

I n t e r n a t i o n a l T e l e c o m m u n i c a t i o n U n i o n

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Q.819

(01/2022)

SERIES Q: SWITCHING AND SIGNALLING, AND
ASSOCIATED MEASUREMENTS AND TESTS

Q3 interface

REST-based management services

Recommendation ITU-T Q.819

ITU-T

ITU-T Q-SERIES RECOMMENDATIONS
SWITCHING AND SIGNALLING, AND ASSOCIATED MEASUREMENTS AND TESTS

SIGNALLING IN THE INTERNATIONAL MANUAL SERVICE	Q.1–Q.3
INTERNATIONAL AUTOMATIC AND SEMI-AUTOMATIC WORKING	Q.4–Q.59
FUNCTIONS AND INFORMATION FLOWS FOR SERVICES IN THE ISDN	Q.60–Q.99
CLAUSES APPLICABLE TO ITU-T STANDARD SYSTEMS	Q.100–Q.119
SPECIFICATIONS OF SIGNALLING SYSTEMS No. 4, 5, 6, R1 AND R2	Q.120–Q.499
DIGITAL EXCHANGES	Q.500–Q.599
INTERWORKING OF SIGNALLING SYSTEMS	Q.600–Q.699
SPECIFICATIONS OF SIGNALLING SYSTEM No. 7	Q.700–Q.799
Q3 INTERFACE	Q.800–Q.849
DIGITAL SUBSCRIBER SIGNALLING SYSTEM No. 1	Q.850–Q.999
PUBLIC LAND MOBILE NETWORK	Q.1000–Q.1099
INTERWORKING WITH SATELLITE MOBILE SYSTEMS	Q.1100–Q.1199
INTELLIGENT NETWORK	Q.1200–Q.1699
SIGNALLING REQUIREMENTS AND PROTOCOLS FOR IMT-2000	Q.1700–Q.1799
SPECIFICATIONS OF SIGNALLING RELATED TO BEARER INDEPENDENT CALL CONTROL (BICC)	Q.1900–Q.1999
BROADBAND ISDN	Q.2000–Q.2999
SIGNALLING REQUIREMENTS AND PROTOCOLS FOR THE NGN	Q.3000–Q.3709
SIGNALLING REQUIREMENTS AND PROTOCOLS FOR SDN	Q.3710–Q.3899
TESTING SPECIFICATIONS	Q.3900–Q.4099
PROTOCOLS AND SIGNALLING FOR PEER-TO-PEER COMMUNICATIONS	Q.4100–Q.4139
SIGNALLING REQUIREMENTS AND PROTOCOLS FOR IMT-2020	Q.5000–Q.5049
COMBATING COUNTERFEITING AND STOLEN ICT DEVICES	Q.5050–Q.5069

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

Recommendation ITU-T Q.819

REST-based management services

Summary

Recommendation ITU-T Q.819 defines a set of services required to support REST-based interfaces and, with Recommendation ITU-T X.785, composes a framework for REST-based network management interfaces. It specifies protocol requirements and defines some network management-specific support services, which are notification service, heartbeat service and containment service. The JSON/YAML interface definitions for the network management-specific support services are also provided.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Q.819	2022-01-13	2	11.1002/1000/14835

Keywords

5G, AI, energy saving, telecom operation management.

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

FOREWORD

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

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

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

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

NOTE

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

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

INTELLECTUAL PROPERTY RIGHTS

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

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

© ITU 2022

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

Table of Contents

	Page
1 Scope.....	1
2 References.....	1
3 Definitions	2
3.1 Terms defined elsewhere	2
3.2 Terms defined in this Recommendation.....	2
4 Abbreviations and acronyms	2
5 Conventions	3
6 REST-based network management services – goals and requirements.....	4
6.1 Goals.....	4
6.2 Information modelling dependencies	4
6.3 Notifications	5
7 Framework overview and protocol requirements	5
7.1 Framework overview.....	5
7.2 Protocol, languages and services requirements	6
8 REST-based notification service and notification format	7
8.1 REST-based notification service overview	7
8.2 Mapping of operations from [ITU-T M.3702]	7
8.3 Notification format definitions	9
9 REST-based heartbeat service	15
10 REST-based containment service	17
11 Compliance and conformance	20
11.1 System conformance points.....	20
11.2 Basic conformance profile.....	20
Annex A – YAML/JSON scheme definition of framework support services	21
A.1 REST-based notification service interface	21
A.2 REST-based heartbeat service interface	36
A.3 REST-based containment service interface	41
Bibliography.....	46

Recommendation ITU-T Q.819

REST-based management services

1 Scope

The network management architecture defined in [ITU-T M.3010] introduces the use of multiple management protocols. So far, GDMO/CMIP, CORBA GIOP/IIOP, SMI/SNMP and Web Service/SOAP are possible choices at the application layer. Based on the management interface specification methodology defined in [ITU-T M.3020], more technology-based paradigms can be introduced into network management interfaces, and representational state transfer/hyper text transfer protocol (REST/HTTP) is now an additional paradigm for network management.

This Recommendation, together with [ITU-T X.785], sets out to define a framework for defining how interfaces supported by management systems and network elements should be modelled using the REST / JavaScript object notation (JSON) schema. It is within the scope of this Recommendation to provide the following guidelines or instructions:

- protocol requirements for using REST in network management;
- how a REST-based notification service is used in network management interfaces;
- how to monitor the availability of a REST notification forwarding mechanism;
- how to access containment information through a REST containment service;
- compliance and conformance requirements.

2 References

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

- [ITU-T M.3010] Recommendation ITU-T M.3010 (2000), *Principles for a telecommunications management network*.
- [ITU-T M.3020] Recommendation ITU-T M.3020 (2017), *Management interface specification methodology*.
- [ITU-T M.3702] Recommendation ITU-T M.3702 (2010), *Common management services – Notification management – Protocol neutral requirement and analysis*.
- [ITU-T X.701] Recommendation ITU-T X.701 (1997), *Information technology – Open Systems Interconnection – Systems management overview*.
- [ITU-T X.703] Recommendation ITU-T X.703 (1997), *Information technology – Open Distributed Management Architecture*.
- [ITU-T X.721] Recommendation ITU-T X.721 (1992), *Information technology – Open Systems Interconnection – Structure of management information: Definition of management information*.
- [ITU-T X.785] Recommendation ITU-T X.785 (2021), *Guidelines for defining REST-based managed objects and management interfaces*.
- [IETF RFC 3986] IETF RFC 3986 (2005), *Uniform Resource Identifier (URI): Generic Syntax*.

- [IETF RFC 6585] IETF RFC 6585 (2012), *Additional HTTP Status Codes*.
- [IETF RFC 7230] IETF RFC 7230 (2014), *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing*.
- [IETF RFC 7231] IETF RFC 7231 (2014), *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*.
- [IETF RFC 7232] IETF RFC 7232 (2014), *Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests*.
- [IETF RFC 7540] IETF RFC 7540 (2015), *Hypertext Transfer Protocol Version 2 (HTTP/2)*.
- [IETF RFC 8259] IETF RFC 8259 (2017), *The JavaScript Object Notation (JSON) Data Interchange Format (JSON)*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 agent [ITU-T M.3020]: Encapsulates a well-defined subset of management functionality. It interacts with managers using a management interface. From the manager's perspective, the agent behaviour is only visible via the management interface.

3.1.2 managed object class [ITU-T X.701]: A named set of managed objects sharing the same (named) sets of attributes, notifications, management operations (packages), and which share the same conditions for presence of those packages.

3.1.3 manager [ITU-T M.3020]: Models a user of agent(s) and it interacts directly with the agent(s) using management interfaces. Since the manager represents an agent user, it gives a clear picture of what the agent is supposed to do. From the agent perspective, the manager behaviour is only visible via the management interface.

3.1.4 notification [ITU-T X.703]: An interaction for which the contract between the invoking object (client) and the receiving object (server) is restricted to the ability of the server to receive the contents of information sent by the client.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

CMIP	Common Management Information Protocol
CORBA	Common Object Request Broker Architecture
CRUD	Create, Read, Update and Delete
DN	Distinguished Name
GDMO	Guidelines for the Definition of Managed Objects
GIOP	General Inter-ORB Protocol
HTTP	Hyper Text Transfer Protocol
IIOP	Internet Inter-ORB Protocol
JSON	JavaScript Object Notation

MO	Managed Object
REST	Representational State Transfer
SMI	Structure of Management Information
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
TMN	Telecommunications Management Network
URI	Uniform Resource Identifier
URL	Unified Resource Location
XML	extensible Markup Language
YAML	YAML Ain't Markup Language

5 Conventions

A few conventions are followed in this Recommendation to make the reader aware of the purpose of the text. Two qualifiers are used, which are:

M, which stands for mandatory and means that the attribute or field is mandatory for an operation or notification.

O, which stands for optional and means that the attribute or field is optional for an operation or notification.

Examples of YAML and JSON are included in this Recommendation and normative YAML or JSON schema examples specifying the data types, base classes and other modelling constructs of the framework are included in Annex A. The JSON/YAML schemas are written in a 10 point courier typeface:

A JSON schema example	<pre>{ "title": "root", "items": { "title": "array item" } }</pre>
A YAML schema example	<pre>SomeType: type: object properties: attr1: type: string attr2: type: string enum: - e1 - e2</pre>

6 REST-based network management services – goals and requirements

This clause describes the key goals of the services framework and the requirements that help the REST-based network management services support these goals. Clause 6.1 introduces the goals of the REST-based services framework. Clause 6.2 then provides terminology and requirements. The requirements in this clause are requirements that the framework must satisfy and that are based on telecommunications management needs. Clauses 7, 8 and 9 then describe a framework that meets these needs and defines how to achieve the requirements of this clause by using REST-based services in a certain way.

6.1 Goals

This Recommendation sets out a framework for defining how interfaces supported by management systems and network elements should be modelled. Some key goals of the framework are identified here:

- application interoperability;
- common REST-based support services.

This clause elaborates on these two goals.

6.1.1 Application interoperability

A key goal of the network management architecture, and in particular the information architecture, is to promote a standard framework for providing interoperability and information exchange between systems from a diverse set of network management system suppliers. Interoperability between systems involves many aspects of development. At its lowest layer, a common communication mechanism must be in place to support a common syntax, the establishment of connectivity and the exchange of operation requests/replies between systems. This aspect of interoperability is inherently supported by the REST/HTTP specifications.

For network management, there is a need to provide application interoperability. That is, management systems from diverse suppliers will be utilized within a single administration's telecommunications management network (TMN) to support the different functions necessary to support the management of its networks. To simplify the integration of these various suppliers' systems, they must agree on the semantics of the information being exchanged. This is accomplished with the specification of an information model. Guidelines for the definition of REST-based information models are specified in [ITU-T X.785], but the services defined here should also support those guidelines.

6.1.2 Common REST-based support services

A second aspect of this framework is the definition of the common usage and profiling of the distributed processing environment of choice. This aspect of the framework should indicate the reasonable expectations network management system suppliers may have of one another. Rather than redefining the interface capabilities needed to support common network management functions such as a notification service with each information model, the modelling guidelines in [ITU-T X.785] rely upon a set of support services. These support services enable the information models to be simpler, and also enhance interoperability.

6.2 Information modelling dependencies

As described in the previous clause, the explicit modelling of resources that are manageable across an interface is central to application interoperability. The guidelines for defining REST-based managed objects (MOs) detailed in [ITU-T X.785] describe the rules for modelling manageable resources. They also embody several decisions that must be supported by the network management REST-based services framework. This clause summarizes those points.

6.2.1 Access granularity

REST-based service interface granularity refers to the relationship between the resources that are modelled on an interface and the means by which they are accessed using REST-based services. The objects that represent manageable resources are called MOs. Recommendation [ITU-T X.785] uses a service-grain modelling approach which means each modelled resource is only accessible through either the generic MO access service, or a user defined REST service specific for various MO classes.

6.2.2 Representation of containment and naming

Containment is a logical representation of how modelled resources contain other modelled resources. Containment has traditionally been a very important relationship in network management applications because it is a convenient means of identifying the large number of resources that typically must be managed. The [ITU-T X.785] guidelines require that a unique name be assigned to each MO, based in part on the name of the object that contains it. The REST-based management services must provide the means to store these names (and hence the containment relationships they represent).

6.2.3 Object creation and deletion

The REST-based platforms do not provide clients with the particular means to create or delete objects on remote systems. Instead, these functionalities are provided by remote systems for clients to create or delete objects on the remote system. Recommendation [ITU-T X.785] provides a common object accessing service, so object creation and deletion can be model independent. When deleting an object that contains other objects, the containment relationship shall be retained, so that there will be no orphan objects existing in the remote systems. The containment information should be stored in the remote system and be maintained when creation and deletion take place.

6.3 Notifications

The framework needs to support the ability to:

- deliver notifications;
- subscribe for notification types;
- forward notifications to multiple destinations;
- uniquely identify the resource that emits the notification.

7 Framework overview and protocol requirements

7.1 Framework overview

This framework for REST-based management interfaces is a collection of capabilities. The framework defines the support services to be standards on network management interfaces conforming to this framework. The REST interfaces for these services will be defined in Annex A.

To support the software objects representing manageable resources, the framework requires that they implement some common basic capabilities. Therefore, one base class (YAML/JSON-based) is defined in [ITU-T X.785] for use in modelling network management resources. MO classes in information models must inherit and implement a basic set of capabilities from the base class in order to operate within this framework. Finally, some rules and conventions are defined for information modellers developing models for use with this framework. These consist of modelling guidelines and YAML/JSON style idioms. All of these are depicted graphically in Figure 1.

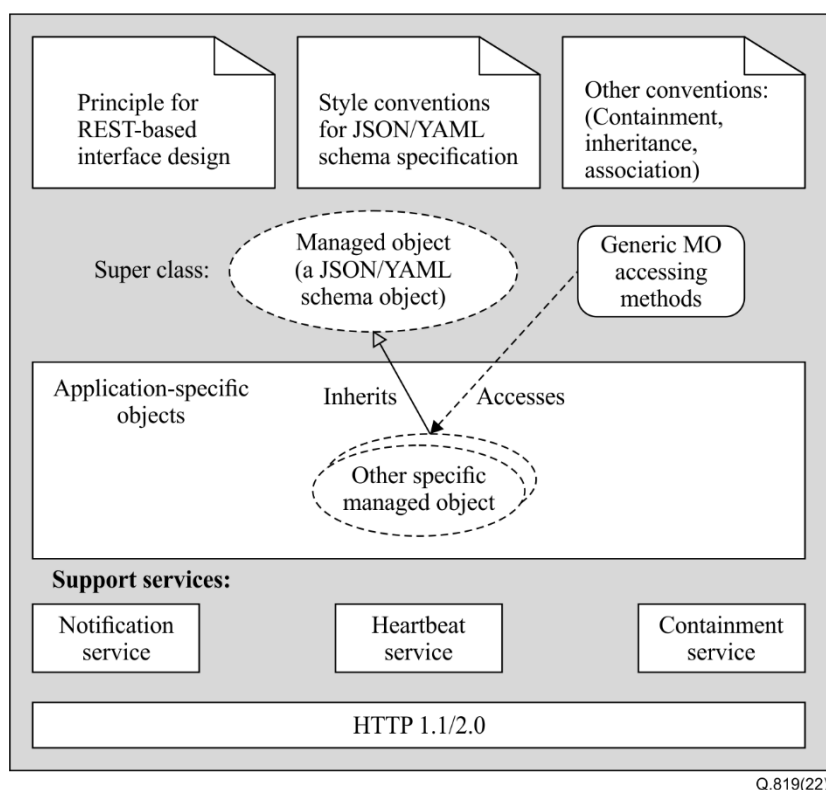


Figure 1 – Framework overview

Figure 1 shows the framework in grey. In the middle are the application-specific objects that are supported by the framework. Along the bottom is a box representing the communication protocol, HTTP 1.1 or 2.0. Above that are a number of boxes with names in them representing the services that compose this framework. Along the top of the figure are icons representing the super class ManagedObject, and the generic MO accessing methods. Each MO supported by this framework must ultimately inherit from the super class. Also shown in Figure 1 are squares with downturned corners representing standard object modelling conventions.

The framework services, represented as boxes with square corners, are defined in this Recommendation. The super class, object modelling conventions and other conversions are defined in [ITU-T X.785].

7.2 Protocol, languages and services requirements

This clause defines the versions of the languages and protocols that are required to support this framework. REST-based technology and protocol specifications are defined by IETF and OAI. Table 1 shows which version (including subsequent versions released up until the publication date of this Recommendation) of the applicable specifications must be supported to comply with this framework.

Table 1 – REST-based language, protocol and versions

Language and protocol	Version	References
YAML Schema (OAI)	3.0	[b-OAI-OAS3]
JSON Schema Core (IETF)	Draft 2020-12 (Draft-08-patch-1, December, 2020)	[IETF RFC 8259]
HTTP protocol (IETF)	1.1 (2014) 2.0 (August, 2017)	[IETF RFC 7230], [IETF RFC 7231],[IETF RFC 7232], [IETF RFC 7540]

8 REST-based notification service and notification format

8.1 REST-based notification service overview

REST-base notification service provides a standard approach for notification consumers and providers to transfer notification information using a topic-based publish/subscribe pattern. The contents of the notification service include standard message exchanges to be implemented by notification providers, operational requirements expected of service providers and requesters that participate in notifications and an JSON model that describes topics.

The REST-based notification service supports the asynchronous exchange of event messages between clients using a subscribe-and-publish paradigm.

8.2 Mapping of operations from [ITU-T M.3702]

Recommendation [ITU-T M.3702] provides protocol neutral requirements and analysis for the generic notification management service, which defines the generic functions that are to be implemented by a protocol-specific management paradigm. In this framework, the generic notification management model is mapped to the JSON/HTTP definition.

Table 2 indicates the mapping from [ITU-T M.3702] to the REST-based notification service.

Table 2 – Notification operations mapping from [ITU-T M.3702] to this Recommendation

No.	[ITU-T M.3702]		[ITU-T Q.819]	
	Operation name		Operation name	
	Request/ response	Parameters	Request/ response	Parameters
1	subscribeNotification		HTTP POST	
	Input parameter	– managerId – notificationTypes – filteringCriteria – destination	Request	"POST" {notificationServiceURI}/"subscriptions" {HTTPVersion} request body schema: \$ref: "notificationApiDataSchema.json#/definitions/ SubscribeNotificationRequest"
	Output parameter	– subscriptionId – status	Response	{HTTPVersion} {statusCode} {reasonPhrase} Location: {notificationServiceURI}/"subscriptions"/ {subscriptionId}{response body}
2	unsubscribeNotification		HTTP DELETE	
	Input parameter	– managerId – subscriptionId	Request	"DELETE" {notificationServiceURI}/"subscriptions"/ {subscriptionId} {HTTPVersion}
	Output parameter	– status	Response	{HTTPVersion} {statusCode} {reasonPhrase} {response body}
3	suspendSubscription		HTTP POST	
	Input parameter	– managerId – subscriptionId	Request	"POST" {notificationServiceURI}/"subscriptions"/ {subscriptionId}/"suspendSubscription" {HTTPVersion}
	Output parameter	– status	Response	{HTTPVersion} {statusCode} {reasonPhrase} {response body} Response body schema: \$ref: "notificationApiDataSchema.json#/definitions/ NotificationInfo"

Table 2 – Notification operations mapping from [ITU-T M.3702] to this Recommendation

No.	[ITU-T M.3702]		[ITU-T Q.819]	
	Operation name		Operation name	
	Request/ response	Parameters	Request/ response	Parameters
4	resumeSubscription		HTTP POST	
	Input parameter	– managerId – subscriptionId	Request	"POST" {notificationServiceURI}/"subscriptions"/ {subscriptionId}/"resumeSubscriptions" {HTTPVersion}
	Output parameter	– status	Response	{HTTPVersion} {statusCode} {reasonPhrase} {response body}
5	getNotificationTypes		HTTP GET	
	Input parameter	– notificationIRPid	Request	"GET" {notificationServiceURI}/NotificationTypes {HTTPVersion}
	Output parameter	– status – notificationType List	Response	{HTTPVersion} {statusCode} {reasonPhrase} {response body} Response body schema: \$ref:"notificationApiDataSchema.json# /definitions/GetNotificationTypesResponse"
6	querySubscription		HTTP GET	
	Input parameter	– subscriptionId	Request	"GET" {notificationServiceURI}/"subscriptions"/ {subscriptionId}{HTTPVersion}
	Output parameter	– subscribed NotificationType – subscriptionStatus – destination – filteringCriteria – status	Response	{HTTPVersion} {statusCode} {reasonPhrase} {response body} Response body schema: \$ref: "notificationApiDataSchema.json# /definitions/QuerySubscriptionResponse"
7	modifySubscription		HTTP PATCH	
	Input parameter	– subscriptionId – filteringCriteria – destination – notificationTypes	Request	"PATCH" {notificationServiceURI}/"subscriptions"/ {subscriptionId} {HTTPVersion} {request body} Request body schema: \$ref:"notificationApiDataSchema.json#/definitions/Mod ifySubscriptionRequest"
	Output parameter	– status	Response	{HTTPVersion} {statusCode} {reasonPhrase} {response body} Response body schema: \$ref: "notificationApiDataSchema.json# /definitions/NotificationInfo"
8	listAllSubscriptionIds		HTTP GET	
	Input parameter	– managerId	Request	"GET" {notificationServiceURI}/"subscriptions"/ listAllSubscriptionIds {HTTPVersion} {request body} Request body schema: \$ref: "notificationApiDataSchema.json# /definitions/ListAllSubscriptionsIdsRequest"
	Output parameter	– subscriptionIdSet – status	Response	{HTTPVersion} {statusCode} {reasonPhrase} {response body} Response body schema: \$ref:"notificationApiDataSchema.json#/definitions/List AllSubscriptionsIdsResponse"

Table 2 – Notification operations mapping from [ITU-T M.3702] to this Recommendation

No.	[ITU-T M.3702]		[ITU-T Q.819]	
	Operation name		Operation name	
	Request/ response	Parameters	Request/ response	Parameters
9	report notification:		HTTP POST (from provider to consumer)	
			Request	"POST" {destinationURI} {HTTPVersion} {request body} Request body schema: \$ref: "notificationApiDataSchema.json# /definitions/NotifyRequest"
			Response	{HTTPVersion} {statusCode} {reasonPhrase} {response body}

A detailed interface definition using YAML can be found in clause A.1.

8.3 Notification format definitions

8.3.1 Notification format to be used in this Recommendation

This clause will provide a description for the notification format to be used in this Recommendation.

8.3.2 Common notification header definition

This clause provides the detailed common notification header described in [ITU-T M.3702].

Table 3 provides the definition of the parameters of the common notification header.

Table 3 – Common notification header definition

Parameter name	Qualifiers	JSON/YAML-schema type	Descriptions
objectClass	M	string	It is the class name of the MO which sends out this notification. See clause 7.3.5 of [ITU-T M.3702]
objectInstance	M	uri	It is the distinguished name (DN) of the MO which sends out this notification. See clause 7.3.5 of [ITU-T M.3702]
notificationId	M	string	It is the identifier of the notification for the purpose of notification correlation. See clause 7.3.5 of [ITU-T M.3702]
eventTime	M	string	It is the time when this event happened. See clause 7.3.5 of [ITU-T M.3702]
systemDN	M	type:string format:uri	It is the DN of the system which sends out this notification. See clause 7.3.5 of [ITU-T M.3702]

Table 3 – Common notification header definition

Parameter name	Qualifiers	JSON/YAML-schema type	Descriptions
notificationType	M	<pre>NotificationType: type: string enum: - objectCreation - objectDeletion - attributeValueChange - stateChange - communicationAlarm - environmentalAlarm - equipmentAlarm - processingErrorAlarm - qualityOfServiceAlarm - integrityViolation - operationalViolation - physicalViolation - securityViolation - timeDomainViolation - relationshipChange - heartbeat</pre>	It indicates the type of notification. See clause 7.3.5 of [ITU-T M.3702]. The possible common notification types defined in this Recommendation can be found in the list in the left column. This list can be extended to include new notification types when they are defined.

8.3.3 Common notification contents definition

- Notification contents for the objectCreation and objectDeletion notifications are shown in Table 4.

Table 4 – Notification contents for objectCreation and objectDeletion

Notification parameter	Qualifiers	Data type	Descriptions
correlatedNotifications	O	<pre>CorrelatedNotifications type: array items: type: string</pre>	List of correlated notifications
additionalText	O	string	Text message
additionalInfo	O	string	Additional information not in text format. (The information will be provided in JSON strings. Below is the same.)

Table 4 – Notification contents for objectCreation and objectDeletion

Notification parameter	Qualifiers	Data type	Descriptions
sourceIndicator	O	SourceIndicator: type: string format: enum - resourceOperation - managementOperation - unknown	Cause of event
attributeList	O	NVPairList: type: object properties: attributeList: type: array items: \$ref: '#/.../NVPair' NVPair: type: object properties: name: type: string value: type: string type: type: string	Attribute values

2) Notification contents for the attributeValueChange notification are shown in Table 5.

Table 5 – Notification contents for attributeValueChange

Notification parameter	Qualifiers	Data type	Descriptions
correlatedNotifications	O	CorrelatedNotifications	List of correlated notifications
additionalText	O	string	Text message
additionalInfo	O	string	Additional information not in text format
sourceIndicator	O	SourceIndicator	Cause of event
attributeChanges	M	NVPairList	Changed attributes