# Standardized Connectivity Management Objects

For use with OMA Device Management

Candidate Version 1.0 – 12 Aug 2008

**Open Mobile Alliance**

OMA-DDS-DM_ConnMO-V1_0- 20080812-C

# Contents



# Figures

# 1. Scope

This document defines a set of managed objects which offer a standardized way to represent connectivity settings in a mobile device OMA Device Management tree. There were three main requirements that guided the drafting of this specification:

- All connectivity parameters bootstrapped using OMA Client Provisioning can be subsequently addressed and managed via an OMA DM server;
- An OMA DM server can add new or managing existing proxies and network access points using this enabler;
- It should be possible to extend management objects with proprietary extensions.

While these objects are optional for any OMA DM implementation, their widespread use will simplify the management of basic connectivity parameters in mobile terminals.

The objects are defined using the OMA DM Device Description Framework [DMTND]. And each of the objects has standardized points of extension to permit implementation-specific parameters to accompany the standardized parameters. This added flexibility is intended to encourage the use of the standardized object while not unnecessarily restricting individual vendor innovations.

# 2. References

## 2.1 Normative References


**[ConnMOIPMO]** *Standardized IP Management Objects, Version 1.0,* Open Mobile Alliance™, OMA-DDS-DM_ConnMO_IP-V1_0-D, URL:http://www.openmobilealliance.org

**[OMADM]** *OMA Device Management Enabler, Version 1.2,* Open Mobile Alliance™ OMA-ERELD- DM-V1_2, URL:http://www.openmobilealliance.org

~~**[RFC1321]** *RFC1321, The MD5 Message-Digest Algorithm,* R. Rivest 1992, URL; http://www.ietf.org/rfc/rfc1321.txt~~

~~**[RFC1334]** *RFC1334, PPP Authentication Protocols,* B. Lloyd, W. Simpson 1992, URL; http://www.ietf.org/rfc/rfc1334.txt~~

~~**[RFC1994]** *RFC1994, PPP Challenge Handshake Authentication Protocol,* W. Simpson 1996, URL; http://www.ietf.org/rfc/rfc1994.txt~~

**[RFC2119]** "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997, URL:http://www.ietf.org/rfc/rfc2119.txt


## 2.2 Normative Authorities of References

Various parameters specified in the management objects defined in this document rely on an authority outside the scope of this specification to supply the set of acceptable values and value formats. In such references to external authority, only the directly cited material is referenced, not the entire external specification. The following authorities of reference are cited in this document:


~~**[AUTH-APN]** *3GPP TS 23.003 Numbering, addressing and identification, chapter 9,* URL:http://www.3gpp.org/~~

**[AUTH-RFC791]** *RFC 791, Internet Protocol*, DARPA, 1981, URL:http://www.ietf.org/rfc/rfc791.txt

~~**[AUTH-RFC2131-DHCPv4]** *RFC 2131, Dynamic Host Configuration Protocol,* Bucknell University, 1997, URL:http://www.ietf.org/rfc/rfc2131.txt~~

~~**[AUTH-RFC2396-ALPHA]** *RFC 2396, Uniform Resource Identifiers (URI): Generic Syntax,* The Internet Society, 1998, URL:http://www.ietf.org/rfc/rfc2396.txt~~

~~**[AUTH-RFC2462-AutoIPv6]** *RFC 2462, IPv6 Stateless Address Autoconfiguration,* The Internet Society, 1998, URL:http://www.ietf.org/rfc/rfc2462.txt~~

~~**[AUTH-RFC3315-DHCPv6]** *RFC 3315, Dynamic Host Configuration Protocol for IPv6 (DHCPv6),* The Internet Society, 2003, URL:http://www.ietf.org/rfc/rfc3315.txt~~

~~**[AUTH-RFC3330-AutoIPv4]** *RFC 3330, Special-Use IPv4 Addresses, definition of 169.254.0.0/16* IANA, 2002, URL:http://www.ietf.org/rfc/rfc3330.txt~~

~~**[AUTH-RFC3513-ADDR]** *RFC 3513, Internet Protocol Version 6 (IPv6) Addressing Architecture*, §§2.2, 2.3 The Internet Society, 2003, URL:http://www.ietf.org/rfc/rfc3513.txt~~

**[E212]** *ITU-T E.212 Identification Plan For Land Mobile Stations,* ITU, URL*:* http://www.itu.org/

~~**[GENFORM]** *WAP General Formats Document,* Open Mobile Alliance™, WAP-188-WAPGenFormats, URL:http://www.openmobilealliance.org~~

**[TIA/EIA-136-005A]** *Introduction, Identity, and Semi-Permanent Memory,* TIA/EIA


## 2.3 Informative References


**[DMStdObj]** *OMA Device Management Standardized Objects, Version 1.2*, Open Mobile Alliance™ OMA-TS-DM-DMStdObj-V1_2, URL:http://www.openmobilealliance.org

**[DMTND]** *OMA Device Management Tree and Description, Version 1.2*, Open Mobile Alliance™ OMA-TS-DM-DMTND-V1_2, URL:http://www.openmobilealliance.org

**[ProvCont]** *Provisioning Content, Version 1.1*, Open Mobile Alliance™, OMA-WAP-ProvCont-v1_1, URL:http://www.openmobilealliance.org

# 3. Terminology and Conventions

## 3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

## 3.2 Definitions

| | |
|---|---|
| **HTTP Proxy** | An endpoint for the HTTP protocol providing authenticated access, content caching, content encoding translation, address redirection, and other functions. |
| **Logical Proxy** | A logical proxy is a set of physical proxies that may, for example, share the same WSP and WTLS context (shared session id value space). This specification does not distinguish between logical and physical proxies but permits the configuration of both logical and physical elements of all proxy types. |
| **Network Access Point** | A physical access point is an interface point between the wireless network and the fixed network. It is often a Remote Access Server, an SMSC, a USSDC, an 802.11 Access Point, or something similar. It has an address (such as a telephone number or 802.11 SSID) and an access bearer. |
| **Physical Proxy** | A physical proxy is a specific address at which proxy functionality is provided. It can be the IP address plus port for an IP accessible proxy, or the SME-address plus port for an SMS accessible proxy. This specification does not distinguish between logical and physical proxies but permits the configuration of both logical and physical elements of all proxy types. |
| **WAP Proxy** | The WAP proxy is an endpoint for the WTP, WSP and WTLS protocols, as well as a proxy that is able to access WAP content. A WAP Proxy can have functionality such as that of, for example, a WSP Proxy or a WTA Proxy. |
| **WSP Proxy** | A generic WAP proxy; similar in functionality to an HTTP proxy. It is a variant of a WAP Proxy. |
| **WTA Proxy** | The WTA Proxy is a Wireless Telephony proxy as defined by WAP. |

## 3.3 Abbreviations

| | |
|---|---|
| **AAA** | Authentication, Authorization, and Accounting |
| **APN** | Access Point Name |
| **CDMA** | Code Division Multiple Access |
| **CDPD** | Cellular Digital Packet Data |
| **CHAP** | Challenge Handshake Authentication Protocol |
| **CSD** | Circuit Swithed Data |
| **DDF** | Device Description Framework |
| **DNS** | Domain Name System |
| **DTD** | Document Type Definition |
| **GPRS** | General Packet Radio Service |
| **GSM** | Global System for Mobile communications |
| **HA** | Home Agent |
| **HTTP** | HyperText Transfer Protocol |
| **IP** | Internet Protocol |

| | |
|---|---|
| **MAN** | Mobitex Subscription Number |
| **NAP** | Network Access Point |
| **OTA** | Over The Air |
| **PAP** | Password Authentication Protocol |
| **SME** | Short Message Entity |
| **SMS** | Short Message Service |
| **SMSC** | Short Message Service Centre |
| **SPI** | Security Parameter Index |
| **URI** | Uniform Resource Identifier |
| **URL** | Uniform Resource Locator |
| **USSD** | Unstructured Supplementary Service Data |
| **USSDC** | Unstructured Supplementary Service Data Centre |
| **WAP** | Wireless Application Protocol |
| **WSP** | Wireless Session Protocol |
| **WTA** | Wireless Telephony Application |
| **WTLS** | Wireless Transport Layer Security |
| **WTP** | Wireless Transaction Protocol |
| **XML** | Extensible Mark-up Language |

# 4. Introduction

Other OMA DM and common specifications define the syntax and semantics of the OMA DM protocol. However, the usefulness of such a protocol would be limited if the managed entities in devices required different data formats and displayed different behaviors. Central to the management of any suite of devices is the ability to configure and update the basic connectivity settings in each device. To avoid the situation where each device vendor defines a specialized and non-standard arrangement for managing connectivity parameters, this specification defines a set of management objects to permit the standardized representation and management of connectivity parameters in devices.

Since device manufacturers will always develop new functions in their devices and since these functions often are proprietary, no standardized management objects will exist for them. To make these functions manageable in the devices that have them, a device description framework is needed that can provide servers with the necessary information they must have in order to manage the new functions. The intention with this framework is that device manufacturers will publish descriptions of their devices as they enter the market. Organizations operating device management servers should then only have to feed the new description to their servers for them to automatically recognize and manage the new functions in the devices.

The introduction of vendor-specific extensions to these standardized objects is anticipated. Each management object contains several points of extension for arbitrary use by client implementations to allow collocation of proprietary innovations along with the standardized parameters. It is anticipated that vendors will include their implementation-specific extensions in published DDF that describes the management objects supported by each terminal model.

Finally, it is usual to find over time that network protocols grow and are replaced as the market cycle plays out. These management objects are structured in such a way as to be resilient to the addition of new bearer and proxy types without requiring wholesale replacement of the object definitions. In this way, the common structure survives into future versions of the management objects thus easing the burden of transition from old bearer types to new.

# 5. Justification

This Reference Release includes several Management Object definitions for use, in conjunction with the OMA Device Management Enabler, to manage data network connectivity settings for mobile terminals over common bearer and proxy types.

## 5.1 Standardized Connectivity Management

Providing a standardized set of management objects for configuration of data network connectivity through the OMA Device Management system will improve the usability and customer experience of mobile terminals that rely upon data services. As proposed, the management object definitions may be used in conjunction with OMA Device Management Candidate and Approved Enabler Releases over a variety of transports including: HTTP, HTTPS, OBEX over IrDA, OBEX over Bluetooth, and various forms of Smart Card.

## 5.2 Application-Neutral

Producing these management object definitions in an application-neutral fashion, we avoid reinvention of solutions to the same set of problems for each of new application that requires data connectivity. This reduces the connectivity parameters that an application must define to a simple reference node, ConRef (Connectivity Reference).

## 5.3 Bearer-Neutral

By presenting the specifications in two parts, a bearer-neutral part and bearer-specific bindings, we reinforce the OMA principle of network neutrality while providing specificity where needed but without bias for or against any particular network type.

# 6. Standardized Connectivity Management Objects

## 6.1 Introduction to Management Objects (Informative)

Management objects are the entities that can be manipulated by management actions carried over the OMA DM protocol. A management object can be as small as an integer or large and complex like a background picture, screen saver, or security certificate. The OMA DM protocol is neutral about the contents, or values, of the management objects and treats the node values as opaque data.

### 6.1.1 Definition and description of management objects

OMA DM management objects are defined using the OMA DM Device Description Framework [DMTND], or DDF. The use of this description framework produces detailed information about the device in question. However, due to the high level of detail in these descriptions, they are sometimes hard for humans to digest and it can be a time consuming task to get an overview of a particular object's structure.

In order to make it easier to quickly get an overview of how a management object is organized and its intended use, a simplified graphical notation in the shape of a block diagram is used in this document. Even though the notation is graphical, it still uses some printable characters, e.g. to denote the number of occurrences of a node. These are mainly borrowed from the syntax of DTDs for XML. The characters and their meaning are defined in the following table.

| Character | Meaning |
|---|---|
| + | one or many occurrences |
| * | zero or more occurrences |
| ? | zero or one occurrences |

If none of these characters is used the default occurrence is exactly once.

There is one more feature of the DDF that needs to have a corresponding graphical notation, the un-named block. These are blocks that act as placeholders in the description and are instantiated with information when the nodes are used at run-time. Un-named blocks in the description are represented by a lower case character in italics, e.g. *x*.

Each block in the graphical notation corresponds to a described node, and the text is the name of the node. If a block contains an *x,* it means that the name is not known in the description and that it will be assigned at run-time. The names of all ancestral nodes are used to construct the URI for each node in the management object. It is not possible to see the actual parameters, or data, stored in the nodes by looking at the graphical notation of a management object.

For a further introduction to this graphical notation, please refer to [DMStdObj].

## 6.2 DDF compliance

The management object descriptions in this document are normative. However, the descriptions also contain a number of informative aspects that could be included to enhance readability or serve as examples. Other informative aspects are, for instance, the ZeroOrMore and OneOreMore elements, where implementations may introduce restrictions. All these exceptions are listed here:

- All XML comments, e.g. "<!-- some text -->", are informative.
- The descriptions do not contain an RTProperties element, or any of its child elements, but a description of an actual implementation of this object MAY include these.
- If a default value for a leaf node is specified in a description, by the DefaultValue element, an implementation MUST supply its own appropriate value for this element. If the DefaultValue element is present in the description of a node, it MUST be present in the implementation, but MAY have a different value.

- The value of all Man, Mod, Description and DFTitle elements are informative and included only as examples.
- Below the interior nodes Ext and BearerParams, an implementation may add further nodes at will.
- The contents of the AccessType element MAY be extended by an implementation.
- If any of the following AccessType values are specified, they MUST NOT be removed in an implementation: Copy, Delete, Exec, Get, and Replace.
- If the AccessType value Add is specified it MAY be removed in an implementation if the implementation only supports a fixed number of child nodes.
- An implementation MAY replace the ZeroOrMore or OneOreMore elements with ZeroOrN or OneOrN respectively. An appropriate value for *N* must also be given with the …*OrN* elements.

### 6.2.1 Conformance Definitions

The status definition in the node definitions indicates if the client supports that node or not. If the status is "REQUIRED" then the client MUST support that node in the case the client supports the parent node.

## 6.3 The OMA DM Connectivity management objects

This specification defines two management objects, that while distinct objects, should be considered together. A *NAP* object permits configuration of *network access points* and contains both bearer-neutral and bearer-specific parameters. A *Proxy* object facilitates configuring *network proxies* of various kinds and is bearer-neutral but may include parameters specific to particular proxy types.

### 6.3.1 Relationship to WAP Client Provisioning (Informative)

These management objects describe various kinds of network access points and proxies. Certain aspects of NAPDEF, PXLOGICAL, and PXPHYSICAL characteristics defined in WAP Client Provisioning [ProvCont] have been altered to improve generality. However, it remains possible to use these object definitions to manage the proxy settings provisioned using [ProvCont].

When compared to the OMA Client Provisioning specifications, specifically [ProvCont], the PXPHYSICAL parameters have been collapsed together with the PXLOGICAL parameters in this mapping. This simplifies other forms of proxy (e.g. HTTP or ISA proxies) and does not limit the ability to specify the mapping of logical to physical proxy.

## 6.4 NAP Object

### 6.4.1 Introduction

The NAP object facilitates management of network access point parameters.

OMA DM [OMADM] protocol compatibility for the NAP MO is version 1.2 or any later compatibility version to this version.

## 6.4.2 Graphical Representation (Informative)

The following figure provides the structure of NAP DM management object.

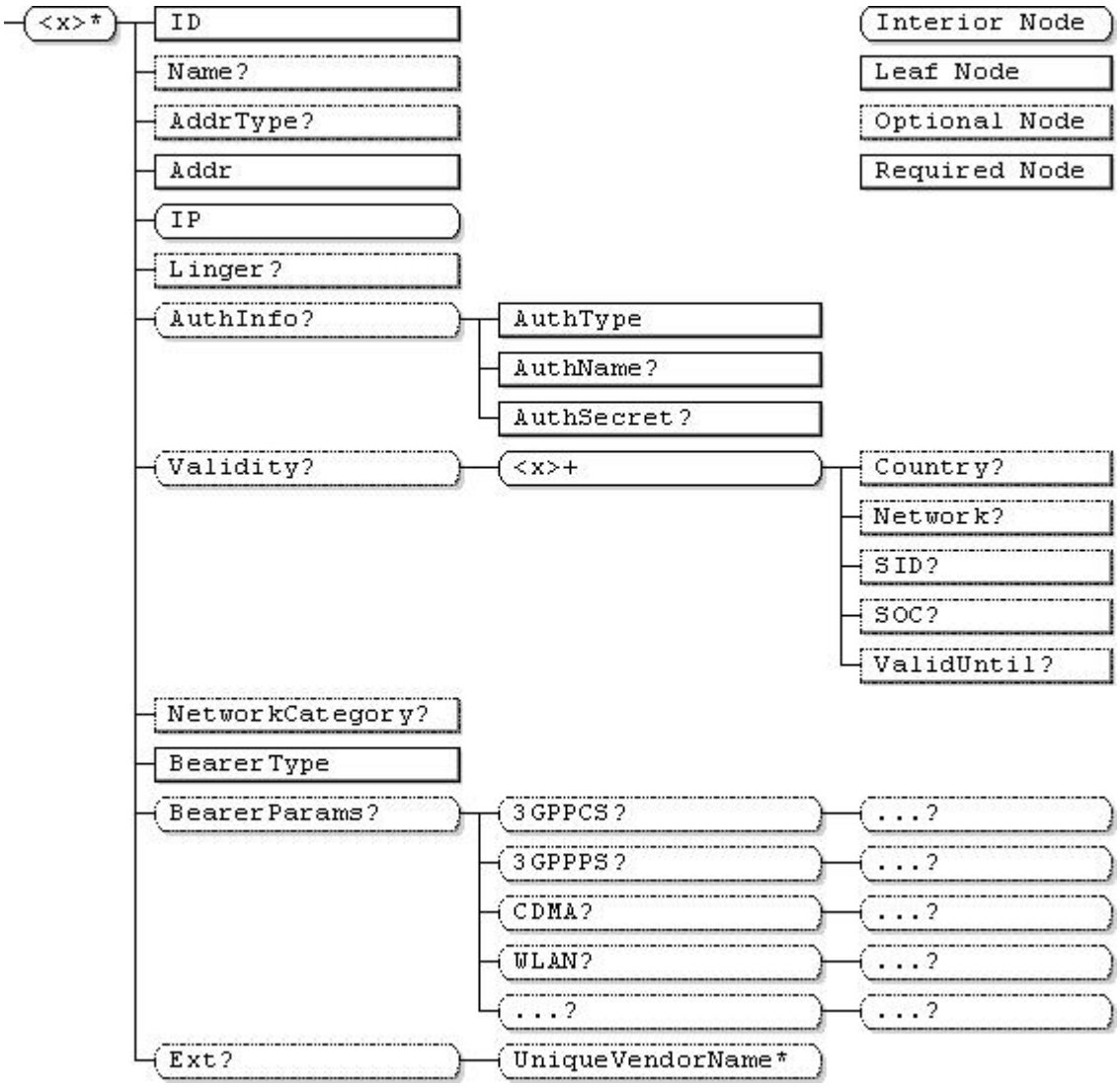


**Figure 1: NAP Management Object (Informative)**

## 6.4.3 Node Descriptions

**…/<x>**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrMore | node | Get |

This interior node specifies the unique object id of a *network access point*, or NAP, management object. The purpose of this interior node is to group together the parameters of a single NAP object. The ancestor elements of this node define the position in the management tree of the NAP object. But the structure of the DM tree and hence positions in the tree of management objects is out of scope of this specification. Management Object Identifier for the NAP MO MUST be: “urn:oma:mo:oma-connmo-nap:1.0”.

**ID**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | One | chr | Get |

This leaf node defines the identity of the one specific network access point which an instance of this management object represents.

**Name**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | chr | Get |

This leaf node provides an optional human readable name for the network access point.

**AddrType**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | chr | Get |

This leaf node specifies the type of NAP address supplied as the **Addr** leaf node value. Authorities of reference and allowed values for **AddrType** are bearer-specific and are not listed here. Implementers are referred to the bearer specifications and the listed normative references describing each protocol for the proper usage context and details of each address type.

If the bearer type-specific parameter specification does not define a value for this node then the default value “IPv4” MUST be assumed and the **Addr** node MUST represent an IPv4 address [AUTH-RFC791] represented in string form dotted-decimal CIDR notation.

**Addr**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | One | chr | Get |

This leaf node represents the address of the network access point. The value and mapping to a character representation are dependent upon the bearer type. (See the AddrType above for details on address format and acceptable values.)

**IP**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | One | node | Get |

This interior node is parent node of the IP sub-tree specified in the [ConnMOIPMO] specification.

**Linger**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | int | Get |

This optional leaf node, if present, defines how long a connection should be kept active without any traffic. The value is expressed in seconds as a 16 bit unsigned integer.

**AuthInfo**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | node | Get |

This interior node optionally provides authentication credentials to support various forms of network access point authentication.

**AuthInfo/AuthType**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | One | chr | Get |

This leaf node identifies one type of authentication protocol used by the represented network access point. The means for selecting a particular authentication protocol to be used when connecting to the network access point is out of scope of this specification—this node merely links the specified authentication method to the AUTHNAME and AUTHSECRET.

Authorities of reference and allowed values for these authentication protocols are bearer-specific and are not listed here. Implementers are referred to the bearer specifications and the listed normative references describing each protocol for the proper usage context and details of each authentication type.

**AuthInfo/AuthName**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | chr | Get |

The *name* (or user id*)* credential.

**AuthInfo/AuthSecret**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrOne | chr | NO Get |

The *secret* (or password) credential.

**Validity**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | node | Get |

This optional interior node is used to indicate where this NAP is valid.

**Validity/<x>**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | OneOrMore | node | Get |

This interior node distinguishes different Validity rules.

**Validity/<x>/Country**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | chr | Get |

This optional leaf node indicates a Mobile Country Code as defined by ITU-T [E212]. The parameter is used to identify in which country this NAP is valid.

**Validity/<x>/Network**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | chr | Get |

This optional leaf node indicates a list of comma separated Mobile Network Codes in decimal format as defined by ITU-T [E212]. The parameter is used to identify in which network this NAP is valid.

**Validity/<x>/SID**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | chr | Get |

This optional leaf node indicates a list of comma separated System IDs in decimal format as defined by [TIA/EIA-136-005A]. The parameter is used to identify in which SID this NAP is valid.

**Validity/<x>/SOC**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | chr | Get |

This optional leaf node indicates a System Operator Code as defined by [TIA/EIA-136-005A]. The parameter is used to identify in which SOC this NAP is valid.

**Validity/<x>/ValidUntil**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | int | Get |

This optional leaf node defines the end of the (time) period of validity. The parameter is expressed as a 32bit unsigned integer in seconds, from the time it is received by the client device. When the server receives this value, for example using the Get Command, the device MUST calculate and deliver the remaining valid time in seconds.

**NetworkCategory**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | chr | Get |

This optional leaf node describes the network category name that can be used to categorize NAP definitions. The value is an arbitrary identifier, e.g. INTERNET, and it can be shared by different NAP definitions.

**BearerType**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | One | chr | Get |

This leaf node is used to define the bearer type used to reach this network access point. Conforming management objects describing NAPs reachable over bearers MUST use the associated BearerType value specified in the sub-tree management objects specification representing the bearer-specific configuration parameters. Management objects describing NAPs reachable over bearers not specified are still considered conforming if they otherwise conform to this management object specification. If a BearerType value is not understood, the NAP object MUST be ignored.

**BearerParams**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | node | Get |

This optional interior node defines the specific parameters of the bearer concerned with BearerType. These parameters are used to reach this network access point.

**BearerParams/3GPPCS**

**BearerParams/3GPPCS/…**

**BearerParams/3GPPPS**

**BearerParams/3GPPPS/…**

**BearerParams/CDMA**

**BearerParams/CDMA/…**

**BearerParams/WLAN**

**BearerParams/WLAN/…**

**BearerParams/…**

**BearerParams/…/…**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | Node | Get |

The Names of BearerParams child nodes (eg. 3GPPPS) and the bearer parameters are defined in separate specifications. Nodes in these sub-trees represent the bearer-specific configuration parameters. Bearer parameters in these sub-trees are defined in separate specifications.

**Ext**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | node | Get |

This optional interior node designates the single top-level branch of the NAP DM management object tree into which vendor extensions MAY be supported, permanently or dynamically. Ext sub trees, such as this one, are included at various places in the DM connectivity management objects to provide flexible points of extension for implementation-specific parameters. However, vendor extensions MUST NOT be defined outside of one of these `Ext` sub-trees.

**Ext/UniqueVendorName**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrMore | node | Get |

This interior node is supplied by a vendor to distinguish their extension from those of other vendors. The *UniqueVendorName* SHOULD be a trademark or company name controlled by each vendor to ensure uniqueness. The structure of any su-btree below a *UniqueVendorName* interior node is implementation-specific.

# 6.5 Proxy Object

## 6.5.1 Introduction

The Proxy object is used to configure and update proxy settings for network proxies of various types in a standardized way.

OMA DM protocol compatibility for the Proxy MO is version 1.2 or any later compatibility version to this version.

## 6.5.2 Graphical Representation (Informative)

The following figure shows the structure of the Proxy management object:

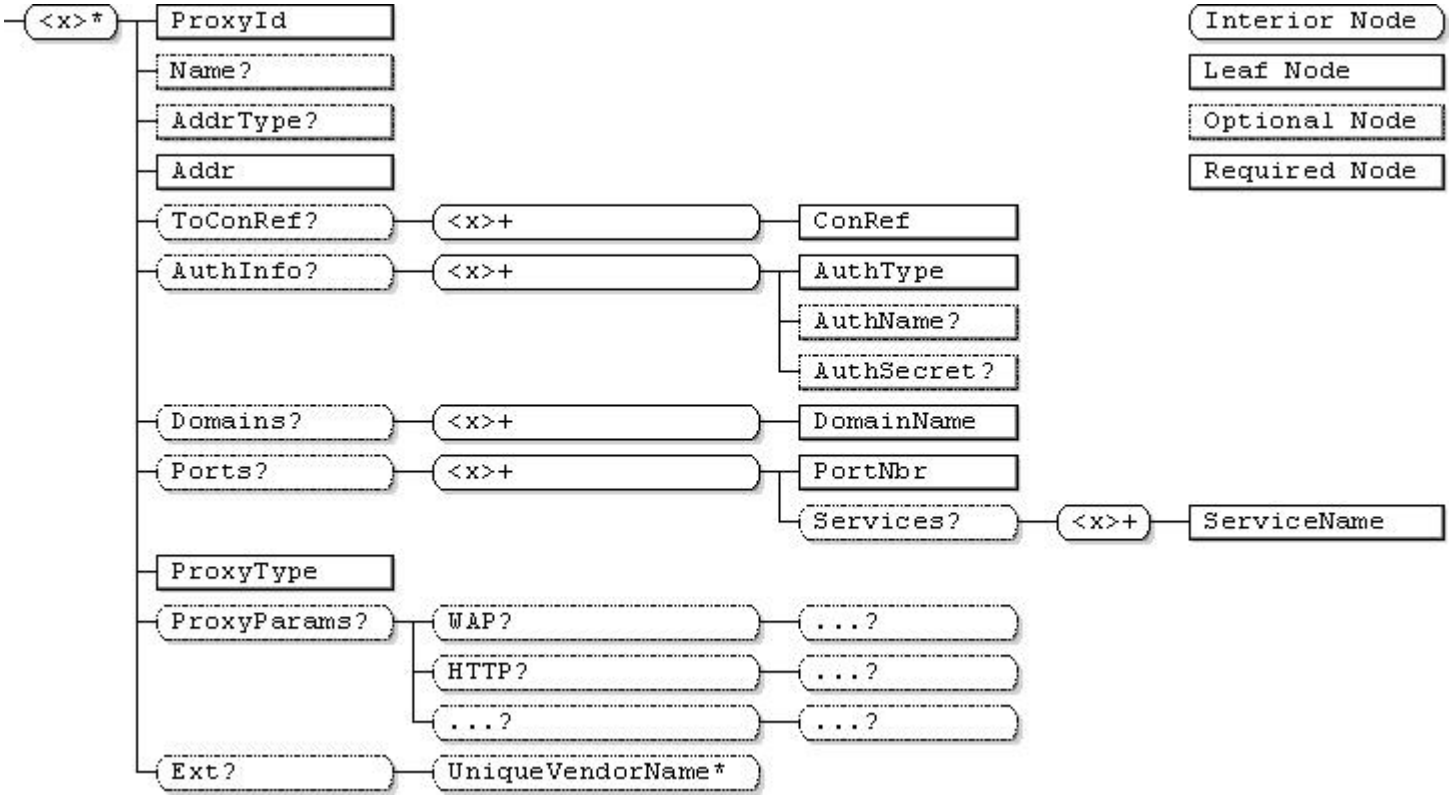


Figure 2: Proxy Management Object (Informative)

## 6.5.3 Node Descriptions

**…/<x>**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | ZeroOrMore | node | Get |

This interior node specifies the unique object id of a proxy management object. The purpose of this interior node is to group together the parameters of a single proxy object. The ancestor elements of this node define the position in the management tree of the proxy object. But the structure of the DM tree and hence positions in the tree of management objects is out of scope of this specification. Management Object Identifier for the Proxy MO MUST be: “urn:oma:mo:oma-connmo-proxy:1.0”.

**ProxyId**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | One | chr | Get |

This leaf node uniquely identifies this Proxy Management Object.

**Name**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | chr | Get |

This leaf node provides an optional human readable name for the proxy.

**AddrType**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | chr | Get |

This optional leaf node specifies the type of proxy address supplied as the **Addr** leaf node value. Authorities of reference and allowed values for **AddrType** are proxy-specific and are not listed here. Implementers are referred to the proxy specifications and the listed normative references describing each protocol for the proper usage context and details of each address type.

If the proxy type-specific parameter specification does not define a value for this node then the default value "IPv4" MUST be assumed and the **Addr** node MUST represent an IPv4 address [AUTH-RFC791] represented in string form dotted-decimal CIDR notation.

**Addr**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | One | chr | Get |

This leaf node provides the address of the proxy. The Addr node can specify addresses of different types, for example an IP address or an SME number. The type of address in the field is defined by the AddrType node. This node MUST be present for the proxy object to be valid.

**ToConRef**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | node | Get |

This optional interior node lists network access points (or other connection objects) used by this proxy to reach a network. One proxy could utilize any of several different network access points.

**ToConRef/<x>**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | OneOrMore | node | Get |

This interior node distinguishes the connection object identifier nodes. There must be exactly one ConRef node for each of these interior nodes.

**ToConRef/<x>/ConRef**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | One | chr | Get |

This node specifies a reference to the connectivity. It is expected that either a Proxy or NAP MO is specified, but other, implementation-specific connections may be referenced..

**AuthInfo**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | node | Get |

This interior node optionally provides authentication credentials to support various forms of proxy authentication.

**AuthInfo/<x>**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | OneOrMore | node | Get |

This interior node distinguishes different AuthInfo parameters.

**AuthInfo/<x>/AuthType**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | One | chr | Get |

This leaf node identifies one type of authentication protocol used by the represented proxy. The means for selecting a particular authentication protocol to be used is out of scope of this specification—this node merely links the specified authentication method to the AUTHNAME and AUTHSECRET. The AuthType value must be unique within a proxy MO.

Authorities of reference and allowed values for these authentication protocols are proxy-specific and are not listed here. Implementers are referred to the proxy specifications and the listed normative references describing each protocol for the proper usage context and details of each authentication type.

**AuthInfo/<x>/AuthName**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | chr | Get |

The *name* (or user id*)* credential.

**AuthInfo/<x>/AuthSecret**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | chr | NO Get |

The *secret* (or password) credential.

**Domains**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | node | Get |

This optional node lists the domains which can be reached via this proxy.

**Domains/<x>**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | OneOrMore | node | Get |

This interior node distinguishes the domain name nodes.

**Domains/<x>/DomainName**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | One | chr | Get |

This leaf node names a single domain which can be reached via this proxy.

**Ports**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | node | Get |

This optional interior node lists the ports which can be traversed via this proxy. A port number is defined for each port along with an optional list of services offered by the proxy at that port.

**Ports/<x>**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | OneOrMore | node | Get |

This interior node distinguishes the ports opened by this proxy.

**Ports/<x>/PortNbr**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | One | int | Get |

This leaf node defines the number of a single port as a 16bit unsigned integer which is opened by this proxy.

**Ports/<x>/Services**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | node | Get |

This optional interior node lists the services offered at the specified port.

**Ports/<x>/Services/<y>**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | OneOrMore | node | Get |

These interior nodes distinguish the services offered at the specified port.

**Ports/<x>/Services/<y>/ServiceName**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | One | chr | Get |

This leaf node identifies the name of a single service offered at the specified port.

**ProxyType**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Required | One | chr | Get |

This leaf node is used to define the proxy type. A conforming management object describing a proxy MUST use the associated ProxyType value specified in the sub-tree management objects specification representing the proxy-specific configuration parameters. Management objects describing types of network proxy not specified are still considered conforming if they otherwise conform to this management object specification. If a ProxyType value is not understood, the Proxy object MUST be ignored.

**ProxyParams**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | node | Get |

This optional interior node defines the specific parameters of the proxy concerned with ProxyType. These parameters are used to configure the proxy.

**ProxyParams/WAP**

**ProxyParams/WAP/…**

**ProxyParams/HTTP**

**ProxyParams/HTTP/…**

**ProxyParams/…**

**ProxyParams/…/…**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | node | Get |

Nodes in these sub-trees represent the configuration parameters specific to a single proxy type. Specific Proxy support in these sub-trees is defined in separate specifications.

**Ext**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrOne | node | Get |

This optional interior node designates the single top-level branch of the Proxy DM management object tree into which vendor extensions MAY be supported, permanently or dynamically. Ext sub trees, such as this one, are included at various places in the DM connectivity management objects to provide flexible points of extension for implementation-specific parameters. However, vendor extensions MUST NOT be defined outside of one of these `Ext` sub-trees.

**Ext/UniqueVendorName**

| Status | Occurrence | Format | Min. Access Types |
|---|---|---|---|
| Optional | ZeroOrMore | node | Get |

This interior node is supplied by a vendor to distinguish their extension from those of other vendors. The *UniqueVendorName* SHOULD be a trademark or company name controlled by each vendor to ensure uniqueness. The structure of any subtree below a *UniqueVendorName* interior node is implementation-specific.

# 7. Operational Considerations

ConnMO is normatively dependent on the DM 1.2 specifications. However, this normative dependency should not be seen as restricting these MO definitions only to DM clients implementing that version of the DM enabler.

For example, a management authority may exchange ConnMO data-files using means not specifically defined in the DM 1.2 enabler.

# Appendix A. Change History (Informative)

## A.1 Approved Version History

| Reference | Date | Description |
|---|---|---|
| n/a | n/a | No prior version |

## A.2 Draft/Candidate Version 1.0 History

| Document Identifier | Date | Sections | Description |
|---|---|---|---|
| Draft Versions<br>OMA-DDS-DM_ConnMO-V1_0 | 03 Oct 2007 | All | First version based on the corresponding TS |
| | 07 Oct 2007 | 5 | Added from the WID |
| | 17 Oct 2007 | 7<br><br>6 | Added from the approved ETR<br>Added CR:<br>OMA-DM-ConnMo-2007-0014R01-CR_NAP_Nokia_Revisions |
| | 07 Feb 2008 | All | Editorial updates:<br>- New template<br>- Table format for node definitions |
| | 15 April 2008 | 6 | Editorial update<br>- Using the ToConRef/<x>/ConRef syntax from StdObj |
| | 08 May 2008 | 5, 6 | Editorial updated according to consistency review comments C001 & C002 |
| Candidate Versions<br>OMA-DDS-DM_ConnMO-V1_0 | 12 Aug 2008 | n/a | Status changed to Candidate by TP<br>TP ref#: OMA-TP-2008-0286-INP_Connectivity_Management_Objects_V1_0_RRP_for_Candidate_Approval |