If the message is not forwarded using a <delivery> resource, the receiver CSE  
1249 shall forward the message as is to the next CSE via the established Mcc  
1250 communication connection.
- 1251
- 1252 1.4. If the message could not be forwarded successfully to the next CSE before it expired  
1253 (e.g. due to repeated unsuccessful attempts to establish a Mcc communication  
1254 connection or due to the lack of usable underlying networks), the receiver CSE shall  
1255 carry out the following:
- 1256
- 1257 1.4.1. If the message was a response message, ignore the message. End this cycle of  
1258 CMDH message forwarding and wait for new triggering events.
- 1259
- 1260 1.4.2. If the message was a request message:
- 1261
- 1262 1.4.2.1. If the request was a blocking request:  
1263 Send a unsuccessful response to the pending blocking request with a matching  
1264 Request-ID and Originator indicating the reason for failure and close the  
1265 blocking request. End this cycle of CMDH message forwarding and wait for  
1266 new triggering events.
- 1267
- 1268 1.4.2.2. If the request was a non-blocking request:  
1269 Update the associated <request> resource with matching Request-ID and  
1270 Originator using an unsuccessful response code indicating the reason for  
1271 failure. If the non-blocking request was made in asynchronous mode, send a  
1272 notification with the unsuccessful response to the notification target(s) of the  
1273 request. End this cycle of CMDH message forwarding and wait for new

1274 triggering events.

1275  
1276 1.5. Else, i.e. if the message was forwarded successfully to the next CSE:

1277  
1278 1.5.1. If the message was a response and the Receiver CSE has an open blocking request  
1279 context with a matching Request-ID and matching Originator, mark the open  
1280 blocking request as closed, end this cycle of CMDH message forwarding and wait  
1281 for new triggering events.

1282  
1283 1.5.2. If the message was a request message:

1284  
1285 1.5.2.1. If the request was a blocking request:

1286 Keep the context of the pending blocking request with matching Request-ID  
1287 and matching Originator open and wait for an incoming response message  
1288 with the same Request-ID and Originator. End this cycle of CMDH message  
1289 forwarding and wait for new triggering events.

1290  
1291 1.5.3. If the request was a non-blocking request:

1292 Wait for a response to the forwarded request (e.g. response with acknowledgement  
1293 or unsuccessful response). Update the associated <request> resource with the  
1294 matching Request-ID and Originator using a response code that reflects the status  
1295 of the forwarded request (e.g. accepted by next CSE, unsuccessful). If the next CSE  
1296 responded with an unsuccessful response message and the request was in non-  
1297 blocking asynchronous mode, send a notification request message to the Originator  
1298 of the forwarded request containing the unsuccessful response of the next CSE.  
1299 End this cycle of CMDH message forwarding and wait for new triggering events.

1300  
1301  
1302 2. Else, i.e. when the 'ec' parameter of the messages does not have the value 'immediate':

1303  
1304 **Buffer the message to be forwarded in the CMDH forwarding buffer:**  
1305 **The processing in this situation is described by the flow chart in**

1306 2.1. Figure H.2.4-2 below.

1307  
1308 2.1.1. If the message is a request message and the 'ec' parameter of the messages has the value  
1309 'latest':

1310  
1311 2.1.1.1. If the request message is a notification triggered by a subscription:

1312  
1313 2.1.1.1.1. Find any buffered request message that is a notification triggered by a  
1314 subscription with the same subscription reference.

1315  
1316 2.1.1.2. Else, i.e. if the request message is not a notification triggered by a  
1317 subscription:

1318  
1319 2.1.1.2.1. Find any buffered request message that has the same values in the ('fr', 'to',  
1320 'op' ) parameters as the message being processed

1321  
1322 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  
1323 found message from the CMDH forwarding buffer.

1325 2.1.2. If there is not enough memory available to buffer the message being processed in  
1326 the CMDH forwarding buffer:

1327  
1328 2.1.2.1. Find any buffered messages with storage priority values lower than the one  
1329 assigned to the message being processed.

1330 2.1.2.2. If any messages are found:

1331 Purge enough messages among the found messages so that the message being  
1332 processed can be buffered in the CMDH forwarding buffer. Messages which  
1333 entered the buffer later shall be purged first. In case any request messages  
1334 need to be purged, carry out the following:

1335 2.1.2.2.1. In case of purging a non-blocking request messages:

1336 Update the associated <request> resource with the same Request-ID as  
1337 the purged request message with a status indicating unsuccessful  
1338 completion. If the purged message was made in asynchronous mode,  
1339 send a response to the notification target(s) of the pending non-blocking  
1340 request

1341 2.1.2.2.2. In case of purging a blocking request message:

1342 Send a unsuccessful response to the open blocking request with the same  
1343 Request-ID as in the purged request message and close the blocking  
1344 request.

1345 2.1.2.3. Due to the checking of sufficient memory in CMDH message forwarding  
1346 buffer during CMDH message validation, there should be enough memory  
1347 available to accommodate the message to be buffered at this point. If that is  
1348 still not the case, then do the following:

1349 2.1.2.3.1. In case the message to be buffered is a response message:

1350 Ignore the message to be buffered. End this cycle of CMDH message  
1351 forwarding and wait for new triggering events.

1352 2.1.2.3.2. In case the message to be buffered is a non-blocking request message:

1353 Update the associated <request> resource with the same Request-ID as  
1354 the request message to be buffered with a status indicating unsuccessful  
1355 completion. If the request message to be buffered was made in  
1356 asynchronous mode, send a response to the notification target(s) of the  
1357 pending non-blocking request. End this cycle of CMDH message  
1358 forwarding and wait for new triggering events.

1359 2.1.2.3.3. In case the message to be buffered is a blocking request message:

1360 Respond with an unsuccessful response message to the open blocking  
1361 request with the same Request-ID as in the request message to be  
1362 buffered and close the blocking request. End this cycle of CMDH  
1363 message forwarding and wait for new triggering events.

1364 2.1.3. Store the message to be buffered with its assigned storage priority in the CMDH  
1365 forwarding buffer. Include it in future evaluations for possible message forwarding.  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374

1375 2.2. Evaluate if any message forwarding is currently allowed:

1376  
1377 2.2.1. For all buffered messages that are pending in CMDH message forwarding carry  
1378 out the following evaluation steps:

1379  
1380 2.2.1.1. Among all [cmdhNetworkAccessRules] child resources of the provisioned active  
1381 [cmdhPolicy] resource do the following selection:

1382  
1383 2.2.1.1.1. If present, select the [cmdhNetworkAccessRules] resource containing a  
1384 value in the list defined by the '*applicableEventCategory*' attribute that is  
1385 equal to the value of the '*ec*' parameter of the buffered message to be evaluated  
1386 for forwarding. If a match is found, continue processing with step 2.2.1.2.

1387  
1388 2.2.1.1.2. Select the [cmdhNetworkAccessRules] resource that contains the string  
1389 'default' in the list defined by the '*applicableEventCategory*'.

1390  
1391 2.2.1.2. Lookup all [cmdhNwAccessRule] child resources of the selected  
1392 [cmdhNetworkAccessRules] resource

1393  
1394 2.2.1.3. If the attribute '*otherConditions*' is present in any of the found  
1395 [cmdhNwAccessRule] resources, discard all [cmdhNwAccessRule] resources from  
1396 the list of found items for which the conditions expressed by '*otherConditions*' at  
1397 time of evaluation of the message for forwarding are not met, respectively.

1398  
1399 2.2.1.4. Among the all remaining found [cmdhNwAccessRule] resources find those  
1400 for which  
1401 - the <schedule> child resource 'allowedSchedule' is currently allowing  
1402 usage of the corresponding target network, and  
1403 - for which the corresponding target network could be used to reach the next  
1404 CSE for forwarding the message under evaluation.  
1405 If any allowed target network was found, memorize the message under  
1406 evaluation as an allowed message and the allowed target network(s) for the  
1407 message under evaluation and continue with the next evaluation of buffered  
1408 messages

1409  
1410 2.2.2. When all buffered messages have been evaluated, remove from the memorized list  
1411 of allowed messages and their allowed target networks those target networks where  
1412 the amount of data to be forwarded – accumulated over all allowed messages of the  
1413 same event category – is less than the amount of data indicated in the  
1414 '*minReqVolume*' attribute of the corresponding [cmdhNwAccessRule] resource.

1415  
1416 2.2.3. Remove any messages from the list of allowed messages for forwarding if no  
1417 allowed target network is left for that message after the previous step.

1418  
1419 2.3. Process messages allowed for forwarding to the next CSE:

1420 If any messages can be forwarded, i.e. if any evaluation of step 2.2 was positive, apply  
1421 the following steps:

1422  
1423 2.3.1. Reuse already established Mcc communication connections or – if needed –  
1424 establish new Mcc communication connection(s) so that all the messages that are  
1425 allowed to be forwarded to their next CSE can be forwarded. Some messages may  
1426 be allowed on the same target network. Follow the procedure outlined in clause

1427 H.2.5. for setting up a Mcc communication connection to another CSE via a  
1428 particular target network. If no usable Mcc communication connection could be  
1429 established for forwarding a particular allowed message before the message  
1430 expires, execute step 1.4 in this clause above for that message.

1431 1.

1432 2.3.2. For all messages allowed for forwarding and for which Mcc communication  
1433 connections are established, apply steps 1.3 through 1.5 in this clause above.

1434  
1435 2.4. Else, i.e. currently no message forwarding is allowed:

1436 End this cycle of CMDH message forwarding and wait for new triggering events.  
1437



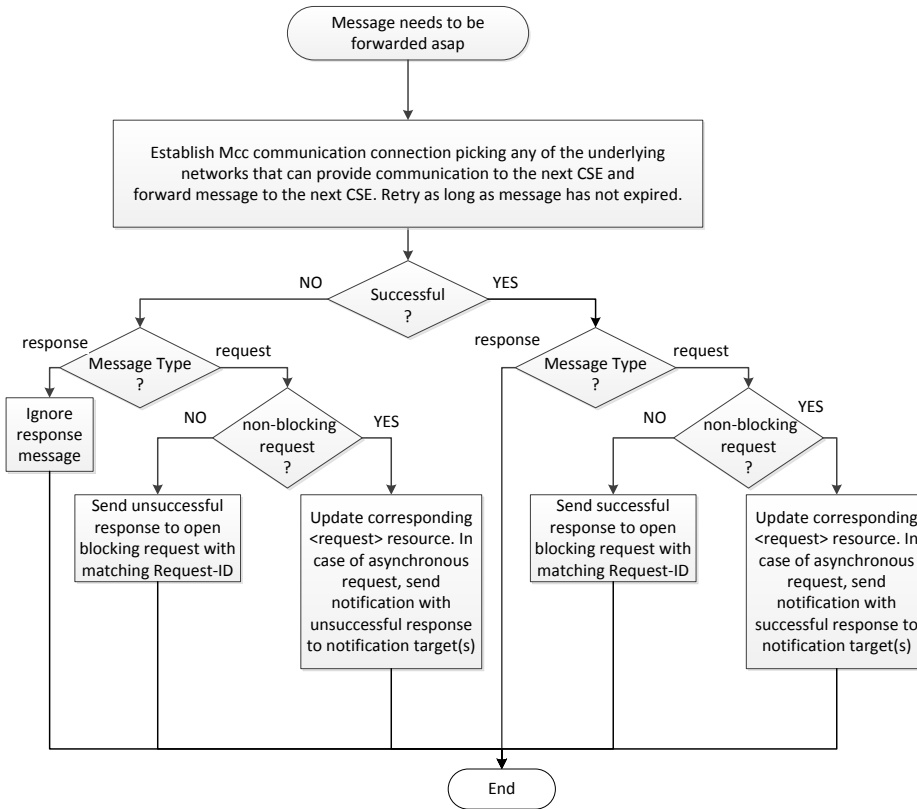
1438 When any of the underlying networks becomes usable for message forwarding due to transition(s) in allowed  
 1439 schedule(s) or due to establishing of availability of connectivity (e.g. cable plugged-in, coverage established), carry out  
 1440 the processing above in this clause starting with step 2.2.

1441

1442 When any of the underlying networks becomes unusable for message forwarding due to transition(s) in allowed  
 1443 schedule(s) or due to loss of availability of connectivity (e.g. cable unplugged, coverage lost), complete – if at all  
 1444 possible – any ongoing message forwarding procedures. End this cycle of CMDH message forwarding and wait for new  
 1445 triggering events.

1446

1447 When any message buffered for CMDH forwarding expires, carry out step 1.4 in this clause above. End this cycle of  
 1448 CMDH message forwarding and wait for new triggering events.



1449

1450 **Figure H.2.4-2: Forwarding of messages with 'ec' = 'immediate'.**

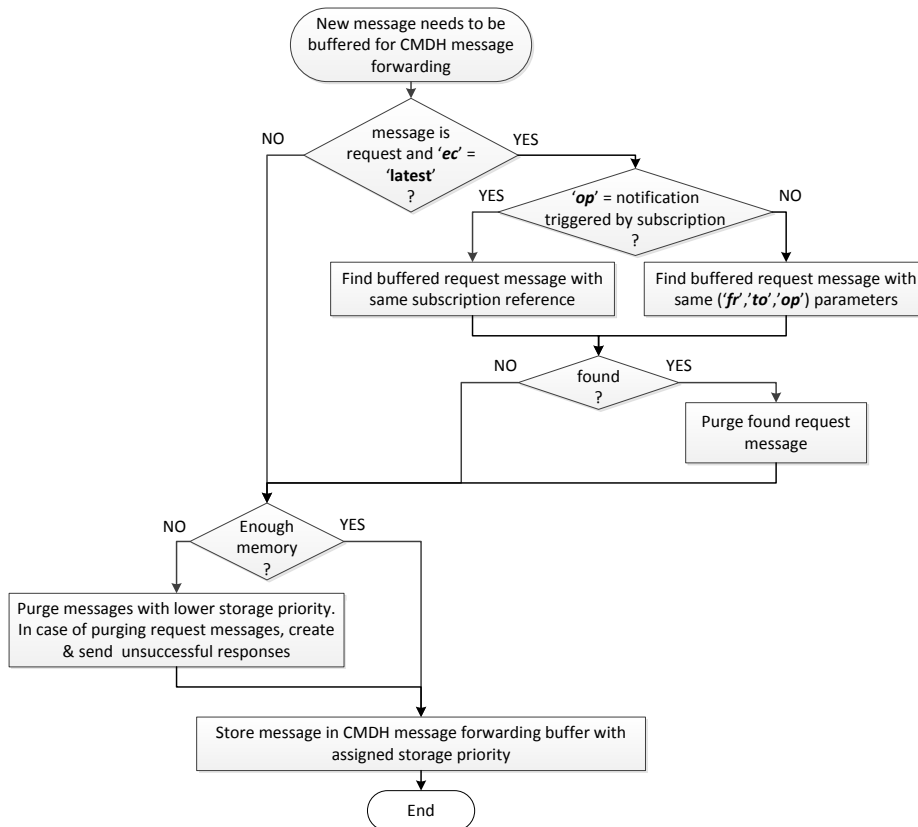


Figure H.2.4-3: Buffering of messages for CMDH message forwarding.

## 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.

1468 If establishing the Mcc communication connection via the selected target network fails, a new attempt to establish that  
1469 communication connection shall only be made after waiting for a back-off time according to the value given in the  
1470 'back-off time' component of the '**backOffParameters**' attribute of the effective [cmdhNwAccessRule] resource.

1471 When establishing the Mcc communication connection via the selected target network still fails, for each subsequent  
1472 new attempt to establish the Mcc communication connection without any successful attempts in-between, the back-off  
1473 time shall be increased by the value given in the 'back-off time increment' component of the '**backOffParameters**'  
1474 attribute of the effective [cmdhNwAccessRule] resource.

1475 The back-off time for waiting before making any new attempt to establish the Mcc communication connection via the  
1476 selected target network shall not exceed the value given by the 'maximum back-off time' component of the  
1477 '**backOffParameters**' attribute of the effective [cmdhNwAccessRule] resource.

1478 When the next CSE is hosted on a node for which a usable Mcc communication connection for forwarding a message to  
1479 the next CSE can only be established by the next CSE itself, device triggering mechanisms as defined in TS-0001 shall  
1480 be used.

1481 In case the next CSE can only be reached via communication connections originating from the node that hosts the next  
1482 CSE, while it is capable of processing incoming oneM2M messages, it is assumed that such a CSE establishes a polling  
1483 channel as defined in TS-0001 in order to effectively receive unsolicited oneM2M messages.

1484  
1485

## Annex I(informative): 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 ,N P	M ,O ,N P	O	N P		Some Resource specific use of <<common attribute name 1>> as text.
<<common attribute name 2>>			O	N P		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 ,N P	M ,O ,N P	O	N P		Some Resource specific use of <<common attribute name 1>> as text.
<<common attribute name 2>>			O	N P		Another Resource specific use of <<common attribute name2>>

1486

**Table 7.3.x.1-4: Child resources of <<resource name>>**

Child Resource Type	Child Resource Name	Multiplicity	Ref. to Resource Type Definition
<<resource type 1>>	[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 7.3.4

### 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*

1487

1488

---

## List of tables and figures

1489		
1490	Figure 5.3.1-1: Primitive overview .....	17
1491	Figure 5.3.2-1: Primitives modeling .....	18
1492	Table 6.2-1: M2M Identifiers.....	19
1493	Table 6.3.1-1: Simple Data Types incorporated from XML Schema.....	21
1494	Table 6.3.2-1: oneM2M Simple Data Types.....	23
1495	Table 6.3.3.1-1: Example of oneM2M Enumeration Type Definition .....	23
1496	Figure 6.3.3.2.1-1: Example of XSD version of oneM2M Enumeration Type.....	24
1497	Table 6.3.3.2.1-1: Interpretation of resourceType .....	24
1498	Table 6.3.3.2.2-1: Interpretation of commaList .....	25
1499	Table 6.3.3.2.3-1: Interpretation of cseTypeID.....	25
1500	Table 6.3.3.2.4-1: Interpretation of locationSource .....	25
1501	Table 6.3.3.2.5-1: Interpretation of eventCatType.....	25
1502	Table 6.3.3.2.6-1: Interpretation of operation .....	26
1503	Table 6.3.3.2.7-1: Interpretation of responseType.....	26
1504	Table 6.3.3.2.8-1: Interpretation of resultContent .....	26
1505	Table 6.3.3.2.9-1: Interpretation of discResType .....	26
1506	Table 6.3.3.2.10-1: Interpretation of statusCode.....	27
1507	Table 6.3.3.2.11-1: Interpretation of requestStatus.....	27
1508	Table 6.3.3.2.12-1: Interpretation of operationResult .....	27
1509	Table 6.3.3.2.13-1: Interpretation of memberType.....	28
1510	Table 6.3.3.2.14-1: Interpretation of consistencyStrategy.....	28
1511	Table 6.3.3.2.15-1: Interpretation of cmdType .....	28
1512	Table 6.3.3.2.16-1: Interpretation of execModetType.....	29
1513	Table 6.3.3.2.17-1: Interpretation of execStateType .....	29
1514	Table 6.3.3.2.19-1: Interpretation of pendingNotification .....	30
1515	Table 6.3.3.2.20-1: Interpretation of notificationContentType.....	30
1516	Table 6.3.3.2.21-1: Interpretation of resourceStatus.....	31
1517	Table 6.3.3.2.22-1: Interpretation of operation .....	31
1518	Table 6.3.3.2.23-1: Interpretation of attribute .....	31
1519	Table エラー! 指定したスタイルは使われていません。 -1: Interpretation of status .....	31
1520	Table 6.3.3.2.25-1: Interpretation of batteryStatus .....	32
1521	Table 6.3.3.2.26-1: Interpretation of mgmtDefinition .....	32

1522	Table 6.3.3.2.27-1: Interpretation of logTypeId .....	32
1523	Table 6.3.3.2.28-1: Interpretation of logStatus .....	33
1524	Table 6.3.5-1: Resource Common Attributes .....	40
1525	Table 6.4.1-2: Data Types for filterCriteria parameter .....	42
1526	Table 6.5-1: oneM2M Structured Data Types .....	43
1527	Table 6.6.2-1 RSC Framework Overview .....	45
1528	Table 6.6.3.2-1: RSCs for Un-successful Responses for Release 1 .....	46
1529	Table 6.6.3.3-1: RSCs for Acknowledgement Responses for Release 1 .....	47
1530	Table 7.2.1.1-1: Request Primitive Parameters .....	49
1531	Table 7.2.1.1-2: Response Primitive Parameters (1/2) .....	50
1532	Table 7.2.1.1-3 : Response Primitive Parameters (2/2) .....	50
1533	Figure 7.2.1.2.1-1: Generic procedure of Originator .....	51
1534	Figure 7.2.1.2.2-1: Generic procedure of Receiver .....	52
1535	Table 7.2.2.1.4-1: Request parameter settings .....	55
1536	Table 7.2.2.2.4-1: Common attributes settings for <request> resource .....	56
1537	Table 7.2.2.2.4-2: Resource-specific attributes settings for <request> resource .....	56
1538	Table 7.2.2.2.5-1: Response parameter settings .....	56
1539	Table 7.2.2.2.7-1: Common attributes settings for <request> resource .....	57
1540	Table 7.2.2.2.7-2: Resource-specific attributes settings for <request> resource .....	57
1541	Table 7.3.1.1-1: Data Type Definition of accessControlPolicy .....	65
1542	Table 7.3.1.1-2: Applicable Common Attributes on accessControlPolicy .....	65
1543	Table 7.3.1.1-3: Data Types for resource specific attributes .....	65
1544	Table 7.3.1.1-4: Child resources of accessControlPolicy .....	65
1545	Table 7.3.2.1-1: Data Type Definition of <CSEBase> .....	66
1546	Table 7.3.2.1-2: Common Attributes on <CSEBase> .....	66
1547	Table 7.3.2.1-3: Resource specific attributes on <CSEBase> .....	67
1548	Table 7.3.2.1-4: Reference to child resources of <CSEBase> .....	67
1549	Table 7.3.3.1-1: Data Type Definition of <remoteCSE> .....	68
1550	Table 7.3.3.1-2: Common Attributes on <remoteCSE> .....	69
1551	Table 7.3.3.1-3: Resource specific attributes on <remoteCSE> .....	69
1552	Table 7.3.3.1-4: Reference to child resources of <remoteCSE> .....	69
1553	Table 7.3.4.1-1: Data Type Definition of AE .....	70
1554	Table 7.3.4.1-2: Applicable Common Attributes on AE .....	70
1555	Table 7.3.4.1-3: Data Types for resource specific attributes .....	71
1556	Table 7.3.4.1-4: Child resources of AE .....	71

1557	Table 7.3.7.1-1: Data Type Definition of subscription.....	75
1558	Table 7.3.7.1-2: Applicable Common Attributes on <subscription> .....	76
1559	Table 7.3.7.1-3: Data Types for resource specific attributes.....	76
1560	Table 7.3.7.1-4: Reference of child resources.....	76
1561	Table 7.3.8.1-1: Data Type Definition of <schedule>.....	78
1562	Table 7.3.8.1-4: Extended Crontab Data String structure .....	79
1563	Table 7.3.9.1-1: Data Type Definition of locationPolicy .....	81
1564	Table 7.3.10.1-1: Data Type Definition of delivery .....	84
1565	Table 7.3.10.1-2: Common Attributes on delivery .....	84
1566	Table 7.3.10.1-3: Resource Specific Attributes on delivery .....	84
1567	Table 7.3.10.1-4: Child resources for delivery.....	84
1568	Table 7.3.11.1-1: Data Type Definition of request.....	86
1569	Table 7.3.11.1-4 : Reference of child resources.....	87
1570	Table 7.3.12.1-1: Data Type Definition of <group>.....	89
1571	Table 7.3.12.1-2: Common Attributes on <group> .....	89
1572	Table 7.3.12.1-3: Resource Specific Attributes on <group>.....	89
1573	Table 7.3.12.1-4: Reference of child resources.....	90
1574	Table 7.3.14.1-1: Data Type Definition of <mgmtObj> .....	94
1575	Table 7.3.14.1-2: Common Attributes on <mgmtObj>.....	95
1576	Table 7.3.14.1-3: Resource Specific Attributes on <mgmtObj> .....	95
1577	Table 7.3.14.1-4: Child resources of <mgmtObj>.....	95
1578	Table 7.3.15.1-1: Data Type Definition of <mgmtCmd> .....	97
1579	Table 7.3.15.1-2: Common Attributes on <mgmtCmd> .....	97
1580	Table 7.3.15.1-3: Resource Specific Attributes on <mgmtCmd>.....	98
1581	Table 7.3.15.1-4: Child resources of <mgmtCmd> .....	98
1582	Table 7.3.16.1-1: Data Type Definition of <execInstance>.....	101
1583	Table 7.3.16.1-2: Data Types for resource specific attributes.....	101
1584	Table 7.3.17.1-1: Data Type Definition of <node> .....	103
1585	Table 7.3.17.1-2: Common Attributes on <node> .....	103
1586	Table 7.3.17.1-3: Resource Specific Attributes on <node> .....	103
1587	Table 7.3.17.1-4: Child resources of <node>.....	103
1588	Table 7.3.18.1-1: Data Type Definition of <m2mServiceSubscriptionProfile>.....	104
1589	Table 7.3.18.1-2: Common Attributes on <m2mServiceSubscriptionProfile>.....	104
1590	Table 7.3.18.1-3: Resource Specific Attributes on <m2mServiceSubscriptionProfile>.....	104
1591	Table 7.3.18.1-4: Child resources of <authorizedNodeProfile>.....	104



1592	Table 7.3.19.1-1: Data Type Definition of <authorizedNode>.....	106
1593	Table 7.3.19.1-2: Common Attributes on <authorizedNode > .....	106
1594	Table 7.3.19.1-3: Resource Specific Attributes on <authorizedNode >.....	106
1595	Table 7.3.19.1-4: Child resources of <authorizedNode> .....	106
1596	Table 7.3.20.1-1: Data Type Definition of pollingChannel .....	107
1597	Table 7.3.20.1-2: Reference of child resources.....	107
1598	Table 7.3.22.1-1: Data Type definition of <statsConfig> .....	109
1599	Table 7.3.22.1-2: Common Attributes on <statsConfig> .....	109
1600	Table 7.3.22.1-4: Child resources of <statsConfig>.....	109
1601	Table 7.3.23.1-1: Data Type definition of <eventConfig>.....	110
1602	Table 7.3.23.1-2: Resource-specific attributes of <eventConfig> .....	110
1603	Table 7.3.23.1-3: Reference of child resources of <eventConfig>.....	111
1604	Table 7.3.24.1-1: Data Type definition of <statsCollect>.....	111
1605	Table 7.3.24.1-2: Resource-specific attributes of <statsCollect> .....	112
1606	Table 7.3.24.1-3: Reference of child resources of <statsCollect>.....	112
1607	Table 7.3.25.1-1: Data Type Definition of Announced Resource .....	113
1608	Table 7.3.X.1-2: Applicable Common Attributes on Announced Resource.....	113
1609	Table A-1: Resource attributes.....	128
1610	Table D.2-1: Data Type Definition of [firmware] .....	138
1611	Table D.2-2: Data Types for resource specific attributes .....	138
1612	Table D.3-1: Data Type Definition of [software] .....	139
1613	Table D.3-2: Data Types for resource specific attributes .....	140
1614	Table D.4-1: Data Type Definition of [memory].....	141
1615	Table D.4-2: Data Types for resource specific attributes .....	141
1616	Table D.5-1: Data Type Definition of [areaNwkInfo].....	142
1617	Table D.5-2: Data Types for resource specific attributes .....	142
1618	Table D.6-1: Data Type Definition of [areaNwkDeviceInfo].....	143
1619	Table D.6-2: Data Types for resource specific attributes .....	144
1620	Table D.7-1: Data Type Definition of [battery] .....	145
1621	Table D.7-2: Data Types for resource specific attributes .....	145
1622	Table D.8-1: Data Type Definition of [deviceInfo].....	146
1623	Table D.8-2: Data Types for resource specific attributes .....	146
1624	Table D.9-1: Data Type Definition of [deviceCapability] .....	148
1625	Table D.9-2: Data Types for resource specific attributes .....	148
1626	Table D.10-1: Data Type Definition of [reboot].....	149

1627 Table D.10-2: Data Types for resource specific attributes ..... 149

1628 Table D.11-1: Data Type Definition of [eventLog] ..... 150

1629 Table D.11-2: Data Types for resource specific attributes ..... 151

1630 Table D.12-1: Data Type Definition of [cmdhPolicy] ..... 152

1631 Table D.12-2: Data Types for resource specific attributes ..... 152

1632 Table D.12.1-1: Data Type Definition of [activeCmdhPolicy] ..... 152

1633 Table D.12.1-2: Data Types for resource specific attributes ..... 153

1634 Table D.12.2-1: Data Type Definition of [cmdhDefaults] ..... 153

1635 Table D.12.2-2: Data Types for resource specific attributes ..... 153

1636 Table D.12.3-1: Data Type Definition of [cmdhDefEcValue] ..... 153

1637 Table D.12.3-2: Data Types for resource specific attributes ..... 154

1638 Table D.12.4-1: Data Type Definition of [cmdhEcDefParamValues] ..... 154

1639 Table D.12.4-2: Data Types for resource specific attributes ..... 154

1640 Table D.12.5-1: Data Type Definition of [cmdhLimits] ..... 154

1641 Table D.12.5-2: Data Types for resource specific attributes ..... 155

1642 Table D.12.6-1: Type Definition of [cmdhNetworkAccessRules] ..... 155

1643 Table D.12.6-2: Data Types for resource specific attributes ..... 155

1644 Table D.12.7-1: Data Type Definition of [cmdhNwAccessRule] ..... 155

1645 Table D.12.7-2: Data Types for resource specific attributes ..... 156

1646 Table D.12.8-1: Data Type Definition of [cmdhBuffer] ..... 156

1647 Table D.12.8-2: Data Types for resource specific attributes ..... 156

1648 Table G.1.2-3: Applicable NetAPI for Terminal Location ..... 165

1649 Table G.1.2-4: Resource Type Definition – TerminalLocation ..... 166

1650 Table G.1.2-5: Resource Type Definition – PeriodicNotificationSubscription ..... 166

1651 Table 0-6: Resource Type Definition – CircleNotificationSubscription ..... 166

1652 Figure G.1.3.1-1: Single Query Toward Location Server ..... 167

1653 Figure G.1.4-1: Subscribe to Notification for Periodic Location Updates ..... 168

1654 Figure H.2.4-1: (Title is TBD) ..... 179

1655 Figure H.2.4-2: Forwarding of messages with 'ec' = 'immediate'. ..... 185

1656 Figure H.2.4-3: Buffering of messages for CMDH message forwarding. .... 186

1657 Table 7.3.x.1-1: Data Type Definition of <<resource name>> ..... 188

1658 Table 7.3.x.1-2: Common Attributes on <<resource name>> ..... 188

1659 Table 7.3.x.1-3: Resource Specific Attributes on <<resource name>> ..... 188

1660 Table 7.3.x.1-4: Child resources of <<resource name>> ..... 189

1661

1662

## History

<b>Draft history</b> (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 contribution 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	<p>Incorporated agreed contributions at PRO #11.0:</p> <p>Ref: PRO-2014-195R03- Child Resource Conversion Rules</p> <p>Ref: PRO-2014-216R02- Response Status Clause</p> <p>Ref: PRO-2014-217R03- Attribute Creation and Deletion</p> <p>Ref: PRO-2014-220R02- management-common-operations</p> <p>Ref: PRO-2014-229R03- Location Request on Mcn Reference Point</p> <p>Ref: PRO-2014-242- Correction to Table 7.2.1.1-2 in TS-0004</p> <p>Ref: PRO-2014-219R02- Resource Discovery Procedure</p> <p>Ref: PRO-2014-224R01- design_principles_scalability</p> <p>Ref: PRO-2014-218R05- Announcement Procedures</p> <p>Ref: PRO-2014-249R03- Resource Type Definition Template Update</p> <p>Ref: PRO-2014-251R01- remoteCSE Resource Type Definition</p> <p>Ref: PRO-2014-236R01-Stage 3 text for Resource Type &lt;statsConfig&gt;</p> <p>Ref: PRO-2014-237R02-Stage 3 text for Resource Type &lt;eventConfig&gt;</p> <p>Ref: PRO-2014-238R02-Stage 3 text for Resource Type &lt;statsCollect&gt;</p> <p>Incorporated agreed contribution at PRO #11 (by Nobu U)</p> <p>Ref: PRO-2014-0152R02-group-management-procedures</p> <p>Ref: PRO-2014-0194R04-CSEBase_Resource_Type_Definition</p> <p>Ref: PRO-2014-0221R03-mgmtObj-management-procedures</p> <p>Ref: PRO-2014-0228R01- LocationPolicy_Resource_Primitive</p> <p>Ref: PRO-2014-0230R03- mgmtCmd_input_TS0004</p> <p>Ref: PRO-2014-0231R04-Request_resource_type_definition</p> <p>Ref: PRO-2014-0235R02- pollingChannel_procedures</p> <p>Incorporated PRO-2014-0239R03-Resource_Type_XSD_example in Annex with Editor's changes</p> <p>Incorporated agreed contributions (missing in R01)</p> <p>Ref: PRO-2014-223R01-<a href="#">node-resource-type-procedures</a></p> <p>Ref: PRO-2014-227R02-<a href="#">cmdh_policies_TS0004</a></p>
v.0.5.1	17 Jun 2014	<p>Incorporated pended Agreed contribution:</p> <p>Ref: PRO-2014-222R02-Primitive_templates_and_procedure_outlines_modification</p> <p>Removed resolved Editor's Notes and unused sub-clauseses.</p>

V0.5.2	25 Jun 2014	<p>Clean-up done by <a href="mailto:edithelp@etsi.org">editHelp!</a>  e-mail: <a href="mailto:edithelp@etsi.org">mailto:edithelp@etsi.org</a></p> <p>Incorporated agreed contribution at PRO #11.1 (by Nobu U)  Ref: PRO-2014-0252R02-XML_Schema_Further_Changes  Ref: PRO-2014-0262R01-schedule</p> <p>Fixed reference link between section 1.1 and section 7.3.8</p> <ul style="list-style-type: none"> <li>• PRO-2014-0228R01- LocationPolicy_Resource_Primitive</li> </ul> <p>Added missing annex</p> <ul style="list-style-type: none"> <li>• PRO-2014-229R03- Location Request on Mcn Reference Point</li> </ul> <p>Incorporated agreed contribution at PRO #11.2:  Ref: PRO-2014-0274- TS0004 Abbreviations</p>
V0.5.3	4 Jul 2014	<p>Incorporated agreed contribution at PRO #11.3 (by Shingo):  Ref: PRO-2014-0276R01-additional-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</p>

V0.5.4	28 Jul 2014	<p>Incorporated agreed contributions before PRO #12 (by Shingo)</p> <p>Ref:</p> <p>Ref:PRO-2014-0268R04-Add_common_attributes_on_ResType_template</p> <p>Ref:PRO-2014-0272R02-Messages_parameters_data_types_clause</p> <p>Ref:PRO-2014-0286R03-Container_and_Container_Instance_ResType</p> <p>Ref:PRO-2014-0287R02-TS-0004_Section_7_3_Cleanup</p> <p>Ref:PRO-2014-0298R02-Generic_Procedure_Clean_Up</p> <p>Ref:PRO-2014-0304R03-data_types (this has NOT been incorporated.)</p> <p>Ref:PRO-2014-0308R06-subscription_definition_and_procedures</p> <p>Ref:PRO-2014-0320R02-Request-related_common_procedures</p> <p>Ref:PRO-2014-0324R02-notification_definition_and_procedures</p> <p>Ref:PRO-2014-0326R02-procedure-for-fanOutPoint</p> <p>Ref:PRO-2014-0328-Request_resource_type_definition_-_Update</p> <p>Ref:PRO-2014-0331R01-Operation_Applicability_Reference</p> <p>Ref:PRO-2014-0334R04-New_section_for_services_provided_by_the_underlying_network.</p> <p>Ref:PRO-2014-0335R02-Clarify_M2M-Ext-ID</p> <p>Ref:PRO-2014-0336R01-Clarify_Device_Triggering</p> <p>Ref:PRO-2014-0337R01-Notification_procedure_for_subscription</p> <p>Ref:PRO-2014-0342R01-Clarification_on_entrance_for_resource_announcement</p> <p>Ref:PRO-2014-0347R01-Announced_Resource_Procedures</p> <p>Ref:PRO-2014-0355R01-Corrections_of_locationPolicy_Resource</p> <p>Ref:PRO-2014-0373R01-Res_Def_Template_Update</p>
--------	-------------	--

V0.5.5	29 Jul 2014	<p>Incorporated agreed contributions on July 29 at PRO #12 (by Nobu)</p> <p>PRO-2014-0309R03-Notify_Request_Re-targeting</p> <p>PRO-2014-0310R02-AE_Resource_Type_and_Procedure</p> <p>PRO-2014-0311R03-TS-0004_Cleanup</p> <p>PRO-2014-0312R01-accessControlPolicy_Resource_Type_and_Procedure</p> <p>PRO-2014-0356R02-Area-based_notification_Service_of_OMA_Location_API</p> <p>PRO-2014-0357R01-Correction_of_the_parameters_at_the_Device_Triggering_commands</p> <p>PRO-2014-0364-remoteCSE_XSD</p> <p>PRO-2014-0366R02-Extensibility_issues</p> <p>PRO-2014-0375R02-oneM2M_enumeration_types</p> <p>PRO-2014-0380R03-Correction_of_notification_procedure</p> <p>PRO-2014-0381R01-Resource_Discovery_Procedure</p> <p>PRO-2014-0383R01-Clarify_the_Trigger-Recipient-ID</p> <p>PRO-2014-0385R01-restructuring_notification_text</p> <p>PRO-2014-0389-Request_Applicability_on_Attributes</p>
V0.5.6	30 Jul 2014	<p>Incorporated agreed contributions on July 30 at PRO #12 (by Nobu)</p> <p>PRO-2014-0315R08-TS-0004_Annex_D_Cleanup</p> <p>PRO-2014-0327R07-resources-for-mgmtObj</p> <p>PRO-2014-0329R01-Delivery_resource_type_definition_-_Update</p> <p>PRO-2014-0340R02-Resource_definition_of_AreaNwkInfo_and_AreaNwkDeviceInfo</p> <p>PRO-2014-0343R04-Synchronization_of_announced_attribute</p> <p>PRO-2014-0345R02-devInfo_eventLog_Management_Resource_Procedures</p> <p>PRO-2014-0346R04-TS-0004-Mgmt_obj_common_operations_updates</p> <p>PRO-2014-0348R03-mgmtCmd_update_TS0004</p> <p>PRO-2014-0367R04-Procedures_for_accessing_resources</p> <p>PRO-2014-0388-Enumeration_Data_Type_Definitions</p> <p>PRO-2014-0390-cleaning-for-the-new-template</p> <p>PRO-2014-0391R01-response-status-code-cleaning</p> <p>PRO-2014-0394R02-Cleaning_CSEBase_resource_for_the_new_template</p> <p>PRO-2014-0395R02-Cleaning_remoteCSE_resource_for_the_new_template</p> <p>PRO-2014-0397R01-7_3_Cleanup_for_subscription_pollingChannel_pollingChannelURI</p> <p>PRO-2014-0398R01-Clean_Up_of_locationPolicy_Resource</p>

V0.6.0	01 Aug 2014	<p>Incorporated agreed contributions on July 31<sup>st</sup> at PRO #12 (by Shingo)</p> <p>PRO-2014-0314R02-MIME_type_for_oneM2M_resource_representation</p> <p>PRO-2014-0372R02-Status_Code_Cleanup</p> <p>PRO-2014-0374R03-CMDH_Procedures</p> <p>PRO-2014-0392R03-_schedule_resource_default_text</p> <p>PRO-2014-0399R01-_container_contentInstance_update</p> <p>PRO-2014-0401R02-clean_up_of_announced_resource_type</p> <p>PRO-2014-0403R01-Response_Status_Update</p> <p>PRO-2014-0404R01-authorizedNode_Resource_Type</p> <p>PRO-2014-0405R01-m2mServiceSubscriptionProfile_Resource_Type</p> <p>PRO-2014-0406-Management_common_operations_and_status_codes_updates</p> <p>And, add correction missing implementation of PRO-2014-0304R03</p>
V0.6.1	01 Aug 2014	Prepared for Initial Release. Same content of V0.6.0 (published as Initial Release)
V0.6.2	09 Sep 2014	<p>Correction on editorial errors and notation for attribute/parameter on previous version.</p> <p>Incorporated agreed contribution as of September 6<sup>th</sup> (by Shingo)</p> <p>PRO-2014-0418-CR_notificationEvent_data_type</p>
V0.7.0	26 Sep 2014	<p>Incorporated agreed contributions as of September 24<sup>th</sup> (by Shingo):</p> <p>PRO-2014-0436R03-TS-0004_Annex_F_Cleanup</p> <p>PRO-2014-0443R03-Complex_Type_Definitions_for_MIME_Types</p> <p>PRO-2014-0444R04-Restructuring_Common_Data_Types</p> <p>PRO-2014-0448R02-CR_TS-0004_statsConfig</p> <p>PRO-2014-0464R02-CR_TS-0004_container</p> <p>PRO-2014-0465R03-CR_TS-0004_contentInstance</p> <p>PRO-2014-0467R02-CR_TS-0004_node</p> <p>PRO-2014-0468R02-CR_TS-0004_remoteCSE</p> <p>PRO-2014-0469R01-CR_TS-0004_subscription</p> <p>PRO-2014-0476R01-aggregate-notification</p> <p>PRO-2014-0477R03-procedure-for-service-layer-managements</p> <p>PRO-2014-0483R02-TS-0004_updates_on_ac_and_cmdh_policies</p> <p>PRO-2014-0488R02-TS-0004_sec_7_3_15_cleanup</p> <p>PRO-2014-0507R01-TS-0004_sec_6_3_2_2_update</p> <p>PRO-2014-0508R04-Short_Names_tables</p> <p>PRO-2014-0509R01-Event_Category_in_Notification_Procedure</p>

1664