Device Profile Evolution Technical Specification

Approved Version 1.0 – 05 Jul 2011

Open Mobile Alliance

OMA-TS-DPE-V1_0-20110705-A

Contents

Figures

Tables

1. Scope

The scope of the Device Profile Evolution technical specification document is to define the message flows and descriptions between DPE Server, DPE Client and Service Provider for the Device Profile Evolution enabler. Also, this document defines the DPE core vocabulary.

The mechanisms defined in this document fulfils the functional capabilities needed to support DPE enabler as described in the DPE Requirements document [DPE-RD] and comply to the architecture defined in DPE Architecture document [DPE-AD].

2. References

2.1 Normative References

[IANACharset] IANA MIBEnum Character Set Registry, URL: ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets

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

[RFC4234] "Augmented BNF for Syntax Specifications: ABNF". D. Crocker, Ed., P. Overell. October 2005, URL:http://www.ietf.org/rfc/rfc4234.txt

[SCRRULES] "SCR Rules and Procedures", Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, URL:http://www.openmobilealliance.org/

[DPE-RD] "Device Profile Requirements", Open Mobile Alliance™, OMA-RD-DPE-V1_0, URL:http://www.openmobilealliance.org/

[DPE-AD] "Device Profile Evolution Architecture", Open Mobile Alliance™, OMA-AD-DPE-V1_0, URL:http://www.openmobilealliance.org/

[DPE-MO] "OMA DPE Management Object", Open Mobile Alliance™, OMA-TS-DPE_MO-V1_0, URL:http://www.openmobilealliance.org/

[DPE-Schema] "OMA DPE XML Schema", Open Mobile Alliance™, OMA-SUP-XSD_DPE-V1_0, URL:http://www.openmobilealliance.org/

[PUSH-OTA] "Push Over The Air", WAP Forum™, WAP-235-PushOTA, URL:http://www.openmobilealliance.org/

[PUSH-PAP] "Push Access Protocol", WAP Forum™, WAP-247-PAP, URL:http://www.openmobilealliance.org/

[PUSH-MSG] "Push Message", WAP Forum™, WAP-251-PushMessage, URL:http://www.openmobilealliance.org/

[RFC2616] "Hypertext Transfer Protocol – HTTP/1.1". Fielding R.; Gettys J.; Mogul J.; Frystyk H.; Masinter L.; Leach P.; Berners-Lee T., June 1999. URL: http://www.ietf.org/rfc/rfc2616.txt

[SEC_CF] "Security Common Functions Architecture", Open Mobile Alliance™, OMA-AD-SEC_CF-V1_0-20080902-A, URL: http://www.openmobilealliance.org/

[WBXML] "Binary XML Content Format Specification". WAP Forum™. WAP-192-WBXML. URL:http://www.openmobilealliance.org/

2.2 Informative References

[OMADICT] "Dictionary for OMA Specifications", Version x.y, Open Mobile Alliance™, OMA-ORG-Dictionary-Vx_y, URL:http://www.openmobilealliance.org/

[UAProf] "User Agent Profile", February 2006, OMA-TS-UAProf-V2_0-20060206-A.pdf, 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

DPE Service Provider	An entity that operates a DPE Server and provides DPE enabler service to the end user and Service Provider.
Service Provider	An entity that manages and distributes software-based services and content to the end user.
Device Capability	The overall set of characteristics and related parameters supported by a Device.
Device Property	A hardware, software, or network characteristic that [is] associated with the Device Capability of a Device at a given point in time. A Device property can be either static or dynamic.
Dynamic Device Property	A Device property that may change its value e.g. as a result of hardware, software or configuration changes.
Static Device Property	A Device property that does not change its value. Examples are display resolution, processor type, etc.

3.3 Abbreviations

OMA	Open Mobile Alliance
DCAP	Device Capability
DPE	Device Profile Evolution
SP	Service Provider
WLAN	Wireless Local Area Network
CDMA	Code Division Multiple Access
EDGE	Enhanced Data Rate for GSM Evolution
GNSS	Global Navigation Satellite System
GPRS	General Packet Radio Service
GPS	Global Positioning System
LTE	Long Term Evolution
NFC	Near Field Communication
PAP	Push Access Protocol
PPG	Push Proxy Gateway
SIP	Session Initiation Protocol
TD-SCDMA	Time Division-Synchronous Code Division Multiple Access
USB	Universal Serial Bus
WCDMA	Wideband Code Division Multiple Access
WLAN	Wireless Local Area Network
WiMAX	Worldwide Interoperability for Microwave Access

4. Introduction

Mobile applications and services are required to function in varying network environments with users having devices with a wide range of capabilities. The device capabilities and network conditions can vary dynamically and applications and services need to be able to respond to these changes accordingly.

The Device Profile Evolution enabler is intended to provide a standardized solution to convey information on the device capabilities. While other enablers such as UAProf [UAProf] can only convey information on static device capabilities, the DPE enabler can convey information either on static device capabilities or on dynamic device capabilities, the dynamic aspect being the main added-value of the DPE enabler. This information, in the form of property names and values, directly comes from the device and is communicated to a Service Provider (SP), allowing an enhancement of the quality of the services provided to the device user. This enhancement could be, for example, a better content adaptation, or services matching to the maximum extent the Device capabilities, resulting at the end on a better user experience.

In the DPE architecture, a client-server model is defined. In the existing OMA architecture two new entities are created: a DPE Server and a DPE Client. Any SP willing to get information on the device capabilities of a given user using the DPE enabler has to submit its requests to the DPE Server managing the DPE Client of that given user.

The DPE technical specification document defines the message flows and descriptions between DPE Server, DPE Client and SP.

An SP can also transmit policies to DPE Server. The policies will be applied at the DPE Client level to the associated device capabilities. The use of policies is detailed in this document.

Also, the Core Vocabulary of device capabilities is defined in this document.

4.1 Version 1.0

The DPE V1.0 Enabler supports the following functionality.

- Support for DPE Client ID registration
- Support for query and reporting of single or multiple properties
- Support for setting and release of group for multiple properties
- Support for setting and release of policies to a property or a group
- Support for query and reporting of properties by group
- Support for policy based subscription and notification of property changes

5. DPE Functionality and Usage

5.1 DPE Usage

Within a Device there are a lot of static or dynamic Device Capabilities in the form of property names and values. This information is needed by a lot of services for a better content adaptation, or services matching to the maximum extent the Device Capabilities, resulting at the end on a better user experience.

For a particular service, a set of Device Capabilities / Properties may fall into the interest of the Service Provider. The Service Provider can determine the Device Capabilities / Properties according to the service and then get the values through DPE Server as described below:

- Query mechanism: The Service Provider sends a query request with single or multiple properties to directly get the values. A service can be represented by a set of properties hence the group can be defined and set by the Service Provider. Then the Service Provider sends a group query request with group ID to directly get the values.
- Subscription mechanism: The Service Provider subscribes the property changes with certain policies through DPE Server. The subscription is either targeting individual properties or a property group. The DPE Client then stores the policy and monitors the changes on related Device Capabilities / Properties. Whenever the changes match the policy the DPE Client will send a notification to DPE Server which will be forwarded to the Service Provider for consumption.

5.2 DPE Property Collection

To be made available through the DPE enabler, Device properties must be known by the DPE Client. DPE Clients SHALL support the collection and delivery through DPE Interfaces for the properties shown in Section 9.2.

The method of property collection by the DPE Client is unspecified in DPE 1.0, since there are expected to be variations between Device support for property exposure, e.g. by the Device operating system and software entities in the Device. Future versions of the DPE enabler may further define requirements for property collection methods, e.g. as standards become available for property collection Interfaces.

DPE Clients MAY enable device-based application access to a DPE Client's Dynamic Device Property values via a scripting environment such as EcmaScript Mobile Profile.

5.2.1 DPE Client Identity Delivery to Service Provider

In order for Service Providers to use the DPE-3 Interface to query Device properties for a particular client, the Service Provider must know the DPE Client ID that was assigned by the DPE Server. DPE defines two methods for the Service Providers to get the DPE Client ID:

- by query of the DPE Server (DPE Client ID Query message), using a query key as defined by the DPE Service Provider, e.g. SIP Public Identity, MSISDN, IMEI, User Identity. The DPE Service Provider defines the options for query keys through unspecified means, e.g. through SP onboarding processes.
- by direct delivery from Device clients, using an HTTP header "x-oma-dpe-client-id". DPE Clients SHOULD enable device-based user-agents to obtain the DPE Client Id, e.g. through a scripting environment such as EcmaScript Mobile Profile or other implementation-specific means."

6. Messages between entities

DPE interface messages are introduced in the following sections. The messages and device properties (defined in section 9) are designed per these goals:

- The message parameters (elements and attributes) are defined as “string” type in most cases, to allow flexibility in the implementation and deployment of DPE, for the syntax of the specific parameters.
- The messages are defined using an extensible XML schema [DPE-Schema], to allow for extensions to DPE functionality.
- The device properties are defined with the following specific types accordingly
    - Integer, where the property is clearly associated with a numeric value
    - Boolean, where the property is clearly associated with a “true” / “false” condition
    - As an enumerated token (tokenized string), where clearly associated with a limited set of options
    - As a unenumerated token, where a set of tokenized strings may be present: in this case, section 9 defines the set of typical values, although additional values may be present as extensions
    - As a simple string, where an unrestricted string value may be used

DPE Servers and DPE Clients SHALL generate DPE messages per the definitions in this chapter, and in accordance with [DPE-Schema].

DPE Servers and DPE Clients SHALL validate received messages per [DPE-Schema]. DPE Servers and DPE Clients SHALL silently ignore extensions that they do not support.

6.1 Messages through DPE-1

The DPE-1 Interface connects DPE Server and DPE Client. The DPE-1 Interface allows the communication between the DPE Client and the DPE Server, in order for the DPE Client to retrieve a unique identifier, which is called DPE Client ID and is assigned by the DPE Server. The DPE Client ID will be used afterwards for all the subsequent DPE communications. The respective internal functions involved in this process are the Registration function of the DPE Client and the ID Manager function of the DPE Server.

It is assumed DPE Client has already received the necessary information to access the DPE Server, via a bootstrap process or any other way of provisioning, at least, the DPE Server address. If supported by the DPE Client device, the DPE Server address and other connection information may have been provisioned using the DPE Management Object (MO) [DPE-MO].

DPE Clients will initiate registration with a DPE Server as described in [8.1.1 DPE Client Registration] and [6.1.2 Device Change]. Other than in those cases, how and when the registration process takes process is implementation dependent, and depends upon both client and server design and configuration Example approaches:

- DPE Servers are designed to initiate DPE Client registration upon arbitrary events, e.g. activation of a service that is enhanced via DPE, or detection of a Device change for a user.
- DPE Servers or DPE Clients may re-establish registration on a scheduled basis or upon each Device change.
- The DPE Client ID may be assigned for each registration, once for each user of a Device (i.e. valid while the associated MSISDN / SIP URI and Device identity does not change); or permanently for each specific Device (i.e. DPE Server to assigns a DPE Client ID to a particular Device and never changes it).

But in all these cases, there will be only two ways to start the registration process: DPE Client driven or DPE Server driven. In the case of a DPE Client starting the process, it will send a request to the DPE Server in order to be registered. When the DPE Server wants another registration for a particular DPE Client, it will force the Client to resend a request, in order to restart the whole process.

The DPE Server MAY also include a security token which can be used by the DPE Client for authentication of subsequent messages from the DPE Server. If provided by the DPE Server, the security token MUST be included by the DPE Client in all messages sent to the DPE Server.

If the DPE Server intiates DPE Client registration, it MUST include the current security token value, if any, as previously provided to the DPE Client. This ensures the DPE Client will trust the register command.

The format of the security token is implementation/deployment specific, except that it MUST be a value unique with the DPE Service Provider domain, to avoid creating ambiguity in the authentication of different DPE Clients.

6.1.1 Client Registration

The DPE Client initiated register process could occur in different cases:

- When the DPE Client connects the DPE Server for the first time ever (no DPE Client ID has never been assigned).
- When the Device is switched on and the association of user identity (MSISDN / SIP URI) – Device identity (IMEI / MAC address) has changed.

In any of these scenarios, the registration flow will include the following messages:

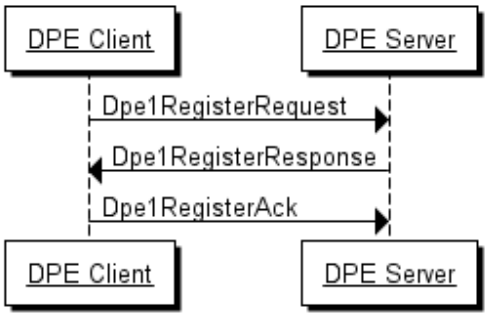

Figure 6-1 . DPE Client initiated registration flow between DPE Client and DPE Server.

Note that the register request attributes may vary according to the design and configuration of the DPE Client. For example, the included Device and User identities may be present, or may be omitted, e.g. being determined through unspecified network means, e.g. insertion by a network proxy or other network-based information delivered to the DPE Server. The security token may vary as well, e.g. a configured pseudo-identity, a OMA SEC-CF defined (e.g. GBA-derived) identity token may be included, or no security token (security then depending upon unspecified network-based means, if necessary at all, per DPE Server policy).

6.1.1.1 DPE-1 Register-Request

The "DPE-1 Register-Request" message is used by the DPE Client to start the registration on the DPE Server.

In the DPE-1 Register Request:

- The Device Identity could be used by the DPE Server to know what kind of Device is trying to register to the DPE Enabler, which could be very useful, for the DPE Server, to retrieve Device info from external sources. Moreover, the Device Identity associated with the User Identity will complete the needed information from a SP to connect the DPE Client in the case of a PUSH connection, and lets the DPE Server know whether the subscriber has changed its Device. Alternate sources for the Device Identity include unspecified network entities (e.g. RADIUS servers, network proxies, etc).

- The User Identity gives information about the subscriber trying to register to the DPE Server. Even when DPE Enabler only conveys information about Device capabilities, regardless of subscriber / end user preferences, his/her identity is needed to be know by the DPE Server. For example, on a push communication, when a SP wants to connect a particular subscriber without any prior request from this subscriber Device. In that case, the SP only

knows the end user wanted to connect, and the DPE Server will need to know the last working association known between User Identity/Device Identity. On the other side, it is important to note that in any case the User Identity will travel outside the DPE Server and reach a 3rd entity as a SP, in a case of normal request initiated by the User Agent of the Device.

The registration process ends up successfully when the DPE Server assigns a DPE Client ID to the DPE Client, to be used on the subsequent communications between DPE Client, DPE Server and the SPs.

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier "Register-Request"
Device-Identity	Optional	String	E.g., IMEI, MAC address. Inclusion of this parameter depends upon DPE Service Provider policy or as determined by the related attribute in the DPE MO.
User-Identity	Optional	String	E.g. MSISDN, SIP URI. Inclusion of this parameter depends upon DPE Service Provider policy or as determined by the related attribute in the DPE MO
DPE-Version	Mandatory	String	The DPE version information might be used by the DPE Server to assign a specific Client ID to the DPE Client, depending on its version, or even rule out the registration. In the current release, the DPE-Version MUST be set to "1.0".
Security-Token	Conditional	String	A security token to fulfil tasks of mutual authentication between DPE Client and DPE Server. MUST be present if this Register-Request is in response to a DPE1 Register-Command. If present, MUST be equal to the Security-Token received in DPE-1 Register-Command.
Message-ID	Mandatory	String	Identifies all the messages related to this query. Therefore the response to a given query MUST have the same *Message-ID*.

Table 6-1: DPE-1 Register-Request message elements

6.1.1.2 DPE-1 Register-Response

The "DPE-1 Register-Response" message is used by the DPE Server to answer back the DPE Client, with the needed information in order to accomplish the registration process.

If the DPE Server rejects the registration request, the DPE Client SHOULD NOT automatically retry registration.

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier "Register-Response"
Client-ID	Mandatory	String	This is the Unique Identifier given by the DPE Server in order to facilitate the anonymity of the DPE Client in further communications. This Identifier will be sent, hence, in any message coming from the DPE Client. The ID manager is an internal functionality of DPE Server in charge of allocating the Client ID's to the DPE Clients whose request has been received. This could be done depending on the policies set on the DPE Server (e.g., for example, the Client ID's could be ranged depending on the DPE Firmware version of the DPE Client).

Status	Mandatory	String	The status of the registration process informs the DPE Client whether the registration request has been successful or not. Possible values are “Success”, “Fail”.
Security-Token	Optional	String	Optionally included by the DPE Server to fulfil tasks of mutual authentication between DPE Client and DPE Server for all subsequent messages. MUST be a value that is unique within the DPE Service Provider domain.
Message-ID	Mandatory	String	Identifies all the messages related to this query. Therefore the response to a given query MUST have the same *Message-ID*.
Encoding-Supported	Mandatory	Boolean	Indicates that the DPE Server supports binary encoding of DPE messages per Appendix C for messages over the DPE-2 interface.

Table 6-2: DPE-1 Register-Response message elements

6.1.1.3 DPE-1 Register-Ack

The “DPE-1 Register-Ack” message is sent by the DPE Client to acknowledge the reception of the DPE Client ID from the DPE Server. This allows the DPE Server to know whether the DPE Session could be considered as established. The DPE Client will use the DPE Client ID granted by the DPE Server to identify itself.

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier “Register-Ack”
Client-ID	Mandatory	String	Current value assigned during registration.
Security-Token	Conditional	String	A security token to fulfil tasks of mutual authentication between DPE Client and DPE Server. MUST be present if a Security-Token was received in DPE1 Register-Response. If present, MUST be equal to the Security-Token received in DPE-1 Register-Response.
Message-ID	Mandatory	String	Identifies all the messages related to this query. Therefore the response to a given query MUST have the same *Message-ID*.

Table 6-3: DPE-1 Register-Ack message elements

6.1.2 DPE Server Initiated Registration

When for any particular reason, the DPE Server wants to change the DPE Client ID assigned to a DPE Client (and consequently, erase any cached property values / policies of this DPE Client), it will inform the DPE Client about the need of a new registration process. For that purpose, the DPE Server will send (push) a message to the DPE Client asking for a new registration, the DPE-1 Register-Command. To initiate re-registration, a DPE Server may also respond to a DPE Client request (e.g. a property update notification) with an error status, indicating that re-registration is required, upon which the DPE Client is expected to re-register. Upon the reception of this message, the DPE Client will start the same registration process shown above.

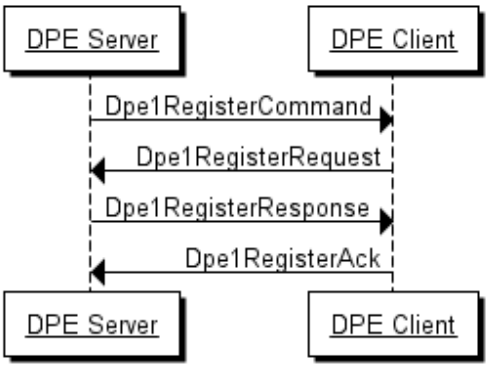

Figure 6-2 DPE Server initiated registration flow between DPE Client and DPE Server.

6.1.2.1 DPE-1 Register-Command

The "DPE-1 Register-Command" message is used by the DPE Server to force a DPE Client to do the registration process again, in order to assign it another DPE Client ID, the DPE Server has to initiate the process and inform the DPE Client.

This is done with the DPE-1Register-Command message. DPE-1 The Register-Command message may be pushed to the DPE Client (e.g. through WAP Push, SIP Push, or unspecified push means), or it may be provided as the error response body to a DPE Client request (e.g. a property update notification).

At the reception of this message, the DPE Client will understand that it is no more registered on the DPE Server and, consequently, there is no communication of its properties until a new successful registration takes place. Therefore, the DPE Client will start the same registration process we have seen in the previous section (§ section 5.2.2.1).

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier "Register-Command"
Client-ID	Mandatory	String	This is the Unique Identifier given by the DPE Server in order to facilitate the anonymity of the DPE Client in further communications. Set to the current value assigned during the prior registration, or a new value to be assigned to the DPE Client.
Security-Token	Conditional	String	A security token to fulfil tasks of mutual authentication between DPE Client and DPE Server. MUST be present if a Security-Token was sent in the prior DPE1 Register-Response. If present, MUST be equal to the Security-Token sent in DPE1 Register-Response.
Message-ID	Mandatory	String	Identifies all the messages related to this query. Therefore the response to a given query MUST have the same *Message-ID*.

Table 6-4: DPE-1 Register-Command message elements

6.2 Messages through DPE-2

The DPE-2 Interface allows to request the value(s) of a single (or a group of) Device property (ies), to transmit policies and to respond to those requests and policies.

The DPE-2 Interface allows the DPE Server to[1]:

- Query the value(s) of a single property or multiple properties
- Set and release groups of properties
- Query the values of a group's properties
- Set and release policies to a property or to a group

The DPE-2 Interface allows the DPE Client to:

- Advertise the value(s) of a single or multiple properties
- Advertise the values of a group's properties
- Acknowledge the setting and releasing of a group of properties (labelling)
- Acknowledge the setting and releasing of a policy to a property or to a group
- Advertise the value(s) of a single (or a group of) Device property(ies) when the conditions of a policy are met

6.2.1 Querying Properties

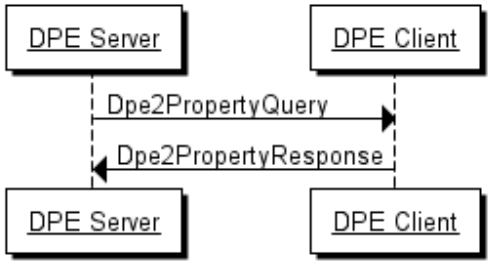

Figure 6-3: Flow of a DPE Server request for property (ies) to DPE Client, and its response.

Message	Requirement	Direction
DPE-2 Property-Query	Mandatory	DPE Server → DPE Client
DPE-2 Property-Response	Mandatory	DPE Server ← DPE Client

6.2.1.1 DPE-2 Property-Query

The "DPE-2 Property-Query" message is used by the DPE Server to:

- Query the value of a single property;
- Query the values of multiple properties

[1] These actions are issued by a Service Provider and transmitted by the DPE Server to the DPE Client.

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier “Property-Query”
Client-Id	Mandatory	String	Current value assigned during registration.
Security-Token	Conditional	String	A security token to fulfil tasks of mutual authentication between DPE Client and DPE Server. MUST be present if a Security-Token was sent in the prior DPE1 Register-Response. If present, MUST be equal to the Security-Token sent in DPE1 Register-Response.
Property-Names	Mandatory	String	Comma-separated list of the names of the properties queried. If “*”, the current value of all supported properties should be reported.
Message-ID	Mandatory	String	Identifies all the messages related to this query. Therefore the response to a given query MUST have the same *Message-ID*.

Table 6-5: DPE-2 Property-Query message elements

6.2.1.2 DPE-2 Property-Response

The “DPE-2 Property-Response” message is used by the DPE Client to advertise the value of a single or of multiple properties.

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier “Property-Response”
Client-Id	Mandatory	String	Current value assigned during registration.
Security-Token	Conditional	String	A security token to fulfil tasks of mutual authentication between DPE Client and DPE Server. MUST be present if a Security-Token was received in DPE1 Register-Response. If present, MUST be equal to the Security-Token received in DPE1 Register-Response.
Values	Mandatory	List of Structures	Contains the response to Property-Query message in the form of {name, value} for each property queried. The number of responses MUST be coherent with number of properties queried in the DPE-Property-Query message. Multiple responses may be included for properties with multiple values.
Unsupported	Conditional	String	The names of all properties that were indicated specifically, but are not supported by the device or DPE Client.
Message-ID	Mandatory	String	MUST be the same Message-ID as in the corresponding Property-Query message.

Table 6-6: DPE-2 Property-Response message elements

The property responses are returned as a sequence of one or more elements defined per the DPE schema:

Information Element	Req	Type	Description
Property-Name	Mandatory	String	Identifies the property being reported.

Property-Value	Mandatory	Per section 9	Current value of the property, or empty if unsupported. Valid values are defined per property in section 9.

Table 6-7: Values element generic structure

6.2.2 Querying Property Groups

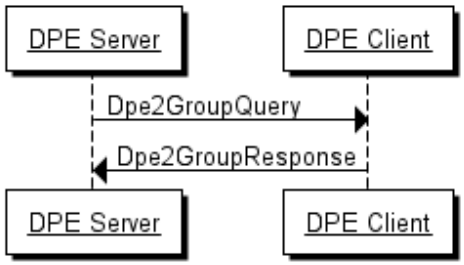

Figure 6-4: Flow of a DPE Server request for the values of a group's properties to DPE Client, and its response

Message	Requirement	Direction
Group-Query	Mandatory	DPE Server → DPE Client
Group-Response	Mandatory	DPE Server ← DPE Client

6.2.2.1 DPE-2 Group-Query

The " DPE-2 Group-Query" message is used by the DPE Server to query the values of a group's properties**.**

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier "Group-Query"
Client-Id	Mandatory	String	Current value assigned during registration.
Security-Token	Conditional	String	A security token to fulfil tasks of mutual authentication between DPE Client and DPE Server. MUST be present if a Security-Token was sent in the prior DPE1 Register-Response. If present, MUST be equal to the Security-Token sent in DPE1 Register-Response.
Group-ID	Mandatory	String	Identifies the group queried.
Message-ID	Mandatory	String	Identifies all the messages related to this query. Therefore the response to a given query MUST have the same *Message-ID*.

Table 6-8: DPE-2 Group-Query message elements

6.2.2.2 DPE-2 Group-Response

The " DPE-2 Group-Response" message is used by the DPE Client to advertise the values of a group's properties.

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier “Group-Response”
Client-Id	Mandatory	String	Current value assigned during registration.
Security-Token	Conditional	String	A security token to fulfil tasks of mutual authentication between DPE Client and DPE Server. MUST be present if a Security-Token was received in DPE1 Register-Response. If present, MUST be equal to the Security-Token received in DPE1 Register-Response.
Group-ID	Mandatory	String	Identifies the group queried.
Values	Mandatory	List of Structures	Contains the response to Group-Query message in the form of {name, value } for each property pertaining to the queried Group. The number of responses MUST be coherent with number of properties defined in the Group. Multiple responses may be included for properties with multiple values. See Table 6-7 for further details.
Message-ID	Mandatory	String	Identifies all the messages related to this query. Therefore the response to a given query MUST have the same *Message-ID*.

Table 6-9: DPE-2 Group-Response message elements

6.2.3 Creating Property Groups

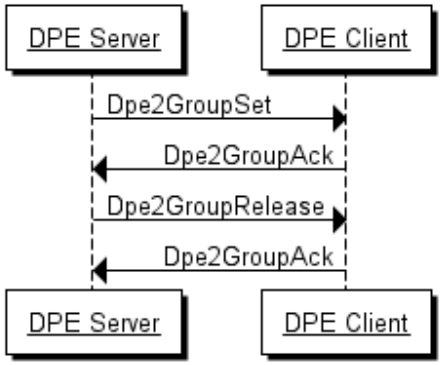

Figure 6-5: Setting and releasing of a group of properties from DPE Server, acknowledging from DPE Client

Message	Requirement	Direction
DPE-2 Group-Set	Mandatory	DPE Server → DPE Client
DPE-2 Group-Ack	Mandatory	DPE Server ← DPE Client
DPE-2 Group-Release	Mandatory	DPE Server → DPE Client

6.2.3.1 DPE-2 Group-Set

The "DPE-2 Group-Set" message is used by the DPE Server to set groups of properties.

Note: group establishment should always be successful, unless all requested properties are unsupported,

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier "Group-Set"
Client-Id	Mandatory	String	Current value assigned during registration.
Security-Token	Conditional	String	A security token to fulfil tasks of mutual authentication between DPE Client and DPE Server. MUST be present if a Security-Token was sent in the prior DPE1 Register-Response. If present, MUST be equal to the Security-Token sent in DPE1 Register-Response.
Group-ID	Mandatory	String	Identifies the group to be set.
Property-Names	Mandatory	String	Indicates the names of the properties defining the group.
Message-ID	Mandatory	String	Identifies all the messages related to this query. Therefore the response to a given query MUST have the same *Message-ID*.

Table 6-10: DPE-2 Group-Set message elements

6.2.3.2 DPE-2 Group-Ack

The "DPE-2 Group-Ack" message is used by the DPE Client to: acknowledge the setting or releasing of a group of properties.

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier "Group-Ack"
Client-Id	Mandatory	String	Current value assigned during registration.
Security-Token	Conditional	String	A security token to fulfil tasks of mutual authentication between DPE Client and DPE Server. MUST be present if a Security-Token was received in DPE1 Register-Response. If present, MUST be equal to the Security-Token received in DPE1 Register-Response.
Group-ID	Mandatory	String	Identifies the group that is set or released.
Status	Mandatory	String	Indicates if the group has been set or released successfully or not. Possible values "Success", "Fail". MUST be set to "Fail" if none of the requested properties are supported by the device or DPE Client.
Unsupported	Conditional	String	The names of all properties that were indicated, but are not supported by the device or DPE Client.

Message-ID	Mandatory	String	Identifies all the messages related to this query. Therefore the response to a given query MUST have the same *Message-ID*.

Table 6-11: DPE-2 Group-Ack message elements

6.2.3.3 DPE-2 Group-Release

The "DPE-2 Group-Release" message is used by the DPE Server to release a defined group or all the groups from a DPE Client.

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier "Group-Release"
Client-Id	Mandatory	String	Current value assigned during registration.
Security-Token	Conditional	String	A security token to fulfil tasks of mutual authentication between DPE Client and DPE Server. MUST be present if a Security-Token was sent in the prior DPE1 Register-Response. If present, MUST be equal to the Security-Token sent in DPE1 Register-Response.
Group-ID	Mandatory	String	Identifies the group to be released. If "*", all groups should be released.
Message-ID	Mandatory	String	Identifies the message.

Table 6-12: DPE-2 Group-Release message elements

6.2.4 Creating Property Reporting Policies

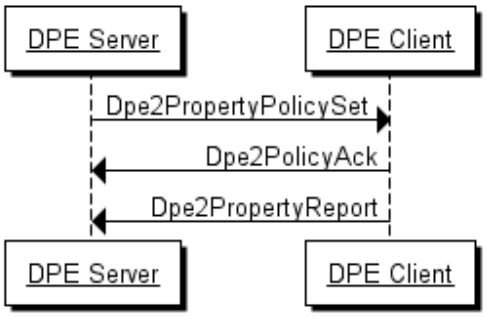

Figure 6-6: Flow of setting and acknowledging of policy for a property, and the property reporting

Message	Requirement	Direction
DPE-2 Property-Policy-Set	Mandatory	DPE Server → DPE Client
DPE-2 Policy-Ack	Mandatory	DPE Server ← DPE Client
DPE-2 Property-Report	Mandatory	DPE Server ← DPE Client

6.2.4.1 DPE-2 Property-Policy-Set

The “DPE-2 Property-Policy-Set” message is used by the DPE Server to set a policy to a property.

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier “Property-Policy-Set”
Client-Id	Mandatory	String	Current value assigned during registration.
Security-Token	Conditional	String	A security token to fulfil tasks of mutual authentication between DPE Client and DPE Server. MUST be present if a Security-Token was sent in the prior DPE1 Register-Response. If present, MUST be equal to the Security-Token sent in DPE1 Register-Response.
Policy-ID	Mandatory	String	Identifies the policy being defined.
Policy-Type	Mandatory	String	Identifies the type of policy (see “Policy Types and Parameters” table)
Policy-Parameters	Mandatory	String	Parameter of the policy (see “Policy Types and Parameters” table)
Property-Name	Mandatory	String	Identifies the property to which the policy is being assigned.
Duration	Optional	Integer	Gives the “time to live” for the Policy, in seconds.
Min-Report-Interval	Optional	Integer	Minimum interval in seconds between reports, for report throttling.
Message-ID	Mandatory	String	Identifies all the messages related to the policy setting. Therefore the acknowledgement MUST have the same *Message-ID*.

Table 6-13: DPE-2 Property-Policy-Set message elements

Policy-Type	Policy-Parameters	Description
Any-Change	n/a	report on any change to the property
Exact-Match	value to match	report on a change to a specific matched value
List-Match	comma-separated list of values to match	report on a change to one of a set of matched values in a comma-separated list
In-Range	range, as comma-separated list (lower value, upper value)	report if the property enters a range
Out-Of-Range	range, as comma-separated list (lower value, upper value)	report if the property exits a range
Property-Support	n/a	Report on any change to the set of supported properties, i.e. properties for which support begins or ends.
Schedule	Comma-separated list (start,stop,frequency)	report on a schedule: “start” and “stop” are either a “date-time” value or a “time-of-day” value, where “date-time” conforms to the "date-time" definition in [ISO8601]. In addition, an uppercase "T" character SHALL be used to separate date and time, and an uppercase "Z" character SHALL be present in the absence of a numeric time zone

		offset. For “time-of-day”, only the time field of the date-time is present. date-time values indicate the absolute start and end of property monitoring. time-of-day values indicate a daily schedule for property monitoring. frequency is the interval for reporting, in seconds.
Extension	unspecified string value	report on an implementation-specific extension (allows the extension to new triggers, e.g. report upon the startup of a specific client/application)

Table 6-14: Policy Types and Parameters

6.2.4.2 DPE-2 Policy-Ack

The “DPE-2 Policy-Ack” message is used by the DPE Client to acknowledge a policy set or released to a property or to a group

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier “Policy-Ack”
Client-Id	Mandatory	String	Current value assigned during registration.
Security-Token	Conditional	String	A security token to fulfil tasks of mutual authentication between DPE Client and DPE Server. MUST be present if a Security-Token was received in DPE1 Register-Response. If present, MUST be equal to the Security-Token received in DPE1 Register-Response.
Policy-ID	Mandatory	String	Identifies the policy being defined.
Status	Mandatory	String	Indicates if the policy has been defined or released successfully or not. Possible values are ”Success”, “Fail”. MUST be set to “Fail” if no policies could be successfully established.
Unsupported	Conditional	String	The names the properties that were indicated, but are not supported by the device or DPE Client.
Message-ID	Mandatory	String	MUST be the same Message-ID as in the corresponding Property-Policy-Set message or Group-Policy-Set message or Policy-Release message.

Table 6-15: DPE-2 Policy-Ack message elements

6.2.4.3 DPE-2 Property-Report

The “DPE-2 Property-Report” message is used by the DPE Client to advertise the value of the property on which the policy is assigned and the conditions of the policy are met.

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier “Property-Report”

Client-Id	Mandatory	String	Current value assigned during registration.
Security-Token	Conditional	String	A security token to fulfil tasks of mutual authentication between DPE Client and DPE Server. MUST be present if a Security-Token was sent in the prior DPE1 Register-Response. If present, MUST be equal to the Security-Token sent in DPE1 Register-Response.
Policy-ID	Mandatory	String	Identifies the policy associated to the response.
Value	Mandatory	Structure	Indicates the value of the property in the form of {name, value } for the property being reported upon. If the value is null, the property is no longer supported.
Message-ID	Mandatory	String	Identifies the message.

Table 6-16: DPE-2 Property-Report message elements

6.2.5 Creating Group Reporting Policies

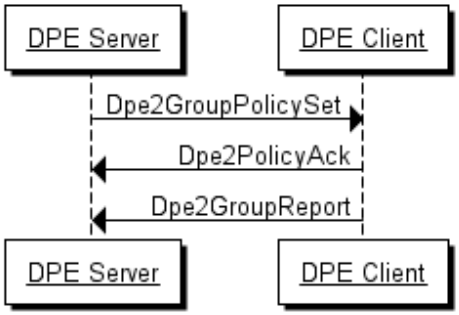

Figure 6-7: Flow of setting and acknowledging of policy for a group, and the properties reporting

Message	Requirement	Direction
DPE-2 Group-Policy-Set	Mandatory	DPE Server → DPE Client
DPE-2 Policy-Ack	Mandatory	DPE Server ← DPE Client
DPE-2 Group-Report	Mandatory	DPE Server ← DPE Client

6.2.5.1 DPE-2 Group-Policy-Set

The "DPE-2 Group-Policy-Set" message is used by the DPE Server to set a policy to a group.

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier "Group-Policy-Set"
Client-Id	Mandatory	String	Current value assigned during registration.

Security-Token	Conditional	String	A security token to fulfil tasks of mutual authentication between DPE Client and DPE Server. MUST be present if a Security-Token was sent in the prior DPE1 Register-Response. If present, MUST be equal to the Security-Token sent in DPE1 Register-Response.
Policy-ID	Mandatory	String	Identifies the policy being defined.
Policy-Type	Mandatory	String	Indicates the type of policy being defined (see "Policy Types and Parameters" table).
Policy-Parameters	Mandatory	String	Defines the parameters associated to the defined policy (see "Policy Types and Parameters" table).
Duration	Optional	Integer	Gives the "time to live" for the Policy, in seconds.
Min-Report-Interval	Optional	Integer	Minimum interval in seconds between reports, for report throttling.
Group-ID	Mandatory	String	Identifies the group to which the policy is being assigned.
Message-ID	Mandatory	String	Identifies all the messages related to the policy setting. Therefore the acknowledgement MUST have the same *Message-ID*.

Table 6-17: DPE-2 Group-Policy-Set message elements

6.2.5.2 DPE-2 Policy-Ack

The "DPE-2 Policy-Ack" message is used by the DPE Client to acknowledge a policy set to a group.

Please refer to section 6.2.4.2 for more information.

6.2.5.3 DPE-2 Group-Report

The "DPE-2 Group-Report" message is used by the DPE Client to report a group of properties when the conditions of the policy assigned on a group are met.

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier "Group-Report"
Client-Id	Mandatory	String	Current value assigned during registration.
Security-Token	Conditional	String	A security token to fulfil tasks of mutual authentication between DPE Client and DPE Server. MUST be present if a Security-Token was received in DPE1 Register-Response. If present, MUST be equal to the Security-Token received in DPE1 Register-Response.
Policy-ID	Mandatory	String	Identifies the policy associated to the response.
Group-ID	Mandatory	String	Identifies the group queried.
Values	Mandatory	List of Structures	Contains the properties in the form of {name, value } for each property pertaining to the Group. The number of responses MUST be coherent with number of properties defined in the Group.

Message-ID	Mandatory	String	Identifies the message.

Table 6-18: DPE-2 Group-Report message elements

6.2.6 Releasing Policies

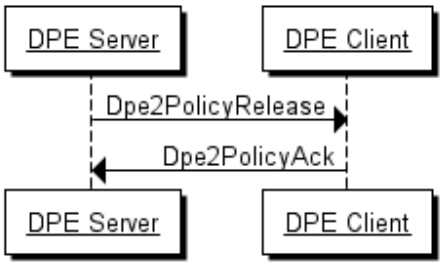

Figure 6-8: Flow of releasing of a policy, and the acknowledging

Message	Requirement	Direction
DPE-2 Policy-Release	Mandatory	DPE Server → DPE Client
DPE-2 Policy-Ack	Mandatory	DPE Server ← DPE Client

6.2.6.1 DPE-2 Policy-Release

The "DPE-2 Policy-Release" message is used by the DPE Server to release a defined Policy from a DPE Client (it can be a policy assigned either to an individual property or to a group).

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier "Policy-Release"
Client-Id	Mandatory	String	Current value assigned during registration.
Security-Token	Conditional	String	A security token to fulfil tasks of mutual authentication between DPE Client and DPE Server. MUST be present if a Security-Token was sent in the prior DPE1 Register-Response. If present, MUST be equal to the Security-Token sent in DPE1 Register-Response.
Policy-ID	Mandatory	String	Identifies the policy associated to the response. If "*", all policies should be released.
Message-ID	Mandatory	String	Identifies the message.

Table 6-19: DPE-2 Policy-Release message elements

6.2.6.2 DPE-2 Policy-Ack

The "DPE-2 Policy-Ack" message is used by the DPE Client to acknowledge a released Policy from a DPE Server (it can be a policy assigned either to an individual property or to a group).

Please refer to section 6.2.4.2 for more information.

6.3 Messages through DPE-3

The DPE-3 Interface allows a SP to request the value(s) of a single (or a group of) Device property (ies), to transmit policies and to transmit the responses of these requests and policies.

The DPE-3 Interface allows the SP to:

- Query the value of a single property or multiple properties
- Set and release groups of properties
- Query the values of a group's properties
- Set and release policies to a property or to a group

The DPE-3 Interface allows the DPE Server to respond:

- To the advertisement of the value of a single or of multiple properties
- To the advertisement of the values of a group's properties
- To the acknowledgement of the setting and releasing of a group of properties
- To the acknowledgement of the setting and releasing of a policy to a property or to a group
- With the value(s) of a property(ies) when the conditions of a policy are met

6.3.1 Querying Properties

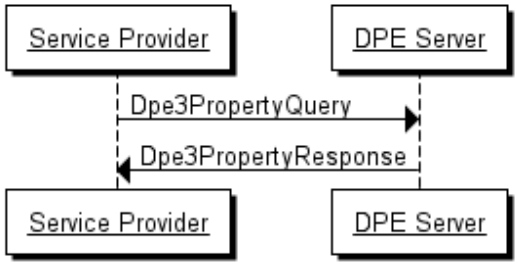

Figure 6-9: Flow of a Service Provider request for property (ies) to DPE Server, and its response.

Message	Requirement	Direction
DPE-3 Property-Query	Mandatory	Service Provider → DPE Server
DPE-3 Property-Response	Mandatory	Service Provider ← DPE Server

6.3.1.1 DPE-3 Property-Query

The "DPE-3 Property-Query" message is used by the SP to:

- Query the value of a single property;
- Query the values or multiple properties.

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier "Property-Query"
Property-Names	Mandatory	List of strings	Indicates the names of the properties queried. If "*", the current value of all supported properties should be reported.
Max-Age	Mandatory	Integer	Requested max age of the property values, in seconds.
Message-ID	Mandatory	String	Identifies all the messages related to this query. Therefore the response to a given query MUST have the same *Message-ID*.
Client-Id	Mandatory	String	Gives the ID of the client targeted for the request

Table 6-20: DPE-3 Property-Query message elements

6.3.1.2 DPE-3 Property-Response

The "DPE-3 Property-Response" message is used by the DPE Server to advertise the value of a single or of multiple properties.

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier "Property Response"
Values	Mandatory	List of Structures	Contains the response to Property-Query message in the form of {name, value } for each property queried. The number of responses MUST be coherent with number of properties queried in the Property-Query message.
Message-ID	Mandatory	String	MUST be the same Message-ID than in the corresponding Property-Query message.
Client-Id	Mandatory	String	Gives the ID of the client targeted for the request

Table 6-21: DPE-3 Property-Response message elements

6.3.2 Querying Property Groups

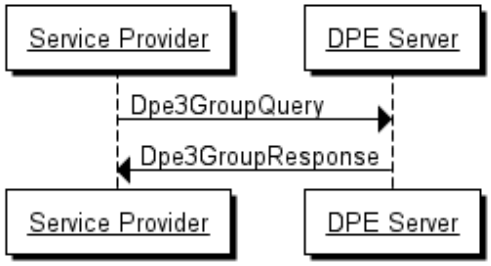

Figure 6-10: Flow of a SP request for group properties to DPE Server, and its answer

Message	Requirement	Direction
DPE-3 Group-Query	Mandatory	Service Provider → DPE Server
DPE-3 Group-Response	Mandatory	Service Provider ← DPE Server

6.3.2.1 DPE-3 Group-Query

The " DPE-3 Group-Query" message is used by the Server Provider to query the values of a group's properties**.**

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier “Group-Query”
Group-ID	Mandatory	String	Identifies the group queried.
Max-Age	Mandatory	Integer	Requested max age of the property values, in seconds.
Client-Id	Mandatory	String	Gives the ID of the client targeted for the request
Message-ID	Mandatory	String	Identifies all the messages related to this query. Therefore the response to a given query MUST have the same *Message-ID*.

Table 6-22: DPE-3 Group-Query message elements

6.3.2.2 DPE-3 Group-Response

The " DPE-3 Group-Response" message is used by the DPE Server to respond to the advertisement of the values of a group's properties

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier “Group-Response”

Group-ID	Mandatory	String	Identifies the group queried.
Values	Mandatory	List of Structures	Contains the response to Group-Query message in the form of {name, value } for each property pertaining to the queried Group. The number of responses MUST be coherent with number of properties defined in the Group.
Client-Id	Mandatory	String	Gives the ID of the client targeted for the request
Message-ID	Mandatory	String	Identifies all the messages related to this query. Therefore the response to a given query MUST have the same *Message-ID*.

Table 6-23: DPE-3 Group-Response message elements

6.3.3 Creating Property Groups

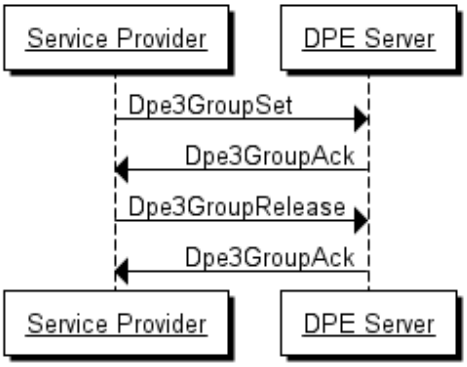

Figure 6-11: Setting of a group of properties from SP, acknowledging from DPE Server and release of the group.

Message	Requirement	Direction
DPE-3 Group-Set	Mandatory	Service Provider → DPE Server
DPE-3 Group-Ack	Mandatory	Service Provider ← DPE Server
DPE-3 Group-Release	Mandatory	Service Provider → DPE Server

6.3.3.1 DPE-3 Group-Set

The "DPE-3 Group-Set" message is used by the SP to set groups of properties.

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier "Group-Set"

Group-ID	Mandatory	String	Identifies the group queried.
Property-Names	Mandatory	String	Indicates the names of the properties defining the group.
Client-Id	Mandatory	String	Gives the ID of the client targeted for the request
Message-ID	Mandatory	String	Identifies all the messages related to this query. Therefore the response to a given query MUST have the same *Message-ID*.

Table 6-24: DPE-3 Group-Set message elements

6.3.3.2 DPE-3 Group-Ack

The "DPE-3 Group-Ack" message is used by the DPE Server to acknowledge the setting of a group of properties.

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier "Group-Ack"
Group-ID	Mandatory	String	Identifies the group queried.
Status	Mandatory	String	Indicates if the group has been defined successfully or not. Possible values "Success", "Fail".
Client-Id	Mandatory	String	Gives the ID of the client targeted for the request
Message-ID	Mandatory	String	Identifies all the messages related to this query. Therefore the response to a given query MUST have the same *Message-ID*.

Table 6-25: DPE-3 Group-Ack message elements

6.3.3.3 DPE-3 Group-Release

The "DPE-3 Group-Release" message is used by the SP to delete a defined Group from a DPE Client.

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier "Group-Release"
Group-ID	Mandatory	String	Identifies the group to be deleted.
Message-ID	Mandatory	String	Identifies the message.
Client-Id	Mandatory	String	Gives the ID of the client targeted for the request

Table 6-26: DPE-3 Group-Release message elements

6.3.4 Creating Property Reporting Policies

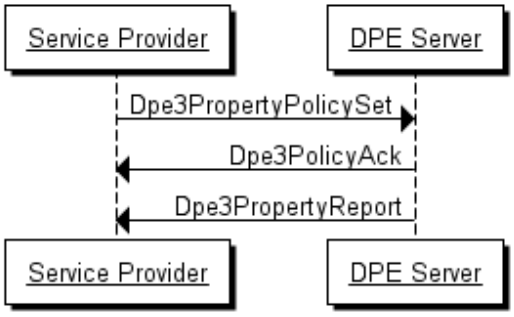

Figure 6-12: Setting of a group of properties from SP, acknowledging from DPE Server and release of the group.

Message	Requirement	Direction
DPE-3 Property-Policy-Set	Mandatory	Service Provider → DPE Server
DPE-3 Policy-Ack	Mandatory	Service Provider ← DPE Server
DPE-3 Property-Report	Mandatory	Service Provider ← DPE Server

6.3.4.1 DPE-3 Property-Policy-Set

The "DPE-3 Property-Policy-Set" message is used by the SP to set a policy to a property.

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier "Property-Policy-Set"
Policy-ID	Mandatory	String	Identifies the policy being defined.
Policy-Type	Mandatory	String	Identifies the type of policy (see "Policy Types and Parameters" table)
Policy-Parameters	Mandatory	String	Parameter of the policy (see "Policy Types and Parameters" table)
Property-Name	Mandatory	String	Identifies the property to which the policy is being assigned.
Client-Id	Mandatory	String	Gives the ID of the client targeted for the request
Duration	Optional	Integer	Gives the "time to live" for the Policy, in seconds.
Message-ID	Mandatory	String	Identifies all the messages related to the policy setting. Therefore the acknowledgement MUST have the same *Message-ID*.

Table 6-27: DPE-3 Property-Policy-Set message elements

6.3.4.2 DPE-3 Policy-Ack

The "DPE-3 Policy-Ack" message is used by the DPE Server to respond an acknowledge a policy set to a property

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier “Policy-Ack”
Policy-ID	Mandatory	String	Identifies the policy being defined.
Status	Mandatory	String	Indicates if the policy has been defined successfully or not. Possible values ”Success”, “Fail”.
Client-Id	Mandatory	String	Gives the ID of the client targeted for the request
Message-ID	Mandatory	String	MUST be the same Message-ID than in the corresponding Policy-Set message.

Table 6-28: DPE-3 Policy-Ack message elements

6.3.4.3 DPE-3 Property-Report

The “DPE-3 Property-Report” message is used by the DPE Server to report that the conditions of the policy are met and advertise the value of the property on which the policy is assigned.

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier “Property-Report”
Policy-ID	Mandatory	String	Identifies the policy associated to the response.
Value	Mandatory	String	Indicates the value of the property in the form of {name, value } for the property being reported upon.
Message-ID	Mandatory	String	Identifies the message.
Client-Id	Mandatory	String	Gives the ID of the client targeted for the request

Table 6-29: DPE-3 Property-Report message elements

6.3.5 Creating Group Reporting Policies

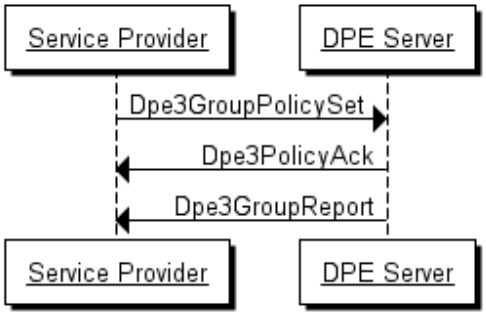

Figure 6-13: Creating a group reporting policy from SP

Message	Requirement	Direction
DPE-3 Group-Policy-Set	Mandatory	Service Provider → DPE Server
DPE-3 Policy-Ack	Mandatory	Service Provider ← DPE Server
DPE-3 Group-Report	Mandatory	Service Provider ← DPE Server

6.3.5.1 DPE-3 Group-Policy-Set

The "DPE-3 Group-Policy-Set" message is used by the SP to set a policy to a group.

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier "Group-Policy-Set"
Policy-ID	Mandatory	String	Identifies the policy being defined.
Policy-Type	Mandatory	String	Indicates the type of policy being defined (see "Policy Types and Parameters" table).
Policy-Parameters	Mandatory	String	Defines the parameters associated to the defined policy (see "Policy Types and Parameters" table).
Group-ID	Mandatory	String	Identifies the group to which the policy is being assigned.
Message-ID	Mandatory	String	Identifies all the messages related to the policy setting. Therefore the acknowledgement MUST have the same *Message-ID*.
Client-Id	Mandatory	String	Gives the ID of the client targeted for the request
Duration	Optional	Integer	Gives the "time to live" for the Group Policy, in seconds.

Table 6-30: DPE-3 Group-Policy-Set message elements

6.3.5.2 DPE-3 Policy-Ack

The "DPE-3 Policy-Ack" message is used by the DPE Server to acknowledge the group policy set.

See section 6.3.4.2 for more information.

6.3.5.3 DPE-3 Group-Report

The "DPE-3 Policy-Group-Report" message is used by the DPE Server to report that the conditions of the policy assigned on a group are met.

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier "Group-Report"

Policy-ID	Mandatory	String	Identifies the policy associated to the response.
Values	Mandatory	List of Structures	Contains the properties in the form of {name, value } for each property pertaining to the Group. The number of responses MUST be coherent with number of properties defined in the Group.
Group-ID	Mandatory	String	Identifies the group queried
Message-ID	Mandatory	String	Identifies the message.
Client-Id	Mandatory	String	Gives the ID of the client targeted for the request

Table 6-31: DPE-3 Group-Report message elements

6.3.6 Releasing Policies

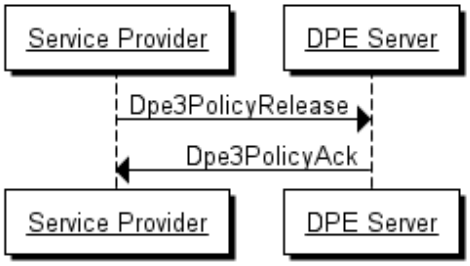

Figure 6-14: Releasing of a group of properties from SP, acknowledging from DPE Server.

Message	Requirement	Direction
DPE-3 Policy-Release	Mandatory	Service Provider → DPE Server

6.3.6.1 DPE-3 Policy-Release

The "DPE-3 Policy-Release" message is used by the SP to delete a defined Policy from a DPE Client (it can be a policy assigned either to an individual property or to a group).

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier "Policy-Release"
Policy-ID	Mandatory	String	Identifies the policy associated to the response.
Client-Id	Mandatory	String	Gives the ID of the client targeted for the request
Message-ID	Mandatory	String	Identifies the message.

Table 6-32: DPE-3 Policy-Release message elements

6.3.6.2 SP_Policy_Ack

The "Sp_Policy_Ack" message is used by the DPE Server to acknowledge the policy release.

See section 6.3.4.2 for more information.

6.3.7 Querying the Client ID

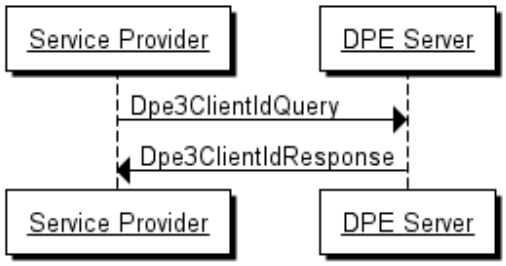

Figure 6-15: Query for DPE Client ID

Message	Requirement	Direction
DPE-3 Client-Id-Query	Mandatory	Service Provider → DPE Server
DPE-3 Client-Id-Response	Mandatory	DPE Server → Service Provider

6.3.7.1 DPE-3 Client-Id-Query

The "DPE-3 Client ID Query" message is used by the SP to determine the DPE Client ID associated with a user or Device.

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier "Client-Id-Query"
Query-Key	Mandatory	String	Key for the query, as defined by the DPE Service Provider, e.g. SIP Public Identity, MSISDN, IMEI, User Identity. The DPE Service Provider defines the options for query keys through unspecified means, e.g. through SP onboarding processes.
Message-ID	Mandatory	String	Identifies the message.

Table 6-33: DPE-3 Client-Id-Query message elements

6.3.7.2 DPE-3 Client-Id-Response

The "DPE-3 Client ID Response" message is used by the DPE Server to return the DPE Client ID associated with a user or Device.

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier “Client-Id-Response”
Client-Id	Mandatory	String	The DPE Client ID associated with the user. A null value indicates that there is no Client ID associated with the user.
Message-ID	Mandatory	String	Identifies the message. Same as the value in the related Client-Id-Query.

Table 6-34: DPE-3 Client-Id-Response message elements

6.3.8 Client ID Reporting

Figure 6-16: DPE Client ID Report

Message	Requirement	Direction
DPE-3 Client-Id-Report	Mandatory	DPE Server → Service Provider

6.3.8.1 DPE-3 Client-Id-Report

The “DPE-3 Client-Id-Report” message is used by the DPE Server to report an update to the association of a user with a Client ID.

Information Element	Req	Type	Description
Message-Type	Mandatory	String	Message Type Identifier “Client-Id-Report”
User-Identity	Mandatory	String	User Identity as defined by the DPE Service Provider, e.g. SIP Public Identity, MSISDN, IMEI, or generic user Identity. The DPE Service Provider defines the options for query keys through unspecified means, e.g. through SP onboarding processes. A null value indicates that there is no current user associated to the Client ID.
Client-Id	Mandatory	String	The DPE Client ID associated with the user. Note for multi-Device users, this will represent one of the currently valid Client IDs for the user. A null value indicates that there is no longer a Client ID associated with the user.

Previous-Client-Id	Optional	String	A DPE Client ID previously associated with the user, which is no longer valid.
Message-ID	Mandatory	String	Identifies the message.

Table 6-35 : DPE-3 Client-Id-Report message elements

7. Transport Protocol Bindings

7.1 DPE-1 and DPE-2 over OMA Push

The DPE Server SHALL support either the Push Access Protocol (PAP) [PUSH-PAP] or the Push-OTA protocol [PUSH-OTA] for point-to-point delivery of the following DPE-1 and DPE-2 Interface messages:

- DPE-1 Register-Command
- DPE-2 Property-Query
- DPE-2 Group-Query
- DPE-2 Group-Set
- DPE-2 Group-Release
- DPE-2 Property-Policy-Set
- DPE-2 Group-Policy-Set
- DPE-2 Policy-Release

For DPE Push messages, the DPE Server SHALL include the Push Application ID header "X-Wap-Application-Id: x-oma-application:dpe.ua".

When using PAP, the DPE Server SHALL submit DPE messages for delivery using the MIME content type "application/vnd.oma.dpe", and MAY support various target client address schemes, e.g. PLMN, USER, SIP URI, IP address, etc.

When delivering DPE messages via Push-OTA, the DPE Server SHALL encode the messages using the MIME content type "application/vnd.oma.dpec".

Push Clients in DPE supporting terminals SHALL support routing of Push messages with the Push Application ID header "X-Wap-Application-Id: x-oma-application:dpe.ua" to the DPE Client.

7.2 DPE-1 and DPE-2 over HTTP

DPE Clients and DPE Servers MUST support Hypertext Transfer Protocol version 1.1 (HTTP [RFC2616]) for the DPE-1 and DPE-2 Interface messages as shown in Table 7-1.

HTTP Method	Request Body	Response Code	Response Body
POST	DPE-1 Register-Request	200 OK	DPE-1 Register-Response
POST	DPE-1 Register-Ack	204 No Content	None
POST	DPE-2 Property-Response	204 No Content	None
POST	DPE-2 Group-Response	204 No Content	None
POST	DPE-2 Group-Ack	204 No Content	None
POST	DPE-2 Policy-Ack	204 No Content	None
POST	DPE-2 Policy-Property-Report	204 No Content	None

POST	DPE-2 Policy-Group-Report	204 No Content	None
POST	Any	200 OK	One or more of: DPE-1 Register-Command DPE-2 Property-Query DPE-2 Group-Query DPE-2 Group-Set DPE-2 Group-Release DPE-2 Property-Policy-Set DPE-2 Group-Policy-Set DPE-2 Policy-Release

Table 7-1: HTTP methods and request bodies for HTTP messaging

DPE Servers and DPE Clients MUST support DPE-1 and DPE-2 Interface messages formatted as entity-bodies with the application/vnd.oma.dpe MIME media type. The application/vnd.oma.dpe MIME media type is used when a single DPE Interface message is included in the HTTP request/response.

DPE Servers and DPE Clients MUST support DPE-1 and DPE-2 interface messages formatted as entity-bodies with the multipart/mixed media type. The multipart/mixed media type is used when multiple DPE Interface messages are concatenated in a single HTTP request/response.

DPE Clients MUST send all DPE-1 and DPE-2 Interface messages as HTTP POST method requests, including:

- the DPE Server address in the request line
- the Host request-header set to the hostname or IP address of the DPE Server
- the User-Agent request-header set to identify the host Device (e.g. “vendor-model/version”), and the name and version of the DPE Client as user agent initiating the request
- the Accept request-header with value “application/vnd.oma.dpe, multipart/mixed”
- the X-Wap-Profile request-header set to the URI of the User Agent Profile for the host Device
- the Accept-Encoding request-header with value per the supported HTTP compression encodings, i.e. deflate and/or gzip
- the Content-Length entity-header set to the length of the entity-body
- the Content-Type entity-header with value “application/vnd.oma.dpe” or “multipart/mixed”, as applicable
- the DPE-1 or DPE-2 message(s) as message-body

DPE Clients MUST send all DPE-1 and DPE-2 interface messages to the DPE Server configured in the device. If the DPE Server address and other connection profile information has been provisioned or pre-configured in the device, the DPE Client MUST:

- send the messages via the HTTP proxy, if specified
- utilize or establish a data connection according to the
    - network preferences, if specified
    - data connection details, if specified

DPE Servers MUST send all DPE-1 and DPE-2 Interface messages which are sent as responses to DPE Client messages as HTTP 200 OK responses, including:

- the ETag entity-header set to a unique value within the scope of the DPE Server
- the Content-Encoding entity-header set to the type of HTTP compression applied, if any
- the Content-Length entity-header set to the length of the entity-body
- the Content-Type entity-header with value “application/vnd.oma.dpe” or “multipart/mixed”, as applicable
- the DPE-1 or DPE-2 message(s) as message-body

8. DPE Operations

8.1 DPE-1 Operations

8.1.1 DPE Client Registration

Upon DPE Client startup, the DPE Client SHALL register with the applicable DPE Server. Registration provides the trigger for the DPE Server to establish or revalidate the current DPE settings for the DPE Client, and to inform related Service Providers of any change in the user and Client ID association, as necessary.

Upon a valid registration request from the DPE Client, the DPE Server SHALL provide a registration response with the Client ID for the Device. Note this may be a new Client ID or a previously-assigned Client ID, per the DPE Service Provider's deployment policy.

To ensure that it can correctly associate subsequent DPE Interface operations with the correct Device, the DPE Server SHALL retain the associated user identity (e.g. MSISDN or SIP URI), Device identity (e.g. IMEI or MAC address), and assigned Client ID. Other than as optionally provided in the registration request, the method of obtaining the network identity and Device identity is unspecified, but MAY include use of RADIUS or network proxy-forwarded HTTP headers.

At a lower level, the DPE Server MAY also know the IP address of the DPE Client, e.g. if provided by RADIUS or forwarded by a network proxy (e.g. in the conventional "x-forwarded-for" HTTP header). However a reason not to use the IP address as main identification of the DPE Client is that it usually changes due to dynamic IP address assignment (DHCP), at least for IPv4-based Devices. On the other hand, considering IPv6, a static IP address may be assigned to the Device. Thus for IPv6 supporting Devices, the DPE Server may be able to use the IP address as the Device identity for the DPE Client.

8.1.2 Mutual authentication

The need for of authentication of DPE Clients and DPE Servers is a deployment decision. DPE Clients SHALL be pre-configured with the necessary DPE Server access parameters and authentication credentials necessary to authenticate a DPE Server if required:

- For DPE Client-initiated transactions:
  - DPE Server address, formatted as a URI. This also allows the definition of the security level for DPE transactions through the URI scheme (HTTP or HTTPS).
  - Method of authentication
    - None: no server authentication is required
    - IP: DPE Server responses MUST be received from the IP address corresponding to the DPE Server Address, i.e. after name resolution if required. Note that not assumption about immunity from man-in-the-middle attack is assumed here; if that is a concern, TLS should be used for all transactions.
    - TLS: DPE Server address is a HTTPS URI, and the DPE Server MUST be validated through delivery of a valid TLS server certificate.
  - If TLS server authentication is required: the root certificate of the Certificate Authority that issued the server certificate.
- For DPE Server-initiated transactions:
  - Method of authentication: none, trusted server address
  - For "trusted server address", a list of trusted addresses in the applicable formats, e.g.
    - For DPE transactions over WAP Push, SME Source Address of the trusted DPE Server or Push Proxy Gateway. Note if the host Device supports OMA Push 2.2 (Push Security), the trusted addresses can be pre-configured for the Push Client or through the Push Security MO (Managed Object).

- For DPE transactions over SIP Push, SIP URI of the trusted DPE Server or Push Proxy Gateway

For DPE Server-initiated transactions that are included as part of responses to DPE Client-initiated transactions, the DPE Server will already have been authenticated if necessary.

DPE Servers SHALL be pre-configured with the necessary credentials to authenticate DPE Clients, if required. Note the choice of credentials is deployment specific, but MAY include:

- the user's network identity, e.g. MSISDN or SIP URI
- the Device identity, e.g. IMEI or MAC address
- source IP address, e.g. the DPE Server MAY be configured to allow requests from a range of IP addresses, without further authentication

8.2 DPE-2/3 Operations

8.2.1 Common operations

8.2.1.1 Mutual verification

In addition to the requirements described in section 8.1.2, the following apply.

When the DPE Server receives a message from the DPE Client, whatever it is, it has to check the following:

- That the Client ID is registered and corresponds to the stored user/Device on the DPE Server table.
- That the Client ID that identifies the DPE Client is an actor in this dialogue.

If something of the above fails, the DPE Server should send an error response to the DPE Client. In addition, DPE Server could retry a number of times the operation that is a configuration parameter.

The necessary access controls for the DPE-3 Interface will depend upon the DPE Server deployment policy of the DPE Service Provider. These MAY vary depending upon the type of business relationship between the DPE Service Provider and the 3rd-Party Service Provider (SP). Other than as described below, the access control methods are unspecified.

When the DPE Server receives a message from the SP, it SHOULD verify that this SP is authorized, by checking:

- That the SP ID is registered.
- That the SP IP address is also authorized and is registered for this SP ID.

If something of the above fails the DPE Server SHOULD ignore the request in order to avoid DoS attacks. It will depend on the policies of DPE Server to admit new connections. In any case, even for authorized SP, DPE Server SHOULD support DPE-3 Interface usage controls, e.g.:

- Limiting the number of connections/request from the same SP,
- Limiting the rate of requests (e.g. via high-consuming traffic policies), in order to avoid DoS attacks.

8.2.1.2 Message forwarding

The DPE Server SHALL forward requests received from the SP through the DPE-3 Interface to the DPE Client as required using the DPE-2 Interface, for example under the conditions:

- The SP wishes to query the value of a single property or multiple properties, for which the DPE Server has no valid cached value, e.g. the cached value's age exceeds the max-age parameter of the DPE-3 Property Query
- The SP wishes to set a property group, that does not yet exist for another SP

- The SP wishes to release a property group, and is the only SP depending upon that group definition
- The SP wishes to query the values of a group's properties, for which the DPE Server has no valid cached value
- The SP wishes to set policies to a property or to a group, that differs from the policy for another SP
- The SP wishes to release policies to a property or to a group, and is the only SP depending upon that particular policy

DPE Servers MAY support caching of DPE-2 reports. If so, the DPE Server SHALL consider a cached report value as valid for inclusion in a SP query response, if the value has been cached for less than the Max_Age indicated by the SP.

The decision whether a SP request via DPE-3 results in a related DPE-2 request will depend upon the DPE Server implementation (e.g. ability to optimize mapping between DPE-3 and DPE-2 transactions) and DPE Service Provider deployment policy (e.g. use of such optimizations according to the business relationship with the SP).

The DPE Server SHALL forward reports received from the DPE Client through the DPE-2 Interface, to the SP through the DPE-3 Interface, under the conditions:

- The report is related to a query by the SP
- The report is related to a policy set by the SP

After receiving a message from the DPE Client the DPE Server MUST do the following:

- Look for the SP ID that identifies the SP that opened the dialogue (identified with a DPE-2 Message ID) the DPE Client.
- Look for the DPE-3 Message ID that identifies the dialogue with el SP (this ID was set by the SP).
- Forward the result to the SP using the SP ID and the DPE-3 Message ID.

8.2.1.3 Message ID Control

At DPE-2 Interface the DPE Client MUST check that the Message_ID from the DPE Server is not repeated. The DPE Client only has to remember the Message IDs from previously programmed policy or policy group reports, so the Message ID of other ones requests from the DPE Server MUST be different.

If it detects a duplicate Message ID in a request, the DPE Client SHALL send the related response with the status “fail” and reason “Duplicate Message ID”.

At DPE-3 Interface, the SP sets the Message ID that identifies the dialogue with the DPE Server.

The DPE Server has to check that the new Message ID that identifies a new dialogue is not already used in another open dialogue with the SP. If the Message ID is repeated, the DPE Server Server SHALL send the related response with the status “fail” and reason “Duplicate Message ID”.

The DPE Server has to register the dialogue with the pair [SP ID – Message ID], in order to distinguish other open dialogues established with others SP’s that could use the same Message ID. DPE Servers MAY support use of the Message ID provided by the SP in DCD-3 messages as the DPE-2 Message ID for the related messages. If so, and a SP-provided Message ID is the same as that provided by another SP, the DPE Server SHALL ensure uniqueness of the DPE-2 Message ID, e.g. by prefixing/suffixing it with a unique value.

8.2.1.4 Delete Message ID

When the DPE Server forwards the DPE Client response to the SP, it can delete the DPE-3 Message ID used to communicate with the SP because that dialogue is finished.

There are two exceptions to this behaviour:

- When the message forwarded is a policy report,
- When the message forwarded is a policy group report.

In those cases the same DPE-3 Message ID will be used in possible future reports so it cannot be deleted until the policy is released.

The DPE Client has to do the same with the DPE-2 Message ID, when it responds to the DPE Server query it has to delete the DPE-2 Message ID taking into account two exceptions of above (policy or policy group report Message ID to be deleted when the policy is released).

8.2.1.5 Flow control mechanism

The DPE Server MUST wait until the DPE Client responds to an existing request, or a timeout occurs on the DPE Client response, before sending a new request.

The DPE Server SHOULD ensure that for each SP, the number of DPE-3 requests do not exceed a reasonable threshold of requests per time window, as established by DPE Service Provider policy.

8.2.2 Device Change

Device changes require special handling by the DPE Enabler. Device change use cases include any case in which there has been a change in user identity or Device identity. Such use cases include:

- A user moves their SIM (or other form of user identity) from one Device to another Device, which was previously not associated with this user (from the DPE Server's perspective); note this is the same case as when a user changes identity directly on the Device
- A multi-Device user uses a Device which was previously not associated with this user (from the DPE Server's perspective)
- A new user (previously unknown to the DPE Server) is using a new Device (also previously unknown to the DPE Server): this is a special case also addressable as a Device change

Upon detection of a change in Device or user identity during registration, the DPE Server SHALL ensure that the current DPE Client has the current DPE settings relevant to the user by:

- Allocating a new Client ID as necessary to avoid conflict with earlier allocated Client IDs for the Device or user
- Resetting any DPE settings for property groups by sending the "Group-Release" message with Group-ID set to "*".
- Resetting any DPE settings for reporting policies by sending the "Policy Release" message with Policy-ID set to "*".
- Establishing or restoring any relevant property groups by sending the "Group-Set" message for all defined property groups.
- Establishing or restoring any relevant property reporting policies by sending the "Property-Policy-Set" message for all defined property reporting policies.
- Establishing or restoring any relevant group reporting policies by sending the "Group-Policy-Set" message for all defined group reporting policies.

Note that “relevant” in the previous paragraph refers to any implementation/deployment-specific criteria that the DPE Server uses to determine which property groups and policies apply for the user in the current Device. For example, these may be based upon groups/policies previously defined, as mapped to the current Device capabilities, or to new groups/policies applicable for the current Device. Further, it is an implementation decision if/how the DPE Server associates any new/restored groups and policies to those previously requested by Service Providers.

8.2.3 Client ID Change

Upon any change in the Client ID associated with a user, the DPE Server SHALL ensure that Service Providers are informed as necessary about the change, by sending the “Client-Id-Report” message.

8.2.4 Offline Operations

DPE Clients may be offline (inaccessible) when an SP defines property groups and reporting policies. In these cases the DPE Server cannot deliver the property group and reporting policy definitions to the DPE Client until the DPE Client becomes online (accessible) again. It is unspecified how the DPE Server determines the offline/online status of DPE Clients.

In some cases, e.g. adaptation of the DPE Enabler for support of Devices which do not have a DPE Client, the DPE Server MAY allocate a Client ID on behalf of the user, and support all DPE-3 operations based upon Device properties obtained from unspecified sources.

DPE Servers SHOULD support the definition of property groups and reporting policies by SP’s, when the target DPE Client is offline. For property groups and reporting policies defined when the DPE Client is offline, DPE Servers SHALL update the DPE Client with the property groups and reporting policies when the DPE Client becomes online.

If the DPE Client is offline when an SP query is received, the DPE Server SHALL provide the requested properties from its cached values, if available and valid.

9. Quality of Experience Considerations

9.1 Privacy

DPE Clients and DPE Servers SHALL honor privacy requirements of the user and DPE Service Provider, i.e. restrict access to specific device properties when not allowed per policy.

Note: determination and management of policies is implementation-specific.

If a query/policy-based report includes a property that is not allowed to be disclosed, the DPE Client or DPE Server SHALL respond with a null value for that property, i.e. represent it as unsupported.

9.2 Usability

The DPE Client MUST exchange Dynamic Device Properties in a manner that is unobtrusive to the user and does not impact the usability of any service being consumed.

9.3 Security

DPE Servers SHOULD minimize risk of spoofing the Client-Id by using hashed values assigned with limited lifetime. If the Client-Id is assigned for long periods, DPE Servers SHOULD assign a Security-Token with a shorter lifetime as an additional measure against spoofing, i.e. as an authentication token in messages either in place of, or in addition to, the Client-Id.

9.4 DPE Core Vocabulary

The effectiveness of the DPE Enabler depends on the value and efficiency of the information to be communicated. Hence, it is very important to have a set of properties that include all dynamic aspects of the Device which are of importance in content adaptation, the offering of new or adapted services, or those that in general lead to a better user experience, etc.

The overall device capability is represented as a set of Device Properties. The Device Properties may be dynamic or static.

When used in transactions between the DPE Client and DPE Server, e.g. queries for specific properties and responses with the current values, the properties are identified by name. While the properties are related to specific device component classes (e.g. hardware) and aspects (e.g. camera), for efficiency purposes the properties are exchanged between the DPE Client and DPE Server as a simple list of property names, thus the names are required to be unique.

The vocabulary below represents a basic set of properties. DPE Servers and DPE Clients SHOULD support extensibility of the property name set, e.g. for alignment with other standardized device capability vocabularies.

9.5 Grouping of capabilities/ properties (Device Division)

DPE Device capabilities are classified into four groups. This division is similar to other Device description solutions, such as [UAProf]. According to the dynamic properties to be communicated, the Device could be seen as formed by the following four modules:

- Hardware
- I/O
- Connectivity
- Software

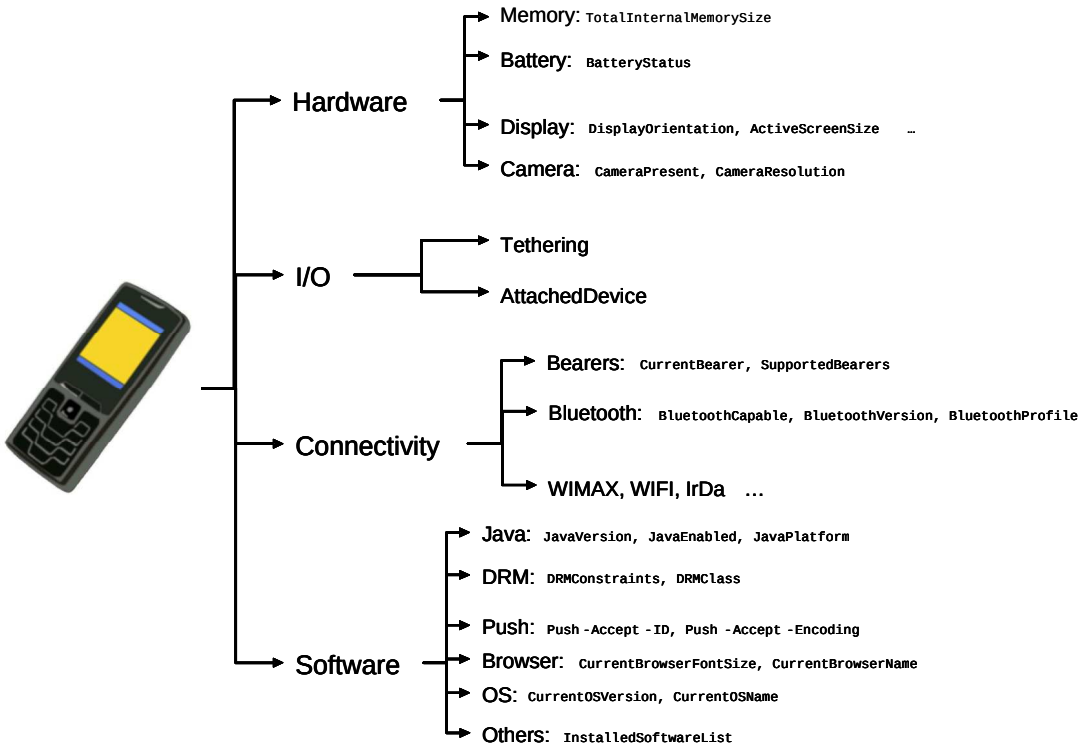

Figure 9-1: Device division for property grouping.

The properties defined for the DPE Core Vocabulary are intended to be as consistent as possible with [UAProf] and other standard schemas (e.g. the W3C's Delivery Context Ontology [W3C DCO]), with the following general exceptions:

- The properties include dynamic attributes not supported by the static property set defined in UAProf
- For simplicity, the properties in some cases summarize the detailed properties of other schemas (e.g. "BluetoothCapable").

OMA manages the DPE vocabulary through processes defined on the OMA website. Additional properties can be established through the process defined by OMA.

9.5.1 Hardware

The Hardware class will contain all the capabilities / properties related with physical elements of the Device. The hardware could be also divided into different subsections, such as Memory, Battery, Display, and Camera.

9.5.2 I/O

The I/O class gathers the capabilities / properties about the connection of the Device with other Devices, for example, in the case when a Device is tethering another one.

9.5.3 Connectivity

The Connectivity class gathers the capabilities / properties giving information about the different connectivity the Device is capable to provide. These could include the current bearer used for mobile communication, whether the WLAN is enabled or not, etc…

9.5.4 Software

The Software class gathers all the capabilities / properties related to the software installed on the Device (e.g. Operating System, firmware, codecs, client applications, etc…).

9.6 Core Vocabulary List

Device Capability	Properties	Description	Data Type	Cardinality	Unit of measure	UAProf2
Hardware Class						
Internal Memory	BuiltinTotalMemorySize	Total internal memory built-in the Device.	Integer	1	Bytes (B)	N/A
	BuiltinAvailableMemorySize	Available internal memory built-in the Device.	Integer	1	Bytes (B)	N/A
External Memory	ExtensionTotalMemorySize	Total removable memory in the Device.	Integer	1	Bytes (B)	N/A
	ExtensionAvailableMemorySize	Available removable memory in the Device.	Integer	1	Bytes (B)	N/A
Battery	BatteryLevel	Informs about the current battery level	Integer	1	%	N/A
	BatteryBeingCharged	This property indicates if the battery is being charged or not	Boolean	1		N/A
Display	BrowserUsableDisplayPixelsX	Current usable width on the screen to display browser content (without the scrollbars)	Integer	1	Pixels	N/A
	BrowserUsableDisplayPixelsY	Current usable height on the screen to display browser content (without the scrollbars)	Integer	1	Pixels	N/A
	TotalDisplayPixelsX	Indicates the total screen width currently available	Integer	1	Pixels	N/A
	TotalDisplayPixelsY	Indicates the total screen height currently available	Integer	1	Pixels	N/A
	DisplayOrientation	The current orientation of the Display: one of “portrait”, “landscape”	String	1		N/A
	PointingResolution	The precision with which pointing can be carried out on a device: one of "character", "line", "pixel"	String	1		PointingResolution
Camera	PrimaryCameraEnabled	States whether the Device has its Primary Camera Enabled.	Boolean	1		N/A
	PrimaryCameraPresent	Indicates if the Device has (at least) one camera	Boolean	1		N/A
	SecondaryCameraEnabled	States whether the Device has its Secondary Camera Enabled.	Boolean	1		N/A
	SecondaryCameraPresent	Indicates if the Device has a second camera	Boolean	1		N/A
	PrimaryCameraResolution	Specifies the current resolution of primary camera	Integer	1	Pixels	N/A
	SecondaryCameraResolution	Specifies the current resolution of secondary camera	Integer	1	Pixels	N/A
Speaker	SpeakerPresent	Indicates if the Device has a speaker	Boolean	1		N/A
I/O Class						
Tethering	Tethering	States whether this Device is actuating "tethered" to another Device, giving the latter some kind	Boolean	1		N/A

		of wireless connection				
OutputModality	OutputModality	States what kind of modality is used to output content on the Device: one or more of “visual”, “audible”, “tactile”	String	1		N/A
InputModality	InputModality	States what kind of modality is used to input content on the Device: one or more of “visual”, “audible”, “tactile”	String	1		N/A
AudioInputEncoder	AudioInputEncoder	Specifies if the Device has one or more audio input encoders Enabled.	Boolean	1		N/A
InputPeripheral	InputPeripheral	Specifies if the Device has one or more input peripherals Enabled.	Boolean	1		N/A
OutputPeripheral	OutputPeripheral	Specifies if the Device has one or more output peripherals Enabled.	Boolean	1		N/A
Ringtone	Ringtone	Specifies if the Device allows ringing.	Boolean	1		N/A
AttachedDevice	AttachedDevice	Stated whether another Device is attached to the current one	Boolean	1		N/A
Keyboard	Keyboard	Specifies if the Device has a keyboard Enabled.	Boolean	1		Keyboard
Ports	USB	Specifies if the device has USB port(s)	Boolean	1		N/A
	SerialPort	Specifies if the device has serial port(s)	Boolean	1		N/A
	ParallelPort	Specifies if the device has parallel port(s)	Boolean	1		N/A
Navigation	GPS	Specifies if the device has GPS support	Boolean	1		N/A
	GNSS	Specifies if the device has GNSS support	Boolean	1		N/A
Connectivity Class						
Bearers	ActiveNetworkBearers	An unordered, comma-separated list of bearers via which the device has an active network connection: one or more of “EDGE”, “GPRS”, “UMTS”, “HSDPA”, “HSUPA”, “HSPA+”, “LTE”, “WLAN”, “PACKET”, “WCDMA”, “CDMA”, “TD-SCDMA”	String	1		
	AvailableNetworkBearers	An unordered, comma-separated list of bearers for which the device is within coverage, and capable of connecting: one or more of “EDGE”, “GPRS”, “UMTS”, “HSDPA”, “HSUPA”, “HSPA+”, “LTE”, “WLAN”, “PACKET”, “WCDMA”, “CDMA”, “TD-SCDMA”	String	1		
	SupportedNetworkBearers	A comma-separated list of bearers supported by the Device: one or more of “EDGE”, “GPRS”, “UMTS”, “HSDPA”, “HSUPA”, “HSPA+”, “LTE”, “WLAN”, “PACKET”, “WCDMA”, “CDMA”, “TD-SCDMA”	String	1		SupportedBearers
Bluetooth	BluetoothEnabled	Indicates if the Device has Bluetooth connectivity enabled	Boolean	1		N/A
	ActiveBluetoothProfile	Active Bluetooth profile: a value as defined in the Bluetooth specification, or null (no active profile)	String	1		BluetoothProfile
	BluetoothStatus	Current status of the Bluetooth bearer: one of “off”, “on”, “discoverable”, “paired”	String	1		N/A
	BluetoothVersion	Indicates the Bluetooth version supported: version number, or null (no version)	String	1		SupportedBluetoothVersion
Infrared	InfraredCapable	Indicates if the Device has	Boolean	1		

		Infrared hardware.				
Software Class						
Java	JavaPackage	Java packages that are available in the active Java runtime environment. Comma-separated list of package names.	String	1		JavaPackage
	JavaPlatform	A comma-separated list of Java platforms and profiles installed in the device. Each item in the list is a name token describing compatibility with the name and version of the java platform specification or the name and version of the profile specification name (if profile is included in the device) Examples: "PersonalJava", "CLDC", "MIDP"	String	1		JavaPlatform
	JavaProtocol	Details about protocols supported by the device over and above those that are part of the standard Java profile indicated and the versions of these additional protocols Examples: "sms/1.0", "file/1.0"	String	1		JavaProtocol
	JavaScriptEnabled	Indicates whether JavaScript is supported and enabled in the browser.	Boolean	1		JavaScriptEnabled
	JavaScriptVersion	Version of the JavaScript language supported by the browser.	String	1		JavaScriptVersion
Push	PushAcceptAppID	A comma-separated list of Push applications the device supports, where each item in the list is an application-id on absoluteURI format as specified in [PushMsg]. A wildcard ("*") may be used to indicate support for any application. The authoritative values for this attribute could be found at http://www.openmobilealliance.org/tech/omna/omna-push-app-id.htm Examples: "x-wap-application:wml.ua", "x-wap-application:emn.ua, "x-wap-application:*"	String	1		Push-Accept-AppID
Browser	BrowserActiveName	Name of the current browser being used	String	1		N/A
	BrowserVersion	Version of the current browser being used	String	1		BrowserVersion
OS	OSName	The name of the OS running on the Device	String	1		OSName
	OSVersion	The version of the OS running on the system	String	1		OSVersion
	OSVendor	The Vendor of the OS	String	1		OSVendor
Hardware	Vendor	The device vendor.	String	1		
	Model	The device model.	String	1		
	Version	The device version.	String	1		
Software	UriSchemas	Supported URI Schemes: a comma-separated list of one or more of “http”, “https”, “tel”, “mailto”, “smsto”, “mmsto”, or optional additional schemas.	String	1		
	MimeTypes	Supported MIME types, in additional to that disclosed in UAProf: IANA-registered MIME types, or null.	String	1		

Table 9-1: DPE Core Vocabulary

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-TS-DPE-V1_0-20110705-A	05 Jul 2011	Status changed to Approved by TP: OMA-TP-2011-0225-INP_DPE_V1_0_ERP_for_Final_Approval

Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [SCRRULES].

B.1 SCR for XYZ Client

Item	Function	Reference	Requirement
DPE-C-001-M	DPE-1 Client initiated register process	6.1.1, 8.1.1	
DPE-C-002-M	DPE-1 Server initiated register process	6.1.2	
DPE-C-003-M	DPE-2 Property Query	6.2.1	
DPE-C-004-M	DPE-2 Group Query	6.2.2	
DPE-C-005-M	DPE-2 Group Set and Release	6.2.3	
DPE-C-006-M	DPE-2 Property Policy Set and Policy Property Report	6.2.4	
DPE-C-007-M	DPE-2 Group Policy Set and Policy Group Report	6.2.5	
DPE-C-008-M	DPE-2 Policy Release	6.2.6	
DPE-C-009-M	Policy Types and Parameters	6.4	
DPE-C-010-M	DPE-1 and DPE-2 Over WAP Push	7.1	
DPE-C-011-M	DPE-1 and DPE-2 Over HTTP	7.2	
DPE-C-012-M	Request Validation	8.1.2, 8.2.1.1	
DPE-C-013-M	Message ID Control	8.2.1.3	
DPE-C-014-M	Delete Message ID	8.2.1.4	
DPE-C-015-M	Device Change	8.2.2	
DPE-C-016-M	Client ID Change	8.2.3	
DPE-C-017-M	DPE Core Vocabulary	9	

B.2 SCR for XYZ Server

Item	Function	Reference	Requirement
DPE-S-001-M	DPE-1 Client initiated register process	6.1.1, 8.1.1	
DPE-S-002-M	DPE-1 Server initiated register process	6.1.2	
DPE-S-003-M	DPE-2 Property Query	6.2.1	
DPE-S-004-M	DPE-2 Group Query	6.2.2	
DPE-S-005-M	DPE-2 Group Set and Release	6.2.3	
DPE-S-006-M	DPE-2 Property Policy Set and Policy Property Report	6.2.4	
DPE-S-007-M	DPE-2 Group Policy Set and Policy Group Report	6.2.5	
DPE-S-008-M	DPE-2 Policy Release	6.2.6	
DPE-S-008-M	DPE-3 Property Query	6.3.1	
DPE-S-010-M	DPE-3 Group Query	6.3.2	
DPE-S-011-M	DPE-3 Group Set and Release	6.3.3	
DPE-S-012-M	DPE-3 Property Policy Set and Policy Property Report	6.3.4	
DPE-S-013-M	DPE-3 Policy Release	6.3.5	
DPE-S-014-M	DPE-3 Client ID Query	6.3.6	
DPE-S-015-M	DPE-3 Client ID Report	6.3.7	
DPE-S-016-M	Policy Types and Parameters	6.4	
DPE-S-017-M	DPE-1 and DPE-2 Over WAP Push	7.1	

Item	Function	Reference	Requirement
DPE-S-018-M	DPE-1 and DPE-2 Over HTTP	7.2	
DPE-S-019-M	Request Validation	8.1.2, 8.2.1.1	
DPE-S-020-M	Message Forwarding	8.2.1.2	
DPE-S-021-M	Message ID Control	8.2.1.3	
DPE-S-022-M	Delete Message ID	8.2.1.4	
DPE-S-023-M	Flow Control Mechanism	8.2.1.5	
DPE-S-024-M	Device Change	8.2.2	
DPE-S-025-M	Client ID Change	8.2.3	
DPE-S-026-M	Offline Operations	8.2.4	
DPE-S-027-M	DPE Core Vocabulary	9	

Appendix C. A Compact Binary Representation of DPE-1 and DPE-2

C.1 Introduction

WBXML 1.3 [WBXML] is a simple method that allows compacting XML documents in a lossless manner. A WBXML decoder processes a WBXML encoded document by interpreting it byte-by-byte. Some bytes represent decoding instructions, some represent XML element start tags, attribute names or attribute values. The decoding process is stateful. The decoder maintains one global state, which determines whether it is processing elements, or attributes. Within each state, the decoder maintains an independent notion of a selected code page.

C.1.1 Associating XML Documents with WBXML Token Values

An external typing system has to be used to associate XML documents with WBXML token values.

If the document is transported by WSP or HTTP, the media type has to be used. Since the token values are associated with the document media type, and not a particular version of the document type definition, the tokeniser is independent of the document type version; and can tokenise any version of the document type. To ensure compatibility between different versions of user-agents and tokenisers, the user-agent has to support both the binary token value and the literal value for all tags, attribute names, and attribute values.

C.1.2 WBXML Document Format

A WBXML document construct as a sequence of the following elements:

1. WBXML version used.
2. The Document Public Identifier
3. Character set used to encode strings
4. A string table.
5. WBXML Body containing the encoded XML body

C.2 DPE implementation

DPE Servers and DPE Clients SHALL support use of the compact encoding for downlink messages (i.e. DPE Server to DPE Client) delivered via WAP Push over the SMS bearer.

DPE Servers and DPE Clients MAY support use of the compact encoding for messages sent over HTTP/HTTPS. DPE interface binding to HTTP is described in section 7.2.

DPE Clients that support compact encoding for messages sent over HTTP/HTTPS SHALL:

- announce the support by inclusion of the MIME type application/vnd.oma.dpec in the Accept header of DPE messages
- send all DPE-1 messages in un-encoded form.
- send DPE-2 messages in encoded form only if the DPE Server has acknowledged encoded message support.

DPE Servers that support compact encoding for messages sent over HTTP/HTTPS SHALL:

- announce the support by inclusion of Encoding_Supported attribute set to “True” in the “Register-Response” message
- send all DPE-1 message responses in un-encoded form
- send DPE-2 message responses in encoded form only if the DPE Client has acknowledged encoded message support.

C.2.1 DPE Media Type

The following Media Types are defined by DPE.

Textual form (XML): application/vnd.oma.dpe

Tokenised form (WBXML): application/vnd.oma.dpec

When used with WAP Push over the SMS the token values are assigned:

Media-Type	WSP Assigned number
application/vnd.oma.dpec	0x14
application/vnd.oma.dpe	0x14

Table 9-2 Media Type tokens assignment

Editor's note: Media Type registration is TBD.

C.2.2 WBXML version

For this specification, WBXML 1.3 is used. This is represented by the version byte value 0x03 as specified in [WBXML].

C.2.3 WBXML Document Public Identifier

The Document Public Identifier is "-//OMA//DPE 1.0//EN" is registered by OMNA, and is represented as a single byte value TBD.

Editor's note: Document Public Identifier registration and value are TBD.

C.2.4 Character set

The character set to be used is UTF-8, represented as one byte value 0x6a.
The value of the WBXML Charset field is the MIBEnum value assigned by the IANA for the character encoding ((see [IANACharset])).

C.2.5 String table

The string table is always empty for DPE. This is represented as one byte value 0x00 (representing a string table length of 0).

C.2.6 WBXML Body

Following the above header's parameters, there is a body. This is a sequence of codes from the code pages, signaling processing instructions, element tags, attribute names and values. The decoder starts with tag code page 0, and attribute code page 0.

The next sections contain tables of the DPE assigned numbers. OMA is responsible for administering the values. The following describes the process requirements for administration of DPE assigned numbers:

- New entities SHALL be added at the end of any tables.
- When removing an entity in one of those tables, the assigned number SHALL be deprecated and it SHALL NOT be re-used for another entity.
- If the encoding rules of an entity need to be changed, a new entity SHALL be created.

C.2.6.1 Element tag Tokens

The following token codes represent tags in code page zero (0). All numbers are in hexadecimal.

Tag Name	Token [0x]
GroupResponse	5
GroupAck	6
GroupPolicySet	7
GroupQuery	8
GroupRelease	9
GroupSet	A
PolicyAck	B
GroupReport	C
PropertyReport	D
PolicyRelease	E
PropertyPolicySet	F
PropertyQuery	10
PropertyResponse	11
RegisterAck	12
RegisterCommand	13
RegisterRequest	14
RegisterResponse	15

Table 9-3 Element Tag Tokens

C.2.6.2 Attribute Start Tokens

The following token codes represent the start of an attribute in code page zero (0). All numbers are in hexadecimal.

Attribute Name	Attribute Value Prefix	Token [0x]
COMMON ATTRIBUTES		
xsi:schemaLocation		5
xsi:schemaLocation	"http://www.openmobilealliance.com/oma-dpe/1.0"	6
xmlns:DPE		7

Attribute Name	Attribute Value Prefix	Token [0x]
xmlns:DPE	"http://www.openmobilealliance.com/oma-dpe/1.0"	8
xmlns:xsi	"http://www.w3.org/2001/XMLSchema-instance"	9

Table 9-4 Common Attribute Start Tokens

Attribute Name	Attribute Value Prefix	Token [0x]
Device-Identity		A
DPE-ClientId		B
DPE-Version	1.0	C
Duration		D
Encoding-Supported	False	E
Encoding-Supported	True	F
Group-Id		10
IMEI		11
Max-Age		12
Message-Id		13
Message-Type	Group-Ack	14
Message-Type	Group-Policy-Set	15
Message-Type	Group-Query	16
Message-Type	Group-Release	17
Message-Type	Group-Response	18
Message-Type	Group-Set	19
Message-Type	Policy-Ack	1A
Message-Type	Group-Report	1B
Message-Type	Property-Report	1C
Message-Type	Policy-Release	1D
Message-Type	Property-Policy-Set	1E
Message-Type	Property-Query	1F
Message-Type	Property-Response	20
Message-Type	Register-Ack	21
Message-Type	Register-Command	22

Attribute Name	Attribute Value Prefix	Token [0x]
Message-Type	Register-Request	23
Message-Type	Register-Response	24
MSISDN		25
Policy-Parameters		26
Policy-Type	Any-Change	27
Policy-Type	Exact-Match	28
Policy-Type	Extension	29
Policy-Type	In-Range	2A
Policy-Type	List-Match	2B
Policy-Type	Out-Of-Range	2C
Policy-Type	Schedule	2D
Property-Name		2E
Property-Names		2F
Property-Value		30
Security-Token		31
Status	Fail	32
Status	Success	33
User-Identity		34

Table 9-5 Message Attribute Start Tokens

C.2.6.3 Attribute Value Tokens

The following token codes represent attribute values in code page zero (0). All numbers are in hexadecimal.

Editor's note: Tokens will be assigned once the schema is completed.

Attribute Value	Token [0x]
.com/	85
.edu/	86
.net/	87
.org/	88
.mobi/	89

Attribute Value	Token [0x]
www.	8A
.com	8B
.co	8C
wap	8D
http://	8E
http://www.	8F
https://	90
https://www.	91
false	92
true	93
ActiveBluetoothProfile	94
AttachedDevice	95
AvailableNetworkBearers	96
BatteryBeingCharged	97
BatteryLevel	98
BluetoothCapable	99
BluetoothStatus	9A
BluetoothVersion	9B
BrowserActiveName	9C
BrowserUsableDisplayPixelsX	9D
BrowserUsableDisplayPixelsY	9E
BrowserVersion	9F
BuiltinAvailableMemorySize	A0
BuiltinTotalMemorySize	A1
DisplayOrientation	A2
ExtensionAvailableMemorySize	A4
ExtensionTotalMemorySize	A5
GPS	A6
GNSS	A7
InfraredCapable	A8

Attribute Value	Token [0x]
InputModality	A9
JavaPackage	AA
JavaPlatform	AB
JavaProtocol	AC
JavaScriptEnabled	AD
JavaScriptVersion	AE
Keyboard	AF
MimeTypes	B0
Model	B1
OSName	B2
OSVendor	B3
OSVersion	B4
OutputModality	B5
ParallelPort	B6
PointingResolution	B7
PrimaryCameraEnabled	B8
PrimaryCameraPresent	B9
PrimaryCameraResolution	BA
PushAcceptAppID	BC
SecondaryCameraEnabled	BD
SecondaryCameraPresent	BE
SecondaryCameraResolution	BF
SerialPort	C0
Speaker	C1
SupportedNetworkBearers	C2
Tethering	C3
TotalDisplayPixelsX	C4
TotalDisplayPixelsY	C5
UriSchemas	C6
USB	C7

Attribute Value	Token [0x]
Vendor	C8
Version	C9

Table 9-6 Attribute Value Tokens