

International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.782

(05/2012)

SERIES X: DATA NETWORKS, OPEN SYSTEM
COMMUNICATIONS AND SECURITY

OSI management – Management functions and ODMA
functions

Guidelines for defining web services for managed objects and management interfaces

Recommendation ITU-T X.782

ITU-T X-SERIES RECOMMENDATIONS
DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

PUBLIC DATA NETWORKS

Services and facilities	X.1–X.19
Interfaces	X.20–X.49
Transmission, signalling and switching	X.50–X.89
Network aspects	X.90–X.149
Maintenance	X.150–X.179
Administrative arrangements	X.180–X.199

OPEN SYSTEMS INTERCONNECTION

Model and notation	X.200–X.209
Service definitions	X.210–X.219
Connection-mode protocol specifications	X.220–X.229
Connectionless-mode protocol specifications	X.230–X.239
PICS proformas	X.240–X.259
Protocol Identification	X.260–X.269
Security Protocols	X.270–X.279
Layer Managed Objects	X.280–X.289
Conformance testing	X.290–X.299

INTERWORKING BETWEEN NETWORKS

General	X.300–X.349
Satellite data transmission systems	X.350–X.369
IP-based networks	X.370–X.379

MESSAGE HANDLING SYSTEMS

DIRECTORY	X.400–X.499
-----------	-------------

OSI NETWORKING AND SYSTEM ASPECTS

Networking	X.500–X.599
Efficiency	X.600–X.629
Quality of service	X.630–X.639
Naming, Addressing and Registration	X.640–X.649
Abstract Syntax Notation One (ASN.1)	X.650–X.679
	X.680–X.699

OSI MANAGEMENT

Systems management framework and architecture	X.700–X.709
Management communication service and protocol	X.710–X.719
Structure of management information	X.720–X.729

Management functions and ODMA functions	X.730–X.799
--	--------------------

SECURITY	X.800–X.849
----------	-------------

OSI APPLICATIONS

Commitment, concurrency and recovery	X.850–X.859
Transaction processing	X.860–X.879
Remote operations	X.880–X.889
Generic applications of ASN.1	X.890–X.899

OPEN DISTRIBUTED PROCESSING	X.900–X.999
-----------------------------	-------------

INFORMATION AND NETWORK SECURITY	X.1000–X.1099
----------------------------------	---------------

SECURE APPLICATIONS AND SERVICES	X.1100–X.1199
----------------------------------	---------------

CYBERSPACE SECURITY	X.1200–X.1299
---------------------	---------------

SECURE APPLICATIONS AND SERVICES	X.1300–X.1399
----------------------------------	---------------

CYBERSECURITY INFORMATION EXCHANGE	X.1500–X.1599
------------------------------------	---------------

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

Recommendation ITU-T X.782

Guidelines for defining web services for managed objects and management interfaces

Summary

Recommendation ITU-T X.782 defines a set of guidelines for managed object modelling and a management interface for web services-based network management. It composes a framework for web services-based network management interfaces along with Recommendation ITU-T Q.818. It specifies how service-oriented web service interfaces should be defined. It covers the suitable application scenario of web services in network management interfaces, generic accessing methods of XML-based managed objects, information modelling in web service WSDL and XML schema. Some WSDL definitions and XML schema are provided for defining some basic data types: generic managed object (MO) and generic MO accessing methods. This Recommendation and Recommendation ITU-T Q.818 together compose a framework for web services-based network management interfaces with a wide range of applications.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T X.782	2012-05-14	2

Keywords

Distributed processing, extensible markup language (XML), managed objects, network management interfaces, web service (WS), web services description language (WSDL), XML schema.

FOREWORD

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

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

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

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

NOTE

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

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

INTELLECTUAL PROPERTY RIGHTS

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

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

© ITU 2012

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 Overview of a web services-based management framework	4
7 Principles for service-oriented WSDL-based interface design.....	4
8 Definition of a generic managed object using XML schema	5
8.1 Web-service role in management interfaces.....	5
8.2 Definition of managed objects using XML schema	6
9 Accessing methods for managed objects.....	9
10 Inheritance of managed objects and interface operations.....	11
10.1 Attributes inheritance of managed objects	11
10.2 Considerations for the inheritance of interface operation	12
11 Information modelling guidelines for web services-based interfaces	12
11.1 namespace.....	12
11.2 complexType	12
11.3 attribute.....	12
11.4 request.....	12
11.5 response	13
11.6 notification.....	13
11.7 Name conventions for MOCs, packages, attributes and data types	13
12 Style idioms for XML schema specifications.....	14
12.1 Data model using XML schema	14
12.2 XML schema design considerations.....	14
12.3 Recommendations for schema developers	16
12.4 Guidelines for schema extensions	18
13 Compliance and conformance	18
13.1 Standards document compliance	19
13.2 System conformance	19
13.3 Conformance statement guidelines.....	19
Annex A – Common WSDL and XML schema definitions	20
A.1 XML schema definitions for common data types and a generic managed object	20

	Page
A.2 WSDL and XML schema definition for common object accessing methods.....	28
Appendix I – Overview of web service technology and application scenarios in network management interfaces	34
I.1 Characteristics of web service technology	34
I.2 Suitable and unsuitable application scenarios of web services in network management.....	35
Bibliography.....	37

Recommendation ITU-T X.782

Guidelines for defining web services for managed objects and management interfaces

1 Scope

The network management architecture defined in [ITU-T M.3010] introduces the use of multiple management protocols. So far, the GDMO/CMIP and CORBA GIOP/IIOP 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 web service/XML is now an additional paradigm for network management.

This Recommendation, together with [ITU-T Q.818] sets out to define a framework for defining how interfaces supported by management systems and network elements should be modelled using web service/XML schema. It is within the scope of this Recommendation to provide the following guidelines or instructions:

- service-oriented WSDL-based interface design approach;
- suitable and unsuitable application scenarios for web services in network management interfaces;
- generic accessing methods for managed objects;
- inheritance of managed objects and interfaces;
- information modelling guidelines for web services-based interfaces;
- style conventions for web service WSDL and XML schema specifications.

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 E.164] | Recommendation ITU-T E.164 (2010), <i>The international public telecommunication numbering plan</i> . |
| [ITU-T M.3010] | Recommendation ITU-T M.3010 (2000), <i>Principles for a telecommunications management network</i> . |
| [ITU-T M.3020] | Recommendation ITU-T M.3020 (2011), <i>Management interface specification methodology</i> . |
| [ITU-T M.3701] | Recommendation ITU-T M.3701 (2010), <i>Common management services – State management – Protocol neutral requirements and analysis</i> . |
| [ITU-T Q.818] | Recommendation ITU-T Q.818 (2012), <i>Web service-based management services</i> . |
| [ITU-T X.701] | Recommendation ITU-T X.701 (1997), <i>Information technology – Open Systems Interconnection – Systems management overview</i> . |
| [ITU-T X.703] | Recommendation ITU-T X.703 (1997), <i>Information technology – Open Distributed Management Architecture</i> . |

[ATIS-I-000002]	ATIS Specification ATIS-I-000002 (2011), <i>ATIS XML Schema Development Guidelines</i> .
[OASIS WSN]	OASIS Specification (2006), <i>Web Services Base Notification v1.3</i> .
[OASIS UDDI]	OASIS Specification (2004), <i>Universal Description, Discovery and Integration (UDDI) v3.0.2</i> .
[W3C Primer]	W3C Recommendation (2004), <i>XML Schema Part 0: Primer Second Edition</i> .
[W3C SOAP]	W3C Recommendation (2007), <i>SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)</i> .
[W3C WSDL]	W3C Recommendation (2001), <i>Web Services Description Language (WSDL) 1.1</i> .
[W3C XML]	W3C Recommendation (2000), <i>Extensible Markup Language (XML) 1.0 Second Edition</i> .
[W3C XS-P1]	W3C Recommendation (2004), <i>XML Schema Part 1: Structures Second Edition</i> .
[W3C XS-P2]	W3C Recommendation (2004), <i>XML Schema Part 2: Datatypes Second Edition</i> .

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 agent [ITU-T M.3020]

3.1.2 managed object class [ITU-T X.701]

3.1.3 manager [ITU-T M.3020]

3.1.4 notification [ITU-T X.703]

3.2 Terms defined in this Recommendation

This Recommendation does not define any new terms.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

B2B	Business to Business
BPEL	Business Process Execution Language
C2B	Customer to Business
CMIP	Common Management Information Protocol
CORBA	Common Object Request and Broker Architecture
DCOM	Distribute Component Object Model
DN	Distinguished Name
DTD	Document Type Definition
EMS	Element Management System
GDMO	Guidelines for the Definition of Managed Objects

GIOP	General Inter-ORB Protocol
IDL	Interface Definition Language
IOP	Internet Inter-ORB Protocol
IT	Information Technology
LAN	Local Area Network
MO	Managed Object
MOC	Managed Object Class
MOO	Multiple Object Operation
NMS	Network Management System
OOAD	Object-Oriented Analysis and Design
OS	Operating System
RDN	Relative Distinguished Name
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
TMN	Telecommunications Management Network
UDDI	Universal Description Discovery and Integration
WS	Web Services
WSDL	Web Services Description Language
WSN	Web Services Notification
XML	extensible Markup Language
XSD	XML Schema Definition

5 Conventions

A few conventions are followed in this Recommendation to make the reader aware of the purpose of the text. While most of the Recommendation is normative, paragraphs succinctly stating mandatory requirements to be met by a management system (managing and/or managed) are preceded by a boldface "R" enclosed in parentheses, followed by a short name indicating the subject of the requirement, and a number. For example:

(R) EXAMPLE-1 An example mandatory requirement.

Requirements that may be optionally implemented by a management system are preceded by an "O" instead of an "R". For example:

(O) EXAMPLE-2 An example optional requirement.

The requirement statements are used to create compliance and conformance profiles.

Examples of WSDL and XML are included in this Recommendation and normative WSDL and XML specifying the data types, base classes and other service-oriented modelling constructs of the framework are included in Annex A. The WSDL and XML are written in a 10 point courier typeface:

```

<!-- Example XML schema -->
<xsd:complexType name="AType">
  <xsd:sequence>
    <xsd:element name="a" type="xsd:string"/>
    <xsd:element name="b" type="xsd:long"/>
  </xsd:sequence>
</xsd:complexType>

```

6 Overview of a web services-based management framework

Web services have been widely used in the IT industry. Appendix I provides more information on the features of web services technology. Web services similar to CORBA technology, can be used in network management interfaces.

This Recommendation together with [ITU-T Q.818] sets up a framework for defining how interfaces supported by management systems and network elements should be modelled using WSDL and XML schema.

The web services-based management framework includes the following aspects.

- 1) Managed object and interface definition guidelines:
 - principles for the service-oriented WSDL-based interface design;
 - definition of managed objects using XML schema;
 - accessing methods for MOs;
 - inheritance of MOs and interface operations;
 - information modelling guidelines for web services-based interface operations;
 - style idioms for WSDL and XML schema specifications.
- 2) Web services supporting services for network management:
 - definition of notification services using OASIS web services base notification [OASIS WSN];
 - usage of OASIS UDDI service registration [OASIS UDDI];
 - definition of a heartbeat service;
 - definition of a multiple object operations (MOO) service;
 - definition of a containment service.

This Recommendation mainly deals with the managed object and interface definition guidelines, and [ITU-T Q.818] mainly deals with the web services supporting services for network management. The two Recommendations together form a web services-based management framework.

7 Principles for service-oriented WSDL-based interface design

This clause identifies some interface design considerations that should be addressed by this framework through service-oriented interfaces. It provides the modelling principles for service-oriented managed objects and their accessing methods.

The service-oriented design considerations related to WSDL and XSD repertoire and modelling concerns super-classes, naming of managed objects and service-oriented interfaces, operations and notifications.

A web service is a service-oriented technology compared with the traditional CORBA/IDL and GDMO/CMIP management paradigms. An object-oriented interface analysis and design (OOAD) focuses on the class level, i.e., it encapsulates behaviour and related data in the same object which exposes one or more interfaces for access to their encapsulated states and attributes. The service-

oriented interface design separates the encapsulated data and state from the behaviour, thus loose coupling can be achieved at a higher level. In this approach, the behaviour of some (shared) objects is no longer controlled by themselves but by a few predefined interface operations.

This Recommendation, along with [ITU-T Q.818], defines a lightweight generic use of service-oriented interface design patterns. The management and controlling functions are defined using a service-grained approach, that is, interface operations are organized within the unit of a certain service (management function sets, such as configuration management, performance management, etc.), not an individual management object class. This service-orientation requires the flexibility of application-specific access granularity where well-defined sets of TMN entity types are accessed through predefined WSDL interfaces.

The framework has the following principles to define a service-oriented WSDL and XSD based management information model and interfaces.

- All interface interactions are defined as WSDL operations, each operation includes a request and an optional corresponding response when needed.
- Each MOC is defined as an XML complexType when exchanged through the management interface, and each attribute or state of the MOC is defined as an element in the complexType.
- The naming of MOC instances follows the DN concept, but it is a string containing all of the RDN list.
- A generic service is defined in this framework, which contains five generic object accessing methods: createMO, deleteMO, getMOAttribute, setMOAttribute, and getPackages. Also, the generic object accessing methods use "name-value" pairs to express the properties and their values of different types of MOC instances.
- Other interface control functions are defined as WSDL interface operations which are organized as a service with the granularity of management function sets.
- Common data types are defined as XML schemas which can be shared by application-specific interface definitions.
- Notifications sent from the agent to the manager should follow the format and behaviour defined in [OASIS WSN]. The control of notification management services are further defined in [ITU-T Q.818].

8 Definition of a generic managed object using XML schema

8.1 Web-service role in management interfaces

To support the software objects representing manageable resources, a base class is defined for use in modelling network resources. Other MOCs (managed object class) in information models must be derived from this base class in order to operate within this framework. Some generic accessing methods and some other extended services are defined to provide interfaces to manage MOs.

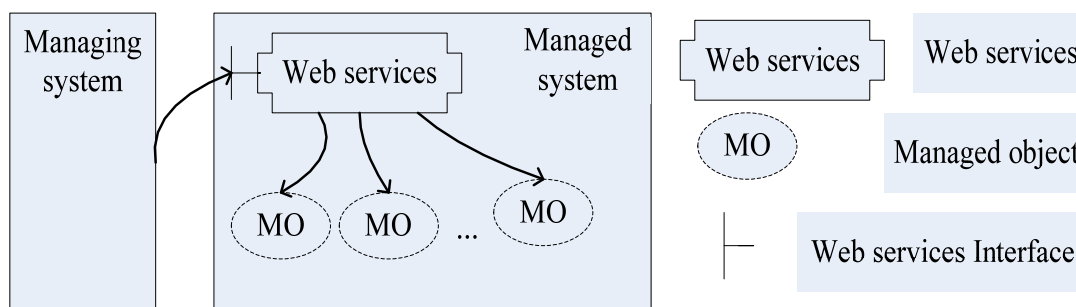


Figure 1 – Web services role

Figure 1 shows how a managing system accesses a managed system that supports a web services interface. A web services interface acts as an intermediate entity that enables a managing system to manage proper MOs in a managed system representing manageable resources.

8.2 Definition of managed objects using XML schema

An MO is the OSI management view of a resource that is subject to management, such as a connection or an item of physical equipment. Thus, an MO is the abstraction of such a resource that represents its properties for the purpose of management. An MO may include attributes that provide information used to characterize itself and operations that represent its behaviours. The purpose of the framework is to provide a collection of capabilities to manage these MOs. MOs need some approaches to describe their properties and behaviours. A service-oriented MO is a managed entity that represents a manageable service resource in terms of shared state and behaviour where state and behaviour are separated through outsourcing of the behaviour to an assigned so-called "managing entity" (e.g., a service and its interface) that takes a steward role with regard to the behaviours of its allocated managed entities. Since an MO's state and behaviour can be separated, state can be described by XML schema and behaviour by web services' interfaces in WSDL. One important benefit of using an XML document to store an MO's state is that web services use XML schema to describe the data type of its exchanged messages, and these XML-based MOs' information can be exchanged without any modification.

8.2.1 Definition of a generic managed object class

A managed object class is a further abstraction of managed objects. All network resources have some common attributes and all MOCs shall inherit, either directly or indirectly, from a super class, namely a ManagedObject. Using ManagedObject to define new MOCs will be easier and faster and provide better maintenance. As mentioned above, all MOCs are described in XML schema and the data type of ManagedObject is given in Table 1 and the attributes can be found in Table 2.

Table 1 – Data type of super class ManagedObject

```
<xsd:complexType name="ManagedObject_C">
  <xsd:sequence>
    <xsd:element name="objectClass" type="xsd:string"/>
    <xsd:element name="objectInstance" type="x782:NameType"/>
    <xsd:element name="packages" type="x782:PackageListType"/>
    <xsd:element name="creationSource" type="x782:SourceIndicatorType"/>
  </xsd:sequence>
</xsd:complexType>
```

Table 2 – Attributes of super class ManagedObject_C

Attribute name	Support qualifier	Read qualifier	Write qualifier
objectClass	Mandatory	Mandatory	–
objectInstance	Mandatory	Mandatory	–
packages	Optional	Mandatory	–
creationSource	Optional	Mandatory	–

As shown in Table 2, ManagedObject is made up of four attributes including objectClass, objectInstance, packages and creationSource. An attribute has an associated value, which may have a simple or a complex structure. Here, the attribute of MO is different from the attribute in the XML

schema specification, and it can be just mapped to the element in the XML schema. The attribute `objectClass` is used to identify the class type of this MO instance. The attribute `objectInstance` is used to uniquely identify an MO instance, and the data type is using `NameType` (also it has the same semantics of DN, distinguished name) as specified in (1). The attribute `packages` is a set of strings to indicate the capacities that the MO supports. The attribute `creationSource` indicates whether an MO is created automatically in a managed system, or by a managing system through a management operation, or unknown.

DN(list of xsd:string) ::= "<attribute_name_1>=<attribute_value_1>",
 "<attribute_name_2>=<attribute_value_2>"
 ...
 "<attribute_name_n>=<attribute_value_n>" (1)

The attributes in the above formula should be naming attributes of MOCs.

A complete XML schema definition for the generic `ManagedObject_C` is defined in clause A.1.

(R) OBJECT-1. All the classes used to model resources on a managed system shall inherit (directly or indirectly) from the *ManagedObject_C* described above and defined in the XML schema in clause A.1. The capabilities described above shall be supported.

8.2.2 Inheritance of managed objects

One MOC shall be defined as a specialization of another MOC by utilizing inheritance such as for all the rest of the MOs that must directly or indirectly inherit from `ManagedObject`. Specialization of an MOC implies that all attributes defined on the super class will be supported by the subclass too. As attributes of MOs are described in XML schema, the inheritance can be achieved by the extension of data types. For example, assume the equipment MOC inherits directly from the base `ManagedObject` class, and equipment can inherit all attributes from `ManagedObject` by extension and declare other attributes as shown in Table 3, such as `userLabel`, for itself.

Table 3 – Data type of Equipment_C by extension

```
<xsd:complexType name="Equipment_C">
  <xsd:complexContent>
    <xsd:extension base="x782:ManagedObject_C">
      <xsd:sequence>
        <xsd:element name="userLabel" type="xsd:string"/>
        ...
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

Some MOs may need multiple inheritance, but the XML schema only supports single inheritance for data types and the use of multiple inheritance is not recommended. If the semantics of multiple inheritance are required, the derived MOC has to singly inherit from one of the superior MOCs and attributes from the other superior class(es) should be added manually.

8.2.3 Package feature

Packages can be used to group certain capabilities (for example, related attributes), or provide conditional support capabilities. A package can be defined as an XSD `complexType`, with a "_P" as its name suffix. When it is used, the element may have a type of the `complexType` representing this

package, with minOccurs="0" maxOccurs="1" as the qualifiers, which indicate this package can be conditional or optional. An example of package definition and usage is shown in Table 4.

Table 4 – Data type of equipment by extension

```
<!-- definition of a Package -->
<xsd:complexType name="StatePackage_P">
  <xsd:sequence>
    <xsd:element name="administrativeState"
type="x782:AdministrativeStateType"/>
    <xsd:element name="operationalState" type="x782:OperationalStateType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Equipment_C">
  <xsd:complexContent>
    <xsd:extension base="x782:ManagedObject_C">
      <xsd:sequence>
        <xsd:element name="equipmentId" type="xsd:string"/>
        <xsd:element name="userLabel" type="xsd:string"/>
        ...
      <!-- usage of the Package -->
      <xsd:element name="statePackage" type="x782:StatePackage_P" minOccurs="0"
maxOccurs="1" /></xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

8.2.4 Common attributes and data types

The following table shows some common attributes as well as some common data types that can be shared by this framework.

Table 5 – Standard attributes and data types

Attribute name	Data type	Description
administrativeState	AdministrativeStateType	See [ITU-T M.3701] for more details
availabilityStatus	AvailabilityStatusSetType	See [ITU-T M.3701] for more details
backedUpStatus	BackedUpStatusType	See [ITU-T M.3701] for more details
controlStatus	ControlStatusSetType	See [ITU-T M.3701] for more details
creationSource (Note)	SourceIndicatorType	See [ITU-T M.3701] for more details
externalTime	ExternalTimeType	
objectClass (Note)	ObjectClassType	It indicates an MOC
objectInstance (Note)	NameType	It indicates an MO instance
operationalState	OperationalStateType	See [ITU-T M.3701] for more details
packages (Note)	StringSetType	It indicates the packages that are supported