

HF-Services V2.0.0 (2017-12-18)

ULE Alliance Standard

**Digital Enhanced Cordless Telecommunications (DECT);
Ultra Low Energy (ULE);**

**Home Area Network Functional (HAN-FUN) Core Services and
Interfaces**



Keywords

DECT, ULE, HAN, HAN-FUN

ULE ALLIANCE
Secretariat

Wabernstrasse 40
3007 Berne
Switzerland

T: +49 89 5166 2456 M: +49 160 9667 96966
Email: secretariat@ulealliance.org
<http://www.ulealliance.org>

Important notice

Individual copies of the present document can be downloaded from:
<http://www.ulealliance.org>

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF).
In case of dispute, the reference shall be the printing on ULE Alliance printers of the PDF version kept on a specific network drive within ULE Alliance Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status without further notice. Information on the current status of this and other ULE Alliance documents is available at
<http://www.ulealliance.org>

If you find errors in the present document, please send your comment to one of the following services:
secretariat@ulealliance.org

*Copyright Notification / Terms and Conditions of Use **

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© ULE Alliance 2014

All rights reserved.

* See the annex of the document.

Contents

Intellectual Property Rights	8
Foreword.....	8
1 Scope	9
2 References	9
3 Definitions, symbols and abbreviations	9
3.1 Definitions	9
3.2 Symbols	9
3.3 Abbreviations.....	10
4 Introduction	11
4.1 HF-Service Versioning	12
4.2 Recommendations.....	12
5 Core Services in U'0	12
5.1 Core Services in HF Concentrator (D'0:U'0)	13
5.1.1 Mandatory Services.....	13
5.1.1.1 Device Management Service	13
5.1.1.2 Device Information Service	13
5.1.1.3 Attribute Reporting Service	13
5.1.2 Optional Services	13
5.1.2.1 Bind Management Service.....	13
5.1.2.2 Group Management Service	14
5.1.2.3 Identify Service	16
5.1.2.4 Batch Program Management Service	16
5.1.2.5 Event Scheduling Service.....	16
5.1.2.6 Weekly Scheduling Service.....	16
5.1.2.7 Tamper Alert Service.....	16
5.1.2.8 Time Service.....	17
5.1.2.9 Power Service	17
5.1.2.10 RSSI.....	17
5.1.2.11 Keep Alive Service	17
5.1.2.12 SUOTA Service.....	17
5.1.2 Core Services in a HF Device (D'x:U'0).....	18
5.2.1 Mandatory Services.....	18
5.2.1.1 Device Management Service	18
5.2.1.2 Device Information Service	18
5.2.1.3 Attribute Reporting Service	18
5.2.2 Optional Services	18
5.2.2.1 Bind Management Service.....	18
5.2.2.2 Group Management Service	19
5.2.2.3 Identify Device Service	19
5.2.2.4 Batch Program Management Service	19
5.2.2.5 Event Scheduling Service.....	19
5.2.2.6 Weekly Scheduling Service	20
5.2.2.7 Tamper Alert Service.....	20
5.2.2.8 Time Service.....	20
5.2.2.9 Power Service	20
5.2.2.10 RSSI.....	20
5.2.2.11 Keep Alive Service	20
5.2.2.12 SUOTA Service.....	21
6 Core Services Interfaces.....	22
6.1 Device Management Interface	22
6.1.1 Server Interface Data structures	22
6.1.1.1 Example	23
6.1.2 Server Attributes	24
6.1.2.1 Number of Entries	24

6.1.3	Client Attributes	24
6.1.4	Server to Client Commands	24
6.1.5	Client to Server Commands	24
6.1.5.1	Register Device	26
6.1.5.1.1	Discriminator Present.....	27
6.1.5.1.2	Device Unique Identifiers (UIDs).....	27
6.1.5.1.2.1	International Portable User Identity (IPUI).....	27
6.1.5.1.2.2	Institute of Electrical and Electronics Engineers Media Access Control (IEEE MAC-48).....	27
6.1.5.1.2.3	Uniform Resource Identifier (URI)	27
6.1.5.2	De-register Device.....	28
6.1.5.3	Start Session Read Registration Info	29
6.1.5.4	End Session Read Registration Info	30
6.1.5.5	Get Entries	30
6.2	Bind Management Interface.....	32
6.2.1	Server Interface Data structures	32
6.2.2	Server Attributes	33
6.2.2.1	Number of Entries	33
6.2.3	Client Attributes	33
6.2.4	Server to Client Commands	33
6.2.5	Client to Server Commands	34
6.2.5.1	Add Bind	37
6.2.5.2	Remove Bind	39
6.2.5.3	Remove Bind by SRC Device Address	40
6.2.5.4	Remove Bind by SRC Device Address and Unit ID	41
6.2.5.5	Remove Bind by DST Address.....	42
6.2.5.6	Remove Bind by DST Address and Unit ID	43
6.2.5.7	Start Session Read Bind Info.....	44
6.2.5.8	End Session Read Bind Info	45
6.2.5.9	Get Bind Entries by Index	46
6.2.5.10	Get Bind Entries by SRC Device Address.....	47
6.2.5.11	Get Bind Entries by SRC Device Address and Unit ID	48
6.2.5.12	Get Bind Entries by DST Address.....	49
6.2.5.13	Get Bind Entries by DST Address and Unit ID	51
6.3	Group Management Interface	53
6.3.1	Server Interface Data structures	53
6.3.2	Server Attributes	53
6.3.2.1	Number of Groups	53
6.3.3	Client Attributes	53
6.3.4	Server to Client Commands	54
6.3.5	Client to Server Commands	54
6.3.5.1	Create Group	55
6.3.5.2	Delete Group	56
6.3.5.3	Add to Group	57
6.3.5.4	Remove from Group	58
6.3.5.5	Get Group Info	59
6.4	Identify Device Interface	62
6.4.1	Server Attributes	62
6.4.2	Client Attributes	62
6.4.3	Server to Client Commands	62
6.4.3.1	Identification.....	62
6.4.4	Client to Server Commands	63
6.4.4.1	Identify	63
6.5	Device Information Interface	64
6.5.1	Server Attributes	64
6.5.1.1	HF Core Release version	65
6.5.1.2	Profile Release version	65
6.5.1.3	Interface Release version	65
6.5.1.4	Extra Capabilities	65
6.5.1.5	Minimum Sleep Time	66
6.5.1.6	Actual Response Time (Paging Interval).....	66
6.5.1.7	Application version	66
6.5.1.8	Hardware version.....	66

6.5.1.9	EMC (Equipment Manufacturer Code)	66
6.5.1.10	RFPI/IPUI.....	66
6.5.1.11	Manufacturer Name	66
6.5.1.12	Location.....	66
6.5.1.13	Device Enable.....	66
6.5.1.14	Friendly Name	66
6.5.1.15	Device Unique Identifier (UID).....	67
6.5.1.16	Device Serial Number	67
6.6	Attribute Reporting Interface.....	68
6.6.1	Server Interface Data structures	68
6.6.1.1	Periodic Driven Reports	68
6.6.1.2	Event Driven Reports	71
6.6.2	Server Attributes	74
6.6.2.1	Number of Reports	74
6.6.2.2	Number of Periodic Reports	74
6.6.2.3	Number of Event Reports	74
6.6.3	Client Attributes	74
6.6.4	Server to Client Commands	74
6.6.4.1	Periodic Report Notification.....	75
6.6.4.2	Event Report Notification.....	76
6.6.5	Client to Server Commands	79
6.6.5.1	Create Periodic Report.....	80
6.6.5.2	Create Event Report.....	82
6.6.5.3	Add Periodic Report Entry	83
6.6.5.4	Add Event Report Entry	85
6.6.5.5	Delete Report.....	87
6.6.5.6	Get Periodic Report Entries	88
6.6.5.7	Get Event Report Entries	89
6.6.5.8	Update Periodic Report Interval	90
6.7	Batch Program Management Interface	92
6.7.1	Server Interface Data Structures.....	92
6.7.2	Server Attributes	94
6.7.2.1	Maximum Number of Entries	94
6.7.2.2	Number of Entries	94
6.7.3	Client Attributes	94
6.7.4	Server to Client Commands	94
6.7.5	Client to Server Commands	95
6.7.5.1	Define Program	96
6.7.5.2	Invoke Program	97
6.7.5.3	Delete Program.....	98
6.7.5.4	Delete All Programs	99
6.7.5.5	Get Program Actions	100
6.8	Event Scheduling Interface	101
6.8.1	Server Interface Data Structures.....	101
6.8.2	Server Attributes	102
6.8.2.1	Maximum Number of Entries	102
6.8.2.2	Number of Entries	102
6.8.2.3	Scheduler Status	102
6.8.3	Client Attributes	102
6.8.4	Server to Client Commands	102
6.8.5	Client to Server Commands	103
6.8.5.1	Activate Scheduler.....	104
6.8.5.2	Define Event	105
6.8.5.3	Update Event Status.....	107
6.8.5.4	Get Event Entry	108
6.8.5.5	Delete Event	109
6.8.5.6	Delete All Events.....	110
6.9	Weekly Scheduling Interface	112
6.9.1	Server Interface Data Structures.....	112
6.9.2	Server Attributes	113
6.9.2.1	Maximum Number of Entries	113
6.9.2.2	Number of Entries	113

6.9.2.3	Weekly Scheduler Status	113
6.9.3	Client Attributes	113
6.9.4	Server to Client Commands	113
6.9.5	Client to Server Commands	113
6.9.5.1	Activate Weekly Scheduler	115
6.9.5.2	Define Weekly Event.....	116
6.9.5.3	Update Weekly Event Status	117
6.9.5.4	Get Weekly Event Entry.....	118
6.9.5.5	Delete Weekly Event	119
6.9.5.6	Delete All Weekly Events	120
6.10	Group Table Interface	122
6.10.1	Server Attributes	122
6.10.2	Number of entries.....	122
6.10.3	Maximum Number of Entries.....	122
6.10.4	Client Attributes	122
6.10.5	Server to Client Commands	122
6.10.6	Client to Server Commands	123
6.10.6.1	Add Entry	124
6.10.6.2	Delete Entry.....	125
6.10.6.3	Delete All Entries	126
6.10.6.4	Read Entries.....	127
6.11	Tamper Interface.....	129
6.11.1	Server Attributes	129
6.11.1.1	State	129
6.11.2	Client Attributes	129
6.11.3	Server to Client Commands	129
6.11.3.1	Tamper Alert On.....	130
6.11.3.2	Tamper Alert Off.....	130
6.11.4	Client to Server Commands	130
6.12	Time Interface.....	131
6.12.1	Server Attributes	131
6.12.1.1	Time.....	131
6.12.2	Client Attributes	131
6.12.3	Server to Client Commands	131
6.12.4	Client to Server Commands	131
6.13	Power Interface	132
6.13.1	Server Attributes	132
6.13.1.1	Mains Voltage	133
6.13.1.2	Mains Frequency	133
6.13.1.3	Mains Alarms	133
6.13.1.4	Mains Minimum Voltage Threshold	133
6.13.1.5	Mains Maximum Voltage Threshold.....	133
6.13.1.6	Mains Voltage Dwell Trip Point.....	133
6.13.1.7	Battery Voltage.....	133
6.13.1.8	Battery Manufacturer.....	133
6.13.1.9	Battery Type	134
6.13.1.10	Battery AHr Rating.....	134
6.13.1.11	Battery Cell Quantity.....	134
6.13.1.12	Battery Rated Voltage	134
6.13.1.13	Battery Status.....	134
6.13.1.14	Battery Minimum Voltage Threshold.....	134
6.13.1.15	Battery Maximum Voltage Threshold	134
6.13.1.16	Battery Charge Level.....	134
6.13.1.17	Battery Minimum Charge Level.....	134
6.13.1.18	Battery Maximum Charge Level	134
6.13.1.19	Power Source	135
6.13.2	Client Attributes	135
6.13.3	Server to Client Commands	136
6.13.3.1	Low Mains Voltage Alert.....	137
6.13.3.2	High Mains Voltage Alert	137
6.13.3.3	Low Battery Voltage Alert	137
6.13.3.4	High Battery Voltage Alert.....	138

6.13.3.5	Low Battery Charge Alert	138
6.13.3.6	High Battery Charge Alert.....	138
6.13.4	Client to Server Commands	138
6.14	RSSI Interface.....	139
6.14.1	Server Attributes	139
6.14.1.1	RSSI.....	139
6.14.2	Client Attributes	139
6.14.3	Server to Client Commands	139
6.14.4	Client to Server Commands	139
6.15	Keep Alive Interface	140
6.15.1	Server Attributes	140
6.15.1.1	Interval.....	140
6.15.2	Client Attributes	140
6.15.3	Server to Client Commands	140
6.15.3.1	I am Alive	140
6.15.4	Client to Server Commands	140
6.16	SUOTA Interface	141
6.16.1	Server Attributes	141
6.16.2	Client Attributes	141
6.16.3	Server to Client Commands	141
6.16.3.1	New Version Available	141
6.16.4	Client to Server Commands	143
6.16.4.1	Check Version	143
6.16.4.2	Upgrade Complete.....	146

Intellectual Property Rights

The Intellectual Property Rights (IPR) regulation is binding on all members and adopters participating in the ULE Alliance. Its purpose is to ensure the widest possible dissemination of the specifications adopted by the ULE Alliance while giving due weight and respect to the IPR of its members.

The IPR regulation can be found at <http://www.ulealliance.org/downloads.aspx?c=w> (Miscellaneous)

Foreword

This document has been produced by the ULE Alliance TWG.

The information in the present document is believed to be correct at the time of publication. However, Home Area Network Functional (HAN-FUN) may rapidly evolve, and consequently, it is possible that some of the information contained in the present document may become incomplete.

The present document is part of a multi-part deliverable covering the HAN-FUN protocol as identified below:

HF-Overview [REF 1]: Overview

HF-Protocol [REF 2]: Protocol Specification

HF-Service [REF 3]: Core Services & Interfaces

HF-Interface [REF 4]: Interface Library

HF-Profile [REF 5]: Profiles

HF-ULE-Interworking [REF 6]: HF & ULE Interworking

1 Scope

The present document specifies HF core services and their interfaces.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

Referenced documents:

- [1] ETSI EN 300 175-1: "Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI); Part 1: Overview"
http://www.etsi.org/deliver/etsi_en/300100_300199/30017501/02.02.01_60/en_30017501v020201p.pdf
-

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

- *Italic* is used to indicate the name designation of attributes and commands.
- Device addresses are indicated as D'xxxx, where xxxx is a hexadecimal number up to four digits. This provides a compact notation for a HF device address. Usually appears as D'0 referring to the network main device, which has the address 0x0000.
- Group addresses are indicated as G'xxxx and have the same notation definitions as a device address.
- Unit IDs are indicated as U'xx, where xx is a hexadecimal number up to two digits. This provides a compact notation for a unit's ID. Usually appears as U'0 referring to the network management unit, with ID 0x00, that every HF device implements.
- Fully qualified HF network addresses are indicated as D'xxxx:U'xx. This compact notation is a combination of the previous two definitions.
- SRC, compact form to denote Source. Usually precedes a field name, indicating that field belongs to the sender of some information.
- DST, compact form to denote Destination. Usually precedes a field name, indicating that field belongs to the receiver of some information.

3.2 Symbols

For the purposes of the present document, the following symbols apply:

M	Provision Mandatory
O	Provision Optional
N/A	Non-applicable

The symbols here defined are applied to interfaces, attributes and commands and their fields in the present document if not explicitly otherwise stated. The interpretation of these status indications is as follows:

- Provision mandatory, means that the indicated interface, attributes, command or command field shall be implemented as described in the present document, and may be subject to testing.
- Provision optional, means that the indicated interface, attribute, command or command field may be implemented, and if implemented, the interface, attribute or command shall be implemented as described in the present document, and may be subject to testing.
- Non-applicable, means that the indicated interface, attribute, command or command field is not applicable in the defined context. As such, is should not be implemented and consequently not subject to testing.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

CMD	Command
COV	Change of Value
DECT	Digital Enhanced Cordless Telecommunications
DFS	DECT Forum Standard
EMC	Equipment Manufacturer Code
GMEP	Generic Media Encapsulation Protocol
GUI	Graphical User Interface
HAN	Home Area Network
HAN-FUN (HF)	Home Area Network Functional
HF-IFL	HAN-FUN-Interface Library
HF-PRF	HAN-FUN-Profiles
HT	High Threshold
ID	Identifier
IPIUI	International Portable User Identifier
LED	Light Emitting Diode
LT	Low Threshold
LSB	Least Significant Byte
MSB	Most Significant Byte
RFPI	Radio Fixed Part Identity
RSSI	Received Signal Strength Indication
SUOTA	Software Update Over The Air
UID	Unique Identifier
ULE	Ultra Low Energy
UTC	Coordinated Universal Time
WG	Working Group
XML	Extensible Markup Language

4 Introduction

The HAN-FUN Core Services and Interfaces (HF-CSI) document defines a set of services that are either fundamental for the correct operation of a HF network or provide advanced network features that may be useful on certain applications. Services from the latter category are optional to implement while fundamental services are mandatory.

Each service has an associated interface. An interface is a collection of commands and attributes usable in units either mandatory or optionally.

For more general information regarding interfaces, consult the HF-Interface document [REF 4].

Table 1 - List of the available HF service Interfaces.

Service Interfaces		
UID	Name	Description
0x0001	Device Management	Service responsible for device registration and discovery.
0x0002	Bind Management	Service that allows for the creation of logical communication links between devices.
0x0003	Group Management	Service that allows for the creation of logical groups for message broadcasts.
0x0004	Identify	Service that provides a simple method of identifying devices without the hassle of looking and matching serial numbers.
0x0005	Device Information	Service that defines information that any HF device can/must provide.
0x0006	Attribute Reporting	Service that allows a unit to receive automatic notifications about other units or device's attributes whenever an event triggers. This is the preferred method for any device to report information.
0x0007	Batch Program Management	Service that allows the creation of batch programs. Batch programs are an aggregation of several actions that can be triggered into execution at any time and in a repeatable manner.
0x0008	Event Scheduling	Service that allows the creation of time based triggers that will invoke batch programs.
0x0009	Weekly Scheduling	Service that allows for the creation of weekly based triggers that will invoke batch programs.
0x000A	Group Table	Service that allows for a device to route group messages to the correct unit.
0x0101	Tamper Alert	Service that allows a device to indicate it is being tampered with.
0x0102	Time	Service that allows a device to maintain time referenced to UTC.
0x0110	Power	Service that allows a device to provide information about the power supplies it has and their characteristics.
0x0111	RSSI	Use when device wants to indicate the RSSI
0x0115	Keep Alive	Service that allows a device to signal it is alive.
0x0400	SUOTA	Interface for SW upgrade over the air

4.1 HF-Service Versioning

The major number of a HF Core Release specifies a certain release of the present document. Interoperability between device profiles is ensured when based on that same release.

4.2 Recommendations

If an optional feature is implemented it must be implemented according the standard and not in a proprietary way.

5 Core Services in U'0

The majority of core services are to be implemented in unit with ID 0x00 (U'0) of a device, either as server or client role. However, each service has particularities that depend if it is implemented on the HF Concentrator device (D'0) or any other HF device (D'x). Such particularities have a huge impact on the service behaviour and mode of operation and are explained in the following sub-sections.

Table 2 summarizes the implementation requirement, mandatory (M) or optional (O), of each service per device type.

Table 2 - Service implementation requirement per device type.

Service Interfaces			
UID	Name	Implementation	
		HF Concentrator (device D'0)	HF device (D'x)
0x0001	Device Management	M	M
0x0002	Bind Management	O	O
0x0003	Group Management	O	O
0x0004	Identify	O	O
0x0005	Device Information	M	M
0x0006	Attribute Reporting	M	M
0x0007	Batch Program Management	O	O
0x0008	Event Scheduling	O	O
0x0009	Weekly Scheduling	O	O
0x000A	Group Table	O	O
0x0101	Tamper Alert	O	O
0x0102	Time	O	O
0x0110	Power	O	O
0x0111	RSSI	O	O
0x0115	Keep Alive	O	O
0x0400	SUOTA	O	O

5.1 Core Services in HF Concentrator (D'0:U'0)

5.1.1 Mandatory Services

5.1.1.1 Device Management Service

Any device that intends to be part of a HF network must register itself with D'0:U'0 of that network. In doing so, a valid HF network address will be provided. As part of the registration process the device will inform upon the units it implements and any optional interfaces such units support. An authorization scheme can be employed to only allow certain HF devices to join the network.

Besides registration and de-registration of devices, D'0:U'0 allows for service discovery by providing a method for any HF device to request information about all network registered devices. So, to allow service discovery and the correct operation of other core services (like Bind Service for example) even D'0:U'0 should register any optional interfaces it implements, using the 0x0000 profile UID. If D'0:U'0 has only mandatory interfaces it does not need to register itself.

D'0:U'0 is required to implement this service's corresponding interface (6.1) as the server role and the required data structures to store devices information as described in 6.1.1.

5.1.1.2 Device Information Service

The Device Information service defines a data set that any HF device can/must provide. Mandatory information relates to the versions of the different HF Protocol layers and communication capabilities, while any other data is optional.

D'0:U'0 is required to implement this service's corresponding interface (6.5) as the server role.

5.1.1.3 Attribute Reporting Service

This service allows a HF unit to be notified when the state of some attribute, in another unit (on the same device or in another HF device) changes or goes over, under or equals some threshold. It can also provide a periodical notification of the value of some attribute.

D'0:U'0 is required to implement this service's corresponding interface (6.6) as the server role and its associated data structures. Therefore allowing applications that require monitoring D'0:U'0 attributes, to register attribute reporting rules and consequently receive notifications.

D'0:U'0 can also implement an authorization scheme to control access and modifications to attribute reporting rules.

5.1.2 Optional Services

5.1.2.1 Bind Management Service

Binding is a service that allows for the creation of logical communication links between, provided they have matching interfaces, two individual device-unit pairs or a single individual device-unit pair and a group of device-unit pairs (see 5.1.2.2 for group information). Each logical link effectively associates triplets of device-unit-interface, being the interface the same, but implemented with a different and matching role by each triplet or group of triplets. That is, due to the well-established Client-Server communication scheme defined between HF interfaces, a client implementation should be bound to a server implementation and vice-versa. Any other combination will simply not work or have unknown effects on the involved devices.

D'0:U'0 is responsible for handling bind requests, providing information about existing bind rules when requested and ensuring the transparent (to HF devices) routing translation function that this service offers. However, this transparent translation creation is only possible for interfaces registered with D'0:U'0; any interface present in a device but not registered is not bind able since D'0:U'0 has no knowledge of its existence.

Any device can use the special device address D'7FFF:U'FF as the destination address when sending commands¹, as shown in Figure 1. Device D'0:U'0 upon receiving such command will search for a matching bind entry and if found, correctly route it. When the receiving device requires sending a reply, it can do so by using the senders address present on the received message.

D'0:U'0 must always receive and route messages sent to D'7FFF:U'FF, which implies it must implement the data structures required to store devices bind information as described in 6.2.1. In addition it may implement this service's corresponding interface (6.2) as the server role to allow access and/or modifications to those data structures. Such accesses can be controlled by implementing an authorization scheme.

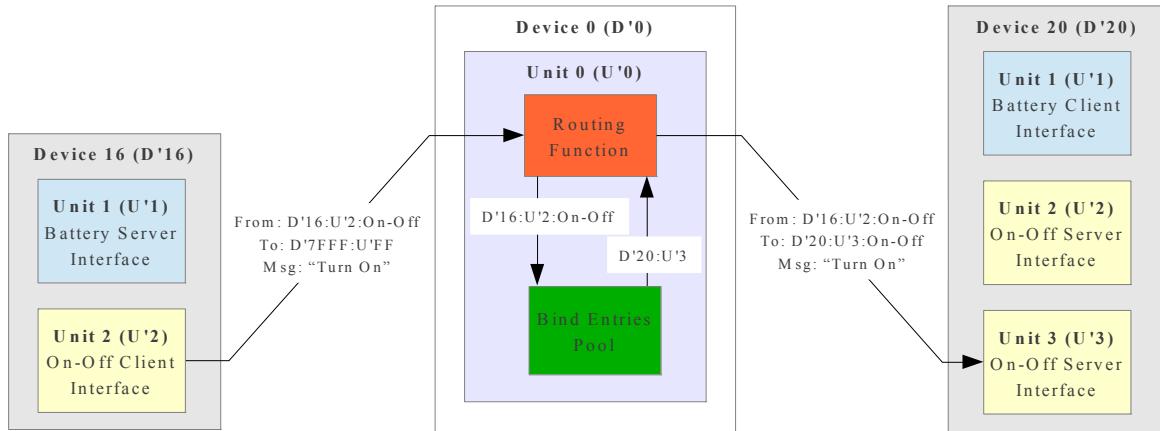


Figure 1 - Example of a communication between 2 bound interfaces.

5.1.2.2 Group Management Service

Group management is a service that allows for the creation of logical addressable unit groups, permitting a message to be delivered to several units using only a single address. Since units live inside HF devices, this service is effectively grouping device-unit pairs. For this service to be available on a HF network D'0:U'0 has to implement this service's corresponding interface (6.3) as the server role and the data structures required to group information has described in 0. It can also implement an authorization scheme to control access and modifications to groups.

D'0:U'0 device has two ways of dispatching group related messages, either it sends unicast messages to each individual unit in each device (Figure 2) or broadcasts the message and then each device would need to internally route it to all units belonging to the group (Figure 3). Both methods have advantages and disadvantages, the former saves memory space on devices at the expense of an increased number of communication exchanges. Receiving devices will also loose the information that the message was send to a group. The broadcast method is exactly the opposite, since at the expense of some memory in each device, it reduces the number of message exchanges while maintaining the information that the message was sent to a group of units.

Since a HF network might grow to a large number of devices, the broadcast method is preferred since it allows for a less congested network, which improves the overall communication quality while lightening D'0:U'0 resource usage. For now, HF only supports the unicast method but broadcast will be available in a future release.

¹ Commands are messages whose Type Code is either 0x01 or 0x02 as defined in [REF 2].

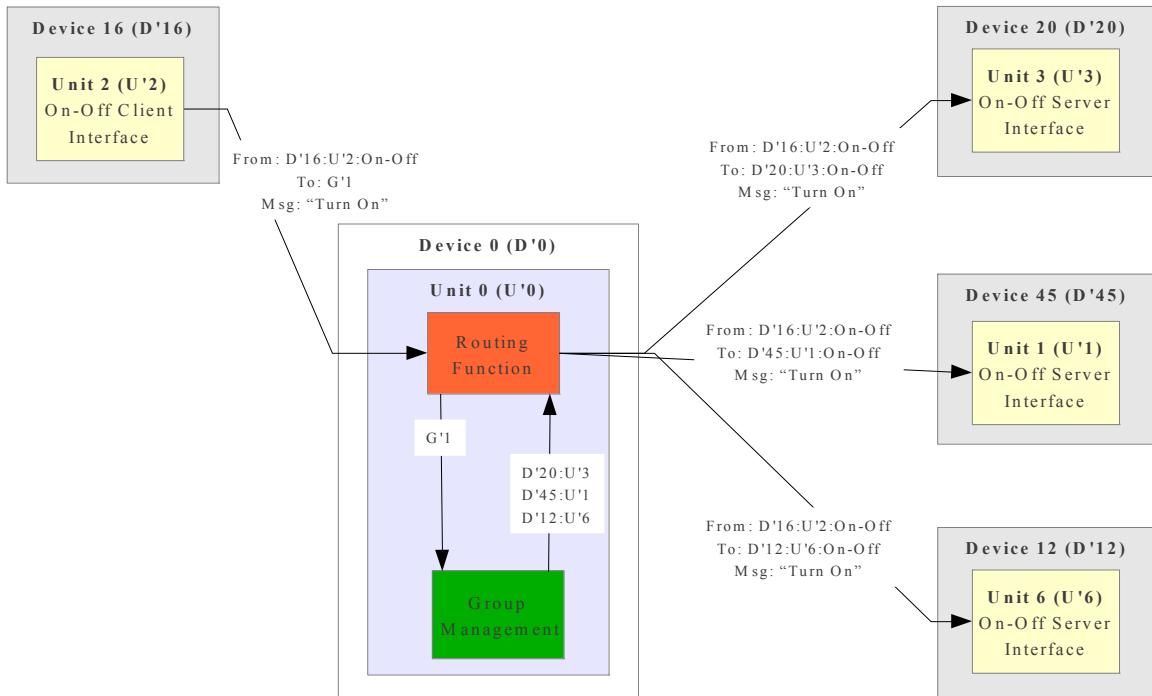


Figure 2 - Group message send using unicast method.

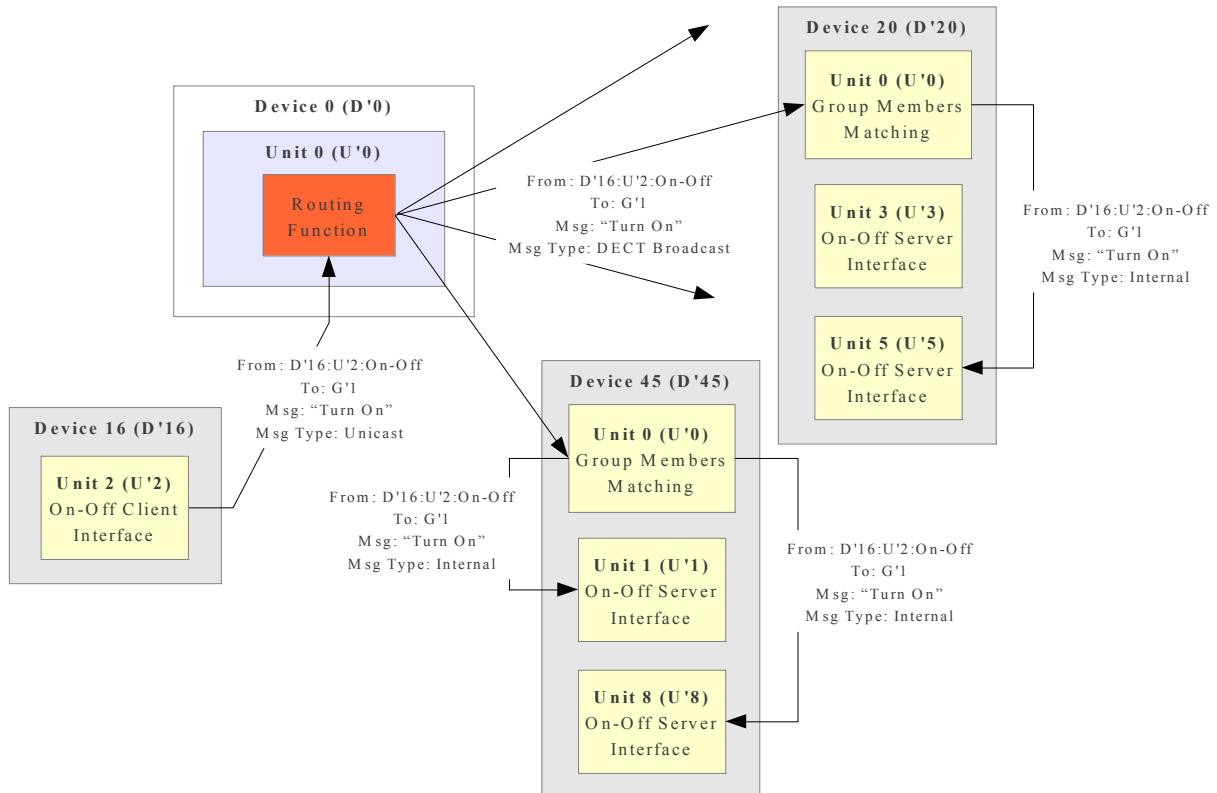


Figure 3 - Group message send using broadcast method.

5.1.2.3 Identify Service

This service provides a simple method of identifying devices without the hassle of looking and matching serial numbers. Solving the specific problem of knowing which device is what, that may arise during an installation of a HF network. The service achieves this by defining simple messages that trigger a device into displaying some physical (for example lighting a led) or virtual (for example sending serial number to a display) action that identifies it.

D'0:U'0 may implement this service's corresponding interface (6.4) as the server role.

5.1.2.4 Batch Program Management Service

This service allows batch programs to be defined. These can be triggered into execution at any time and in a repeatable manner. A batch program is an aggregation of one or more actions. An action is either a *Command*² or a *Set Attribute Request*³, with its respective payload (when applicable) bundled together with both the Unit ID and the Interface ID to which it is destined to.

For example, a batch program can be created with the ON command of an ON-OFF interface on a unit implementing the Simple Light profile. This unit must be on the same device that implements this service.

D'0:U'0 may implement this service's corresponding interface (6.7) as the server role.

5.1.2.5 Event Scheduling Service

This service allows for the creation of time based triggers, specified in seconds, that will invoke batch programs defined using the Batch Program Management service (5.1.2.4). The time reference for these triggers is given by the Time service (5.1.2.8).

A possible use case for this service is in security systems, to turn lights on or off at given times to simulate the presence of people at a house when it is empty (for example during vacations).

D'0:U'0 may implement this service's corresponding interface (6.8) as the server role.

5.1.2.6 Weekly Scheduling Service

This service allows for the creation of time based triggers, specified in a weekly fashion, that will invoke batch programs defined using the Batch Program Management service (5.1.2.4). The time reference for these triggers is given by the Time service (5.1.2.8).

A possible use case for this service is for a central heating system, that rises and lowers a house's temperature, to operate only at scheduled times. This allows for energy savings by turning the system off when a house is empty but also provides the flexibility to turn it on in time for it to pre-heat the house just before people arrive.

D'0:U'0 may implement this service's corresponding interface (6.9) as the server role.

5.1.2.7 Tamper Alert Service

This service allows a device to signal it is being tampered with.

D'0:U'0 may implement this service's corresponding interface (6.10) as the server role.

² Message Type Code 0x01 as defined in [REF 2].

³ Message Type Code 0x06 as defined in [REF 2].

5.1.2.8 Time Service

This service allows a device to maintain a UTC referenced time.

D'0:U'0 may implement this service's corresponding interface (6.12) as the server role.

5.1.2.9 Power Service

This service allows a device to provide information about its power source(s), namely they number, type and characteristics. It is also possible to trigger notifications, by setting high and/or low thresholds on some of the intrinsic characteristics of a power source.

D'0:U'0 may implement this service's corresponding interface (6.13) as the server role.

5.1.2.10 RSSI

This service enables a device to report the RSSI.

The device implementing the RSSI service must be able to create, at least one Periodic and Event report in Attribute Reporting service (see 5.1.1.3 for details), for the RSSI (0x01) attribute.

D'0:U'0 may implement this service's corresponding interface 6.14) as the server role.

5.1.2.11 Keep Alive Service

This interface should be used by any device that requires sending a “keep alive” periodic message. It allows for the periodic interval to be specified with millisecond resolution.

D'0:U'0 may implement this service's corresponding interface (6.15) as the server role.

5.1.2.12 SUOTA Service

This service enables a device to perform Software Update over the Air (SUOTA). Its interface only defines commands that allow for the update procedure to be triggered by either the Server side or the Client side and a command that informs when upgrade completes. The update procedure itself is dependent upon the lower layers so the respective Interworking document must be consulted for details.

D'0:U'0 may implement this service's corresponding interface (6.16) as the server role.

5.2 Core Services in a HF Device (D'x:U'0)

5.2.1 Mandatory Services

5.2.1.1 Device Management Service

Any device that intends to be part of a HF network must register itself to that network. In doing so, the device will inform upon the units it implements and any optional interfaces such units support. In return a valid HF network address will be provided. This registration is achieved by sending the appropriate message to D'0:U'0 which, due to the star topology characteristic of a HF network, is the device that coordinates all network messages and major services.

U'0 from any HF device does not require being registered if it has only mandatory interfaces, otherwise, if it has optional interfaces, U'0 should be registered with profile UID 0x0000 and the record of which optional interfaces it supports. Any other units in the device should be registered to allow for service discovery by other network members and the correct operations of some core services (like Bind Service for example).

A HF device is therefore required to implement, on U'0, this service's corresponding interface (6.1) as the client role.

5.2.1.2 Device Information Service

The Device Information service defines a data set that any HF device can/must provide. Mandatory information relates to the versions of the different HF Protocol layers and communication capabilities, while any other data is optional.

A HF device must implement, on U'0, this service's corresponding interface (6.5) as the server role.

5.2.1.3 Attribute Reporting Service

This service allows a HF unit to be notified when the state of some attribute, in another unit (on the same device or in another HF device) changes or goes over, under or equals some threshold. It can also provide a periodical notification of the value of some attribute.

A HF device is required to implement, on U'0, this service's corresponding interface (6.6) as the server role and its associated data structures. Therefore allowing applications to monitor its attributes by registering attribute reporting rules and consequently receiving notifications.

The device can also implement an authorization scheme to control access and modifications to attribute reporting rules.

5.2.2 Optional Services

5.2.2.1 Bind Management Service

Binding is a service that allows for the creation of logical communication links between, provided they have matching interfaces, two individual device-unit pairs or a single individual device-unit pair and a group of device-unit pairs (see 5.2.2.2 for group information). Each logical link effectively associates triplets of device-unit-interface, being the interface the same, but implemented with a different and matching role by each triplet or group of triplets. That is, due to the well-established Client-Server communication scheme defined between HF interfaces, a client implementation should be bound to a server implementation and vice-versa. Any other combination will simply not work or have unknown effects on the involved devices.

Any HF device that requires managing (adding, removing and retrieving) bind entries will need to implement, on U'0, this service's corresponding interface (6.2) as the client role. The registration of a bind can be done by a HF device that may or may not itself be part of the bind entry. This implies any device implementing bind management client interface, when successfully authorized, can bind any other devices. In turn, those devices do not need to implement this service.

Any HF device that only wants to benefit from this service, provided a bound already exists in D'0:U'0 (see 5.1.2.1 for more information), does not need to implement anything; it can simply send commands⁴ to the special device address D'7FFF:U'FF and D'0:U'0 will correctly route it (see Figure 1). Bound devices save resources since they do not require discovering and storing network information usually required for sending commands to another device.

5.2.2.2 Group Management Service

Group management is a service that allows for the creation of logical addressable unit groups, permitting a message to be delivered to several units using only a single address. Since units live inside HF devices, this service is effectively grouping device-unit pairs.

A HF device only needs to implement this service if it requires managing groups, not if it only requires sending/receiving messages to/from a group. Furthermore, only receiving units need to be part of the group. Sending units do not need to know which group to transmit to, but are not required to be a member of that group.

Managing a group can be done by any device, provided it received authorization to do so. Therefore, an entire HF network can have a single device managing groups, saving resources on other, simpler HF devices, while authorization still maintains a degree of security and controllability over the process.

To implement this service a HF device is required to implement, on U'0, this service's corresponding interface (6.3) as the client role.

5.2.2.3 Identify Device Service

This service provides a simple method of identifying devices without the hassle of looking and matching serial numbers. Solving the specific problem of knowing which device is what, that may arise during an installation of a HF network. The service achieves this by defining simple messages that trigger a device into displaying some physical (for example lighting a led) or virtual (for example sending serial number to a display) action that identifies it.

Any HF device may implement, on U'0, this service's corresponding interface (6.4) as the server role.

5.2.2.4 Batch Program Management Service

This service allows batch programs to be defined. These can be triggered into execution at any time and in a repeatable manner. A batch program is an aggregation of one or more actions. An action is either a *Command*⁵ or a *Set Attribute Request*⁶, with its respective payload (when applicable) bundled together with both the Unit ID and the Interface ID to which it is destined to.

For example, a batch program can be created with the ON command of an ON-OFF interface on a unit implementing the Simple Light profile. This unit must be on the same device that implements this service.

Any HF device may implement, on U'0, this service's corresponding interface (6.7) as the server role.

5.2.2.5 Event Scheduling Service

This service allows for the creation of time based triggers, specified in seconds, that will invoke batch programs defined using the Batch Program Management service (5.1.2.4). The time reference for these triggers is given by the Time service (5.2.2.8).

A possible use case for this service is in security systems, to turn lights on or off at given times to simulate the presence of people at a house when it is empty (for example during vacations).

⁴ Commands are messages whose Type Code is either 0x01 or 0x02 as defined in [REF 2].

⁵ Message Type Code 0x01 as defined in [REF 2].

⁶ Message Type Code 0x06 as defined in [REF 2].

Any HF device may implement, on U'0, this service's corresponding interface (6.8) as the server role.

5.2.2.6 Weekly Scheduling Service

This service allows for the creation of time based triggers, specified in a weekly fashion, that will invoke batch programs defined using the Batch Program Management service (5.1.2.4). The time reference for these triggers is given by the Time service (5.2.2.8).

A possible use case for this service is for a central heating system, that rises and lowers a house's temperature, to operate only at scheduled times. This allows for energy savings by turning the system off when a house is empty but also provides the flexibility to turn it on in time for it to pre-heat the house just before people arrive.

Any HF device may implement, on U'0, this service's corresponding interface (6.9) as the server role.

5.2.2.7 Tamper Alert Service

This interface allows a device to signal it is being tampered with.

Any HF device may implement, on U'0, this service's corresponding interface (6.10) as the server role.

5.2.2.8 Time Service

This interface allows a device to maintain a UTC referenced time.

Any HF device may implement, on U'0, this service's corresponding interface (6.12) as the server role.

5.2.2.9 Power Service

This interface a device to provide information about its power source(s), namely they number, type and characteristics. It is also possible to trigger notifications, by setting high and/or low thresholds on some of the intrinsic characteristics of a power source.

Any HF device may implement, on U'0, this service's corresponding interface (6.13) as the server role.

5.2.2.10 RSSI

This service enables a device to report the RSSI.

The device implementing the RSSI service must be able to create, at least one Periodic and Event report in Attribute Reporting service (see 5.2.1.3 for details), for the RSSI (0x01) attribute.

Any HF device may implement, on U'0, this service's corresponding interface 6.14) as the server role.

5.2.2.11 Keep Alive Service

This service should be used by any device that requires sending a “keep alive” periodic message. It allows for the periodic interval to be specified with millisecond resolution.

Any HF device may implement, on U'0, this service's corresponding interface (6.15) as the server role.

5.2.2.12 SUOTA Service

This service enables a device to perform Software Update over the Air (SUOTA). Its interface only defines commands that allow for the update procedure to be triggered by either the Server side or the Client side and a command that informs when upgrade completes. The update procedure itself is dependent upon the lower layers so the respective Interworking document must be consulted for details.

Any HF device may implement, on U'0, this service's corresponding interface (6.16) as the server role.

6 Core Services Interfaces

6.1 Device Management Interface

This interface defines the commands and attributes to be used by a device implementing Device Management Service.

6.1.1 Server Interface Data structures

A registration record consists of a device entry with the information described in Table 3 and organized (for messaging purposes) as shown in Table 4. In turn, each device entry has one or more unit entries each with the information described in Table 5, organized (for messaging purposes) as shown in Table 6.

The device D'0:U'0 is therefore required to store a device entry for each registered HF device.

Table 3 - Information stored for each registered device (device entry).

Field Name	Field Description	Type	Value	M/O
Device Address	HF device address, provided by the HF Concentrator upon device registration.	U15	0x0000 - 0x7FFE	M
Discriminator Present	Indicates if the device supports proprietary interfaces. If it does, the <i>Discriminator Value</i> field must be present. See 6.1.5.1.1 for more details.	U1	0x00 - Not present 0x01 - Present	M
Discriminator Value	If <i>Discriminator Present</i> is 0x01 then this field must be present and contains the device's EMC.	U16	0x0000 - 0xFFFF	O
Number of units of entry	Number of units, reported by the device, as being implemented.	U8	0x00 - 0xFF	M
Unit Entry	Unit entry, see Table 5 and Table 6.	-	-	M

Table 4 – Byte order and information organization of a device entry.

8	7	6	5	4	3	2	1	Octet
Discriminator Present								1
								2
								3
								4
								5
								6
								⋮
								7 + n

Table 5 - Information stored in each Unit Entry.

Field Name	Field Description	Type	Value	M/O
Size of Unit Entry	Size, in bytes, of unit entry without the <i>Size of Unit Entry</i> byte (that is, from octet 2 onwards, see Table 6).	U8	0x00 - 0xFF	M
Unit ID	Identifier of the unit in a device.	U8	0x00 - 0xFE	M
Profile UID	An UID that identifies the unit's profile. (see HF-Profile document [REF 5] for unit types definition)	U16	0x0000 - 0xFFFF	M
Number of optional Interfaces	Number of optional interfaces that the unit supports.	U8	0x00 - 0x7E	O
Interface Type	Identifier of the type of interface.	U1	0x00 - Client 0x01 - Server	O
Optional Interface UID	UID of the optional interfaces supported.	U15	0x0000 - 0x7FFE	O

Table 6 - Byte order and information organization of a unit entry.

8	7	6	5	4	3	2	1	Octet							
Size of Unit Entry								1							
Unit ID								2							
Profile UID (MSB)								3							
Profile UID (LSB)								4							
Number of optional Interfaces								5							
Interface Type	Optional Interface 1 (MSB)							6							
Optional Interface 1 (LSB)								7							
...								:							
Interface Type	Optional Interface n (MSB)							8 + n							
Optional Interface n (LSB)								9 + n ⁷							

6.1.1.1 Example

For clarification on the data structures defined in 6.1.1, consider the example in Figure 4 which shows the information (and its organization) that D'0:U'0 must store for a device with four units and no discriminator value. The IDs and UIDs shown are fictional, any other information is correct. This example tries to cover all distinct scenarios involving optional interfaces; unit 1 has no optional interfaces, unit 2 has only optional server interfaces, while unit 4 has only optional client interfaces, finally unit 3 has both server and client optional interfaces implemented. Note also that this device has not reported U'0 which means that unit does not support any optional interfaces.

The example focuses on the entries and not on the data structure that will group them, although each type of entries will probably be stacked together in a table, the optimal grouping data structure is out of this document's scope.

⁷ The value **n** represents an accumulation of an arbitrary number of bytes.

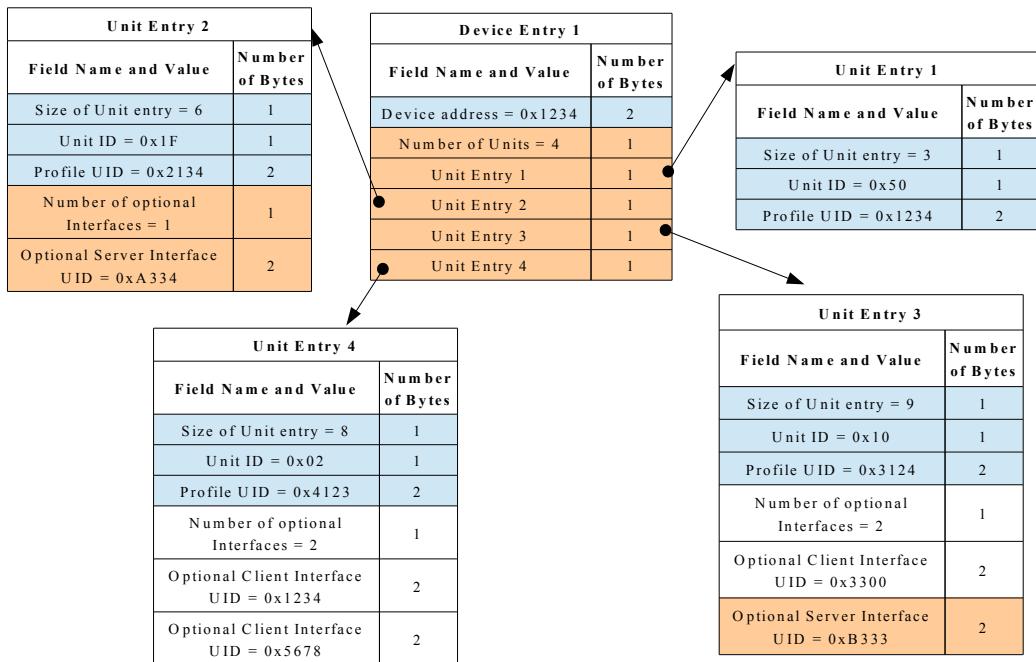


Figure 4 - Example of Device and Unit entries.

6.1.2 Server Attributes

Table 7 - Device Management Interface Server: Attributes.

Attribute ID	Attribute Name	Attribute Type	Attribute Values	Attribute Access	M/O
0x01	Number Of Entries	U16	0x0000 - 0x8000	Read Only	M

6.1.2.1 Number of Entries

Number of Entries attribute records the number of registered devices on any given time.

6.1.3 Client Attributes

None.

6.1.4 Server to Client Commands

None.

6.1.5 Client to Server Commands

The commands provided by this interface can be grouped in two groups: action commands, with the top two commands shown in Figure 5 and information commands, with the remaining three commands also shown in Figure 5. Action commands allow any device to register itself or de-register itself or another from a HF network, while information commands allow any device to discover all other devices.

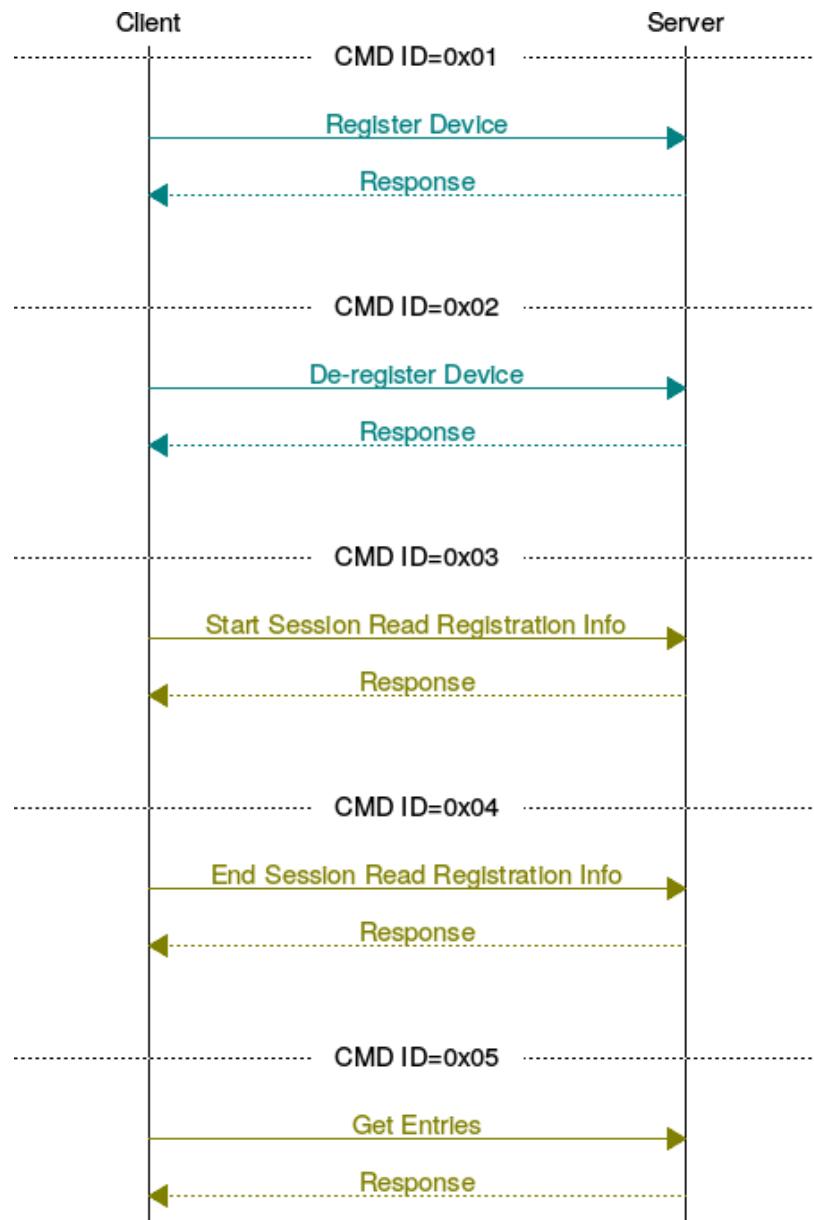


Figure 5 - Device Management Interface: Client Commands.

Table 8 - Implementation status of Device Management Interface Client commands.

Command	Reference	Client Role	Server Role	Response
Register Device	6.1.5.1	M	M	M
De-register Device	6.1.5.2	O	M	M
Start Session Read Registration Info	6.1.5.3	O ⁸	O ⁹	M
End Session Read Registration Info	6.1.5.4			M
Get Entries	6.1.5.5			M

⁸ If one of these commands is required then all three need to be implemented.

⁹ If one of these commands is required then all three need to be implemented.

6.1.5.1 Register Device

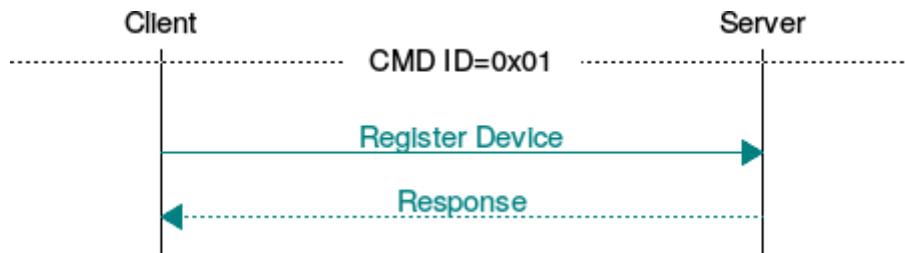


Figure 6 - Device Management: *Register Device* command.

This command, sent with full address D'7FFF:U'0 to D'0:U'0, allows any device to register itself onto a HF network. In doing so, the device will inform upon the units it implements and any optional interfaces such units support. In return a valid HF network address (to be used in all subsequent messages) will be provided in the command's response, if registration is successful.

The command must provide the information described in Table 9, organized according to Table 10.

The response must provide the information described in Table 11, organized according to Table 12. If the Response Code indicates a failure then neither the network address nor the discriminator related fields will be present and the response has only 1 byte. Otherwise the network address will be present and the response grows to 3 bytes total. Additionally if the *Discriminator Present* field is set to “1”, 2 more bytes are sent containing the Concentrator’s *Discriminator Value*, the EMC.

If a device is already registered then the base should update the information for the device and respond with the same device identifier that already belonged to the device.

Table 9 - Data in the payload of a *Register Device* command.

Field Name	Field Description	Type	Value	M/O
Discriminator Present	Indicates if the registering device supports proprietary interfaces. If it does, the <i>Discriminator Value</i> field must be present and have the proprietary EMC. See 6.1.5.1.1 for more details.	U1	0x00 - Not present 0x01 - Present	M
Device UID Type	Indicates the type of unique identifier (UID) the device is using to unequivocally identify itself.	U7	0x00 - None available 0x01 - IPU 0x02 - IEEE MAC-48 0x03 - URI	M
Size of Device UID	Size, in bytes, of the <i>Device UID</i> field. This field depends on <i>Device UID Type</i> , see 6.1.5.1.2 for more details.	U8	0x00 - 0xFF	M
Device UID	Actual value of <i>Device UID</i> . This field depends on <i>Device UID Type</i> , see 6.1.5.1.2 for more details.	-	-	M
Discriminator Value	If <i>Discriminator Present</i> is 0x01 then this field must be present and contains the device’s EMC.	U16	0x0000 - 0xFFFF	O
Number of units	Number of units implemented by the device.	U8	0x00 - 0xFF	M
Unit entry	The information of this field and its organization are the same as in Table 5 and Table 6, respectively, presented in section 6.1.1.	-	-	M

6.1.5.1.1 Discriminator Present

The registering device supports proprietary interfaces if this field is set to “1”, in such case the device must provide its EMC in the *Discriminator Value* field of the *Register Device* command. Otherwise, if the device does not support proprietary interfaces then the *Discriminator Present* field should be “0” and the *Discriminator Value* omitted from the command.

The same process occurs on the response from the HF Concentrator to a *Register Device* command. If the HF Concentrator supports proprietary interfaces then it sets the *Discriminator Present* field to “1” and sends its own EMC in the *Discriminator Value* response field. Otherwise, it sets the *Discriminator Present* field to “0” and omits the *Discriminator Value* field.

6.1.5.1.2 Device Unique Identifiers (UIDs)

6.1.5.1.2.1 International Portable User Identity (IPUI)

This is a 5 byte size, globally unique identifier assigned by the manufacturer to any DECT device.

This UID can be sent on the message or if the size of *Device UID* field is set to zero, then the message should not carry the *Device UID*. The HF layer will ask it directly from a ULE layer.

6.1.5.1.2.2 Institute of Electrical and Electronics Engineers Media Access Control (IEEE MAC-48)

This is a 6 byte size, globally unique identifier assigned by the manufacturer to any network interface involved in communications on the physical network segment.

6.1.5.1.2.3 Uniform Resource Identifier (URI)

This is a variable size string of characters that uniquely identifies a name or a web resource. For example, 1stname.lastname@domain.com, username@sip.domain.com or +97239223322@sip.domain.com.

Table 10 - Data ordering of the payload of a *Register Device* command.

8	7	6	5	4	3	2	1	Octet
Discriminator Present								1
				Device UID Type				2
				Size of Device UID				3
				Device UID (MSB)				...
								4 + n
				Discriminator Value (MSB)				5 + n
				Discriminator Value (LSB)				6 + n
				Number of units				7 + n
				Unit Entry 1				8 + n
			
				Unit Entry n				9 + n ¹⁰

¹⁰ The value **n** represents an accumulation of an arbitrary number of bytes.

Table 11 - Data in the payload of a Response to a *Register Device* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0xFE - Fail: Not enough resources 0xFF - Fail: Unknown reason	M
Discriminator Present	Indicates if the HF Concentrator supports proprietary interfaces. If it does, the <i>Discriminator Value</i> field must be present and have the proprietary EMC. See 6.1.5.1.1 for more details.	U1	0x00 - Not present 0x01 - Present	M
Device Address	The HF device address allocated by D'0:U'0. This field is only present if <i>Response Code</i> is 0x00 - Ok.	U15	0x0001 - 0x7FFE	M
Discriminator Value	If <i>Discriminator Present</i> is 0x01 then this field must be present and contains the device's EMC.	U16	0x0000 - 0xFFFF	O

Table 12 - Data ordering of the payload of a Response to a *Register Device* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
Discriminator Present	Device Address (MSB)							
Device Address (LSB)								3
Discriminator Value (MSB)								4
Discriminator Value (LSB)								5

6.1.5.2 De-register Device

**Figure 7 - Device Management: *De-register Device* command.**

This command, sent to D'0:U'0, allows any authorized device to de-register itself or another device from a HF network. This procedure removes any information, belonging to the de-registered device, stored in D'0:U'0. Such information includes not only the device entry but also any group and binding information. In addition, the HF address used by the device, can be reused in a new device registration.

The command must provide the information described in Table 13, organized according to Table 14.

The response is a single byte – Response Code – with one of the values described in Table 15.

Table 13 - Data in the payload of a *De-register Device* command.

Field Name	Field Description	Type	Value	M/O
Target Device Address	Address of the device to de-register. Any device can either de-register itself or another.	U15	0x0001- 0x7FFE	M

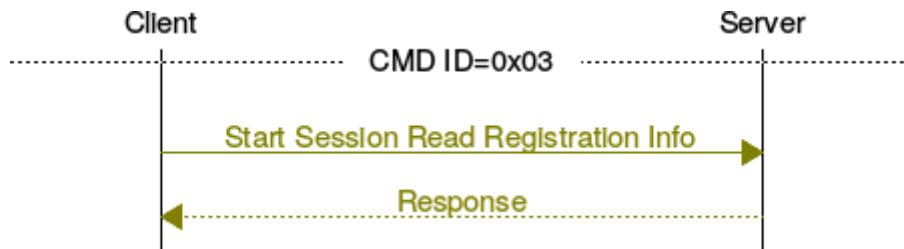
Table 14 - Data ordering of the payload of a *De-register Device* command.

8	7	6	5	4	3	2	1	Octet
x	Target Device Address (MSB)							
	Target Device Address (LSB)							

Table 15 - Data in the payload of a Response to a *De-register Device* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0xFF - Fail: Unknown reason	M

6.1.5.3 Start Session Read Registration Info

**Figure 8 - Device Management: *Start Session Read Registration Info* command.**

This command, sent to D'0:U'0, allows any device to initiate a session for reading the registration information of all devices in a HF network.

Many devices may ask for a read session at the same time causing D'0:U'0 to be temporarily out of resources, in that situation it will reply with the failure “Not enough resources”. The requesting device should wait for some time before trying to request a reading session again.

The command has no payload, but its response must provide the information described in Table 16, organized according to Table 17.

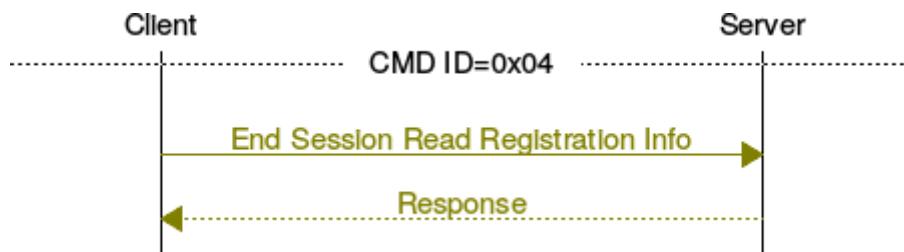
Table 16 - Data in the payload of a Response to a *Start Session Read Registration Info* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0xFE - Fail: Not enough resources 0xFF - Fail: Unknown reason	M
Number of Entries	Number of currently registered devices.	U16	0x0000 - 0x8000	M

Table 17 - Data ordering of the payload of a Response to a *Start Session Read Registration Info* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
Number of Entries (MSB)								2
Number of Entries (LSB)								3

6.1.5.4 End Session Read Registration Info

**Figure 9 - Device Management: *End Session Read Registration Info* command.**

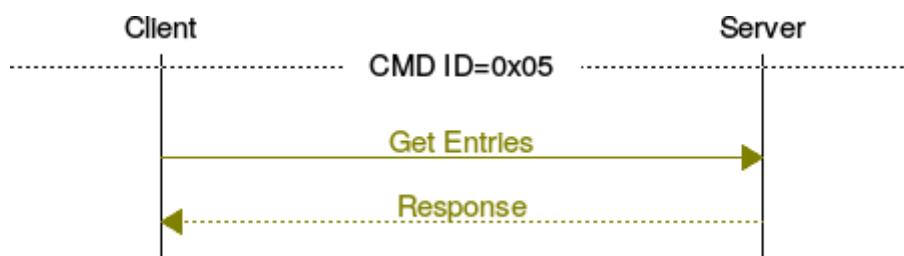
This command, sent to D'0:U'0, terminates a previously requested session for registration info reading, freeing the resources used to maintain such a session.

The command has no payload, but its response has a single byte – Response Code – with one of the values described in Table 18.

Table 18 - Data in the payload of a Response to an *End Session Read Registration Info* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0xFF - Fail: Unknown reason	M

6.1.5.5 Get Entries

**Figure 10 - Device Management: *Get Entries* command.**

This command, sent to D'0:U'0, allows a device to read the registration information of devices in the network when a read session (6.1.5.3) was successfully initialized.

The command must provide the information described in Table 19, organized according to Table 20.

The response must provide the information described in Table 21, organized according to Table 22.

Table 19 - Data in the payload of a *Get Entries* command.

Field Name	Field Description	Type	Value	M/O
Start Entry Index	Indicates the index of the first entry that should be provided.	U16	0x0000 - 0x7FFF	M
Number of Entries	Indicates how many entries should be provided, including the first entry.	U8	0x00 - 0xFF	M

Table 20 - Data ordering of the payload of a *Get Entries* command.

8	7	6	5	4	3	2	1	Octet
Start Entry Index (MSB)								1
Start Entry Index (LSB)								2
Number of Entries								3

Table 21 - Data in the payload of a Response to a *Get Entries* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x02 - Fail: Invalid argument 0x20 - Fail: Read session not established 0x21 - Fail: Entries table was modified 0xFF - Fail: Unknown reason	M
Number of Entries	Total number of entries present on the response payload data.	U8	0x00 - 0xFF	M
Device Entries	Device entries see Table 3 and Table 4.	-	-	M

Table 22 - Data ordering of the payload of a Response to a *Get Entries* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
Number of Entries								2
Device Entry i								3
...								:
Device Entry i+n								4 + n ¹¹

¹¹ The value **n** represents an accumulation of an arbitrary number of bytes.

6.2 Bind Management Interface

This interface defines the commands and attributes to be used by a device implementing Bind Management Service.

Bind rules are only applied to commands. Commands are messages whose Type Code is either 0x01 or 0x02 as defined in [REF 2].

6.2.1 Server Interface Data structures

A bind entry has the information described in Table 23 and organized (for messaging purposes) as shown in Table 24.

Some special values are allowed for fields of a bind entry:

- Setting *Src Device Address* to 0x7FFF, will make the bind entry match any source device address, meaning a message will be routed independently of its source address;
- Setting *Src Unit ID* to 0xFF, will make the bind entry match any source unit ID, meaning a message will be routed independently of its source unit;
- Setting *Interface UID* to 0x7FFF will make the bind entry match any interface, meaning a message will be routed independently of its interface.

The device D'0:U'0 implementing this service is required to store a bind entry for each requested association.

Table 23 - Information stored in each bind entry.

Field Name	Field Description	Type	Value	M/O
SRC Device Address	Address of device that implements Bind Management Client interface and has requested the binding.	U15	0x0000 - 0x7FFF	M
SRC Unit ID	Identifier of the specific unit that has requested the binding.	U8	0x00 - 0xFF	M
DST Interface Type	Type of the interface on destination unit.	U1	0x00 - Client 0x01 - Server	M
Interface UID	Identifier of the interface to bind in both the source and destination device/unit pairs.	U15	0x0000 - 0x7FFF	M
DST Address Type	Type of the address of the destination device.	U1	0x00 - Individual 0x01 – Group	M
DST Device Address	Address of the device that will be bound to the source device.	U15	0x0000 - 0x7FFE	M
DST Unit ID	Identifier of the unit that will be bound to the source unit.	U8	0x00 - 0xFE	M

Table 24 - Byte order and organization of a bind entry.

8	7	6	5	4	3	2	1	Octet
x	SRC Device Address (MSB)							1
	SRC Device Address (LSB)							2
	SRC Unit ID							3
DST Interface Type	Interface UID (MSB)							4
	Interface UID (LSB)							5
DST Address Type	DST Device Address (MSB)							6
	DST Device Address (LSB)							7
	DST Unit ID							8

6.2.2 Server Attributes

Table 25 – Bind Management interface attributes.

Attribute ID	Attribute Name	Attribute Type	Attribute Values	Attribute Access	M/O
0x01	Number Of Entries	U16	0x0000 - 0xFFFF	Read Only	M

6.2.2.1 Number of Entries

Number of Entries attribute records the number of bind entries registered on any given time.

6.2.3 Client Attributes

None.

6.2.4 Server to Client Commands

None.

6.2.5 Client to Server Commands

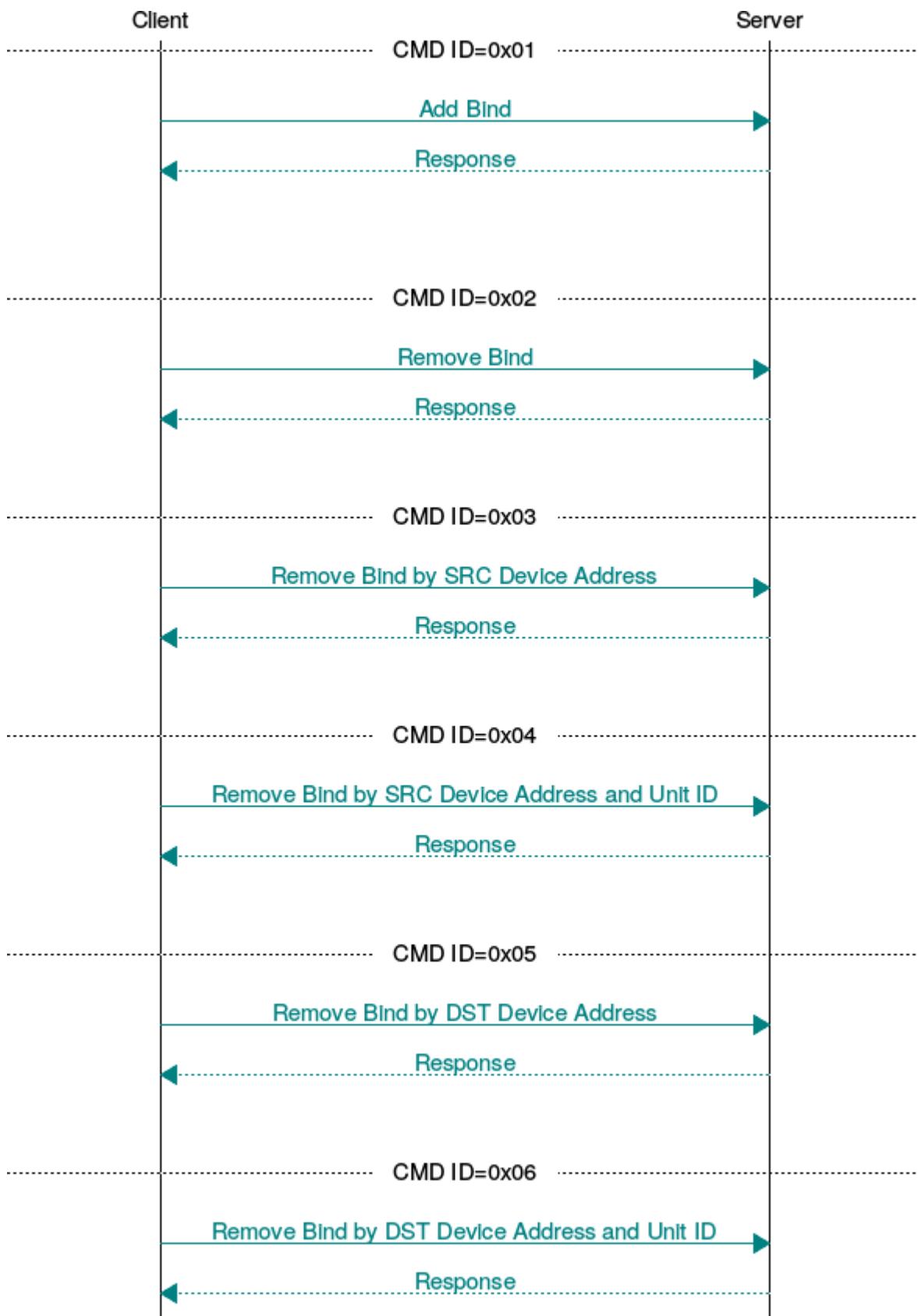


Figure 11 - Bind Management interface: Client Management commands.

Table 26 - Implementation status of Bind Management Interface Client Management commands

Command	Reference	Client Role	Server Role	Response
Add Bind	6.2.5.1	M	M	M
Remove Bind	6.2.5.2	M	M	M
Remove Bind by SRC Device Address	6.2.5.3	O	O	M
Remove Bind by SRC Device Address and Unit ID	6.2.5.4	O	O	M
Remove Bind by DST Address	6.2.5.5	O	O	M
Remove Bind by DST Address and Unit ID	6.2.5.6	O	O	M

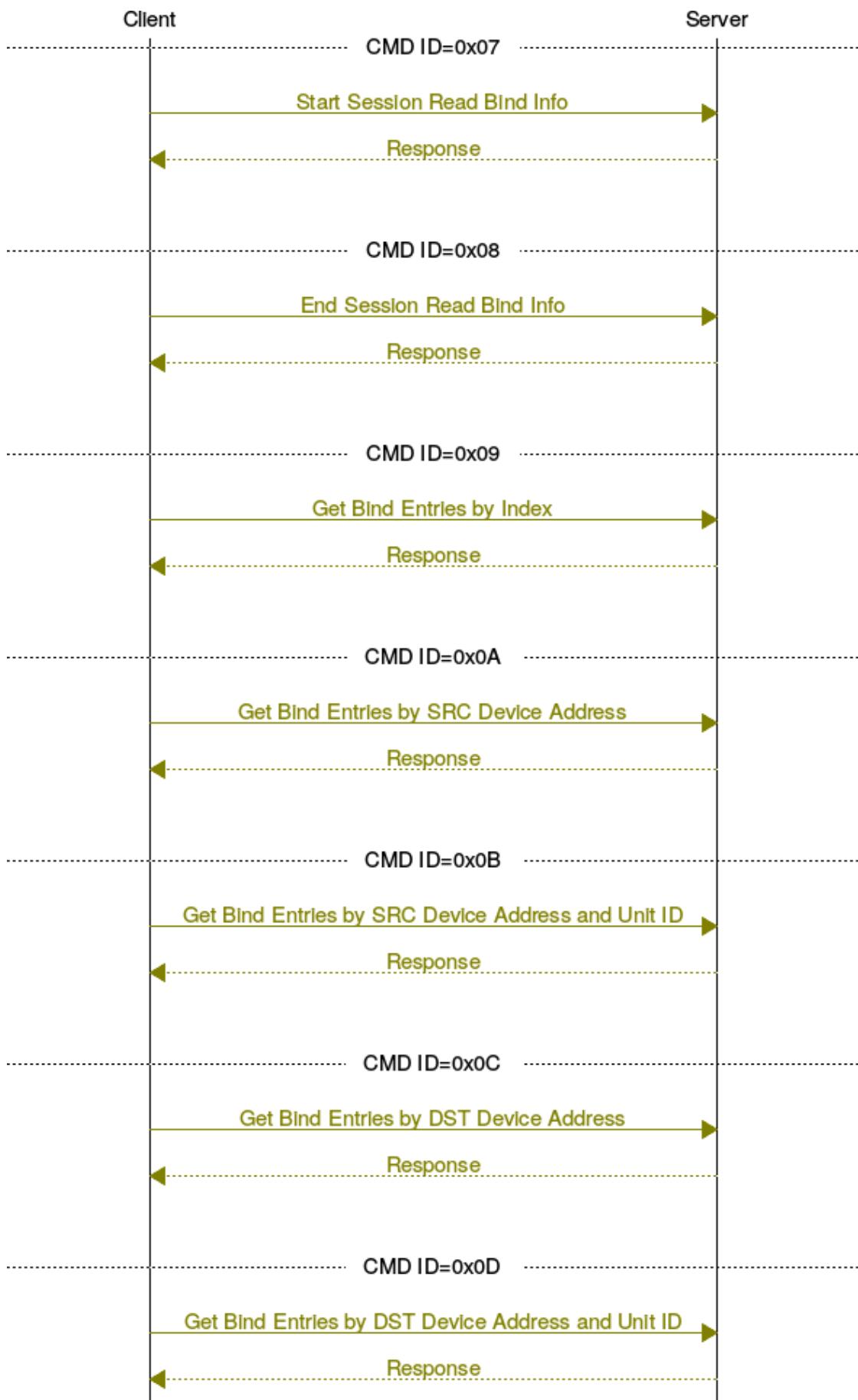
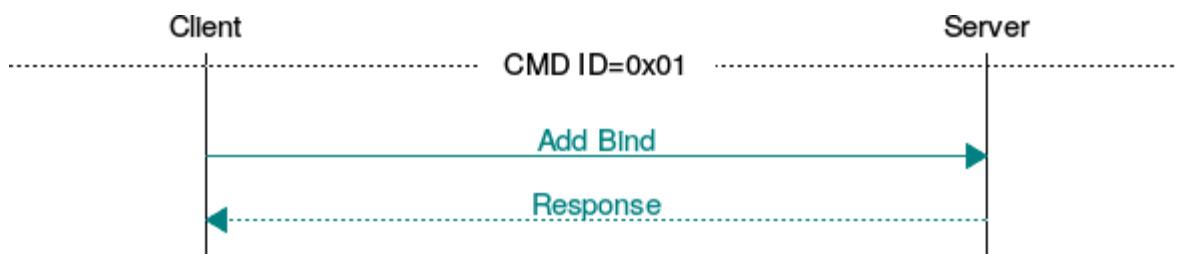


Figure 12 - Bind Management Interface: Client Information commands.

Table 27 - Implementation status of Bind Management Interface Client Information commands

Command	Reference	Client Role	Server Role	Response
Start Session Read Bind Info	6.2.5.7	O ¹²	M	M
End Session Read Bind Info	6.2.5.8			M
Get Bind Entries by Index	6.2.5.9	O	M	M
Get Bind Entries by SRC Device Address	6.2.5.10	O	O	M
Get Bind Entries by SRC Device Address and Unit ID	6.2.5.11	O	O	M
Get Bind Entries by DST Address	6.2.5.12	O	O	M
Get Bind Entries by DST Address and Unit ID	6.2.5.13	O	O	M

6.2.5.1 Add Bind

**Figure 13 - Bind Management: *Add Bind* command.**

This command allows a device (implementing bind client interface) to register a bind entry, effectively associating two triplets of device-unit-interface. The device can bind itself to another, or register a bind entry between two other devices. The bound destination can also be a group address instead of a single device address. For each *Add Bind* request, the D'0:U'0 must create an entry as defined in 0. In addition, it must send a reply indicating the success or failure of the command. In case of failure a reason is given.

After registering a bind entry, whenever the bound unit requires sending a command from its, also bound, interface it must only set the command's destination to the special D'7FFF:U'FF device address. Upon receiving such command, D'0:U'0 will check for a matching bind entry and correctly route it to the intended destination.

The command must provide the information described in Table 28, organized according to Table 29.

The response is a single byte – Response Code – with one of the values described in Table 30.

¹² These two commands need to be implemented if any of the “Get Entries...” is implemented.

Table 28 - Data in the payload of an *Add Bind* command.

Field Name	Field Description	Type	Value	M/O
SRC Device Address	Address of the source device to bind.	U15	0x0000 - 0x7FFF	M
SRC Unit ID	Identifier of the source unit to bind.	U8	0x00 - 0xFF	M
Interface UID	Identifier of the interface to bind in both the source and destination device/unit pairs.	U15	0x0000 - 0x7FFF	M
DST Interface Type	Type of the interface on destination unit.	U1	0x00 - Client 0x01 - Server	M
DST Address	Address of the destination device or group, to bind.	U15	0x0000 - 0x7FFE	M
DST Address Type	Type of <i>DST Address</i> .	U1	0x00 - Individual 0x01 – Group	M
DST Unit ID	Identifier of the destination unit to bind.	U8	0x00 - 0xFE	M

Table 29 - Data ordering of the payload of an *Add Bind* command.

8	7	6	5	4	3	2	1	Octet
x								1
								2
								3
DST Interface Type								4
								5
DST Address Type								6
								7
								8

Table 30 - Data in the payload of a Response to an *Add Bind* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0xFE - Fail: Not enough resources 0xFF - Fail: Unknown reason	M

6.2.5.2 Remove Bind

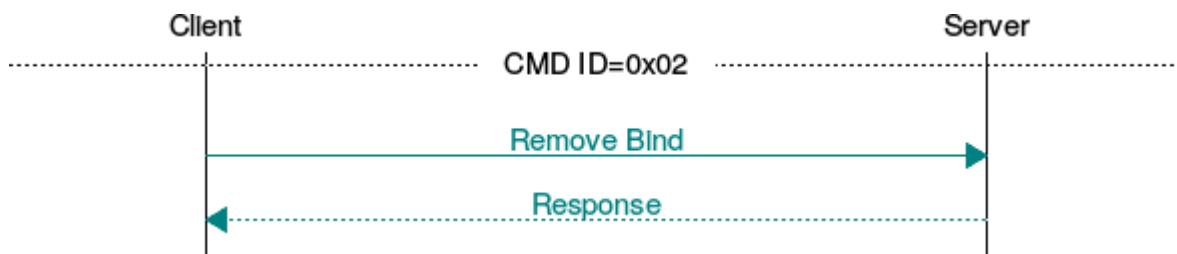


Figure 14 - Bind Management: *Remove Bind* command.

This command allows a device (implementing bind client interface) to remove a bind entry, effectively disassociating two triplets of device-unit-interface. The device can remove a bind between itself and another, or between two other devices (or a group). For each *Remove Bind* request, the D'0:U'0 will remove the specified entry, freeing the resources required to hold it. In addition, it must send a reply indicating the success or failure of the command. In case of failure a reason is given.

After de-registering a bind entry, commands sent by the unbound pair unit/interface to the special D'7FFF:U'FF device address, will silently fail to be delivered to the intended final destination.

The command must provide the information described in Table 31, organized according to Table 32.

The response is a single byte – Response Code – with one of the values described in Table 33.

Table 31 - Data in the payload of a *Remove Bind* command.

Field Name	Field Description	Type	Value	M/O
SRC Device Address	Address of the source device to bind.	U15	0x0000 - 0x7FFF	M
SRC Unit ID	Identifier of the source unit to bind.	U8	0x00 - 0xFF	M
Interface UID	Identifier of the interface to bind in both the source and destination device/unit pairs.	U15	0x0000 - 0x7FFF	M
DST Interface Type	Type of the interface on destination unit.	U1	0x00 - Client 0x01 - Server	M
DST Address	Address of the destination device or group, to bind.	U15	0x0000 - 0x7FFE	M
DST Address Type	Type of <i>DST Address</i> .	U1	0x00 - Individual 0x01 – Group	M
DST Unit ID	Identifier of the destination unit to bind.	U8	0x00 - 0xFE	M

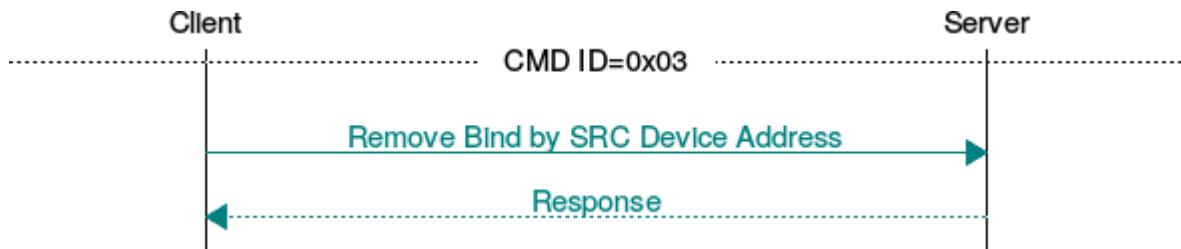
Table 32 - Data ordering of the payload of a *Remove Bind* command.

Octet	8	7	6	5	4	3	2	1
x	SRC Device Address (MSB)							1
	SRC Device Address (LSB)							2
	SRC Unit ID							3
DST Interface Type	Interface UID (MSB)							4
	Interface UID (LSB)							5
DST Address Type	DST Address (MSB)							6
	DST Address (LSB)							7
	DST Unit ID							8

Table 33 - Data in the payload of a Response to a *Remove Bind* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0xFF - Fail: Unknown reason	M

6.2.5.3 Remove Bind by SRC Device Address

**Figure 15 - Bind Management: *Remove Bind by SRC Device Address* command.**

This command allows a device (implementing bind client interface) to remove all bind entries that match the specified source device address. D'0:U'0 must remove all matching entries, freeing the resources required to hold them. In addition, it must send a reply indicating the success or failure of the command. In case of failure a reason is given.

After de-registering bind entries, commands sent from unit-interface pairs, in the unbound device, to the special D'7FFF:U'FF device address, will silently fail to be delivered to the intended final destination.

The command must provide the information described in Table 34, organized according to Table 35.

The response is a single byte – Response Code – with one of the values described in Table 36.

Table 34 - Data in the payload of a Remove Bind by SRC Device Address command.

Field Name	Field Description	Type	Value	M/O
SRC Device Address	Address of the source device for matching with bind entries.	U15	0x0000 - 0x7FFF	M

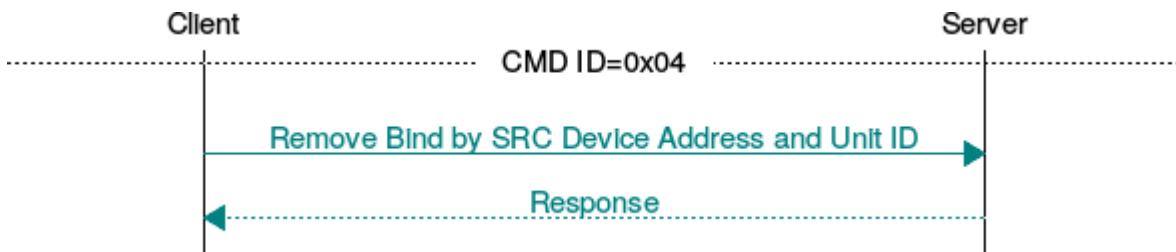
Table 35 - Data ordering of the payload of a Remove Bind by SRC Device Address command.

8	7	6	5	4	3	2	1	Octet
x	SRC Device Address (MSB)							1
SRC Device Address (LSB)							2	

Table 36 - Data in the payload of a Response to a Remove Bind by SRC Device Address command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0x03 - Fail: Not supported 0xFF - Fail: Unknown reason	M

6.2.5.4 Remove Bind by SRC Device Address and Unit ID

**Figure 16 - Bind Management: Remove Bind by SRC Device Address and Unit ID command.**

This command allows a device (implementing bind client interface) to remove all bind entries that match the specified source device address and Unit ID. D'0:U'0 must remove all matching entries, freeing the resources required to hold them. In addition, it must send a reply indicating the success or failure of the command. In case of failure a reason is given.

After de-registering bind entries, commands sent from interfaces of the unbound device-unit pair, to the special D'7FFF:U'FF device address, will silently fail to be delivered to the intended final destination.

The command must provide the information described in **Table 37**, organized according to **Table 38**.

The response is a single byte – Response Code – with one of the values described in Table 39.

Table 37 - Data in the payload of a Remove Bind by SRC Device Address and Unit ID command.

Field Name	Field Description	Type	Value	M/O
SRC Device Address	Address of the source device for matching with bind entries.	U15	0x0000 - 0x7FFF	M
SRC Unit ID	Identifier of the source unit for matching with bind entries.	U8	0x00 - 0xFF	M

Table 38 - Data ordering of the payload of a Remove Bind by SRC Device Address and Unit ID command.

8	7	6	5	4	3	2	1	Octet
x								1
								2
								3

Table 39 - Data in the payload of a Response to a Remove Bind by SRC Device Address and Unit ID command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0x03 - Fail: Not supported 0xFF - Fail: Unknown reason	M

6.2.5.5 Remove Bind by DST Address

**Figure 17 - Bind Management: Remove Bind by DST Address command.**

This command allows a device (implementing bind client interface) to remove all bind entries that match the specified destination address. D'0:U'0 must remove all matching entries, freeing the resources required to hold them. In addition, it must send a reply indicating the success or failure of the command. In case of failure a reason is given.

After de-registering bind entries, commands sent from unit-interface pairs, in the unbound device, to the special D'7FFF:U'FF device address, will silently fail to be delivered to the intended final destination.

The command must provide the information described in Table 40, organized according to Table 41.

The response is a single byte – Response Code – with one of the values described in Table 42.

Table 40 - Data in the payload of a *Remove Bind by DST Address* command.

Field Name	Field Description	Type	Value	M/O
DST Address	Destination address of device or group for matching with bind entries.	U15	0x0000 - 0x7FFF	M
DST Address Type	Type of <i>DST Address</i> .	U1	0x00 - Individual 0x01 – Group	M

Table 41 - Data ordering of the payload of a *Remove Bind by DST Address* command.

8	7	6	5	4	3	2	1	Octet
DST Address Type	DST Address (MSB)							1
	DST Address (LSB)							2

Table 42 - Data in the payload of a Response to a *Remove Bind by DST Address* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0x03 - Fail: Not supported 0xFF - Fail: Unknown reason	M

6.2.5.6 Remove Bind by DST Address and Unit ID

**Figure 18 - Bind Management: *Remove Bind by DST Address and Unit ID* command.**

This command allows a device (implementing bind client interface) to remove all bind entries that match the specified destination address and Unit ID. D'0:U'0 must remove all matching entries, freeing the resources required to hold them. In addition, it must send a reply indicating the success or failure of the command. In case of failure a reason is given.

After de-registering bind entries, commands sent from interfaces of the unbound device-unit pair, to the special D'7FFF:U'FF device address, will silently fail to be delivered to the intended final destination.

The command must provide the information described in Table 43, organized according to Table 44.

The response is a single byte – Response Code – with one of the values described in Table 45.

Table 43 - Data in the payload of a Remove Bind by DST Address and Unit ID command.

Field Name	Field Description	Type	Value	M/O
DST Address	Destination address of device or group for matching with bind entries.	U15	0x0000 - 0x7FFF	M
DST Address Type	Type of DST Address.	U1	0x00 - Individual 0x01 – Group	M
DST Unit ID	Identifier of the destination unit for matching with bind entries.	U8	0x00 - 0xFF	M

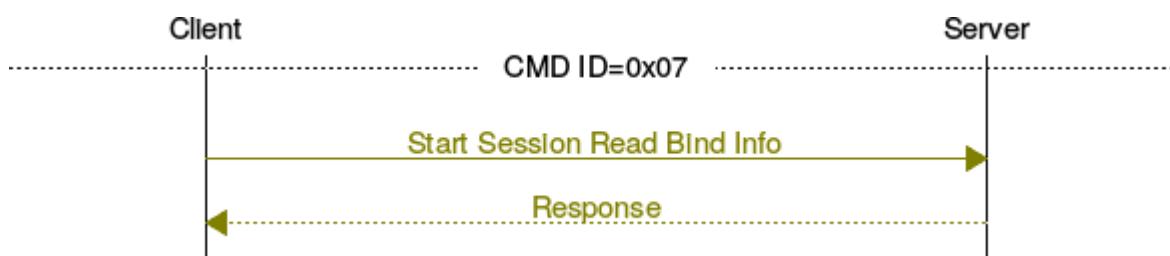
Table 44 - Data ordering of the payload of a Remove Bind by DST Address and Unit ID command.

8	7	6	5	4	3	2	1	Octet
DST Address Type	DST Address (MSB)							1
	DST Address (LSB)							2
	DST Unit ID							3

Table 45 - Data in the payload of a Response to a Remove Bind by DST Address and Unit ID command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0x03 - Fail: Not supported 0xFF - Fail: Unknown reason	M

6.2.5.7 Start Session Read Bind Info

**Figure 19 – Bind Management: Start Session Read Bind Info command.**

This command, allows a device (implementing bind client interface) to initiate a session for reading bind entries from D'0:U'0. Within such session, rules can be requested by absolute index (6.2.5.9), by source device address (6.2.5.10), by source device address and unit ID (6.2.5.11), by destination device address (6.2.5.12) and by destination device address and unit ID (6.2.5.13).

Many devices may ask for a read session at the same time causing D'0:U'0 to be temporarily out of resources, in such situation it will reply with the failure “Not enough resources”. The client device should wait for some time before trying to request a reading session again.

The command has no payload, but its response must provide the information described in Table 46, organized according to Table 47.

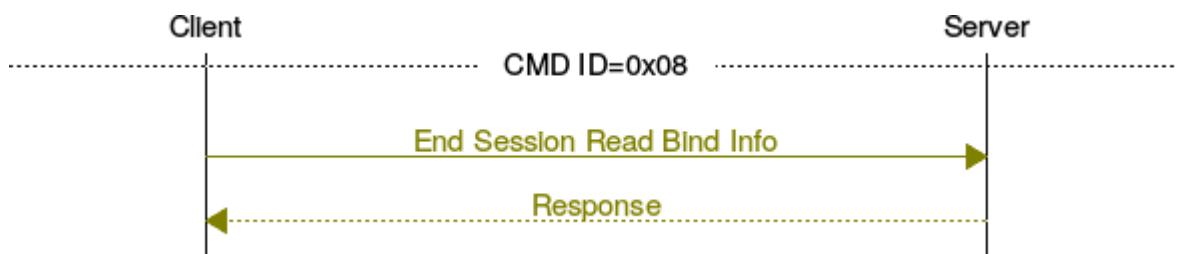
Table 46 - Data in the payload of a Response to a *Start Session Read Bind Info* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0xFE - Fail: Not enough resources 0xFF - Fail: Unknown reason	M
Number of Entries	Number of bind entries currently set on D'0:U'0.	U16	0x0000 - 0xFFFF	M

Table 47 - Data ordering of the payload of a Response to a *Start Session Read Bind Info* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
Number of Entries (MSB)								2
Number of Entries (LSB)								3

6.2.5.8 End Session Read Bind Info

**Figure 20 – Bind Management: *End Session Read Bind Info* command.**

This command, allows a device (implementing bind client interface) to terminate a previously requested session for bind entries reading, freeing the resources used to maintain such a session on D'0:U'0.

The command has no payload, but its response has a single byte – Response Code – with one of the values described in Table 48.

Table 48 - Data in the payload of a Response to an *End Session Read Bind Info* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0xFF - Fail: Unknown reason	M

6.2.5.9 Get Bind Entries by Index



Figure 21 – Bind Management: *Get Bind Entries by Index* command.

This command, allows a device (implementing bind client interface) to retrieve bind entries, present on D'0:U'0, by specifying the index of the first rule entry and the number of entries to read. Before issuing this command, the client device must have a read session (6.2.5.7) successfully initialized.

The command must provide the information described in Table 49, organized according to Table 50.

The response must provide the information described in Table 51, organized according to Table 52.

Table 49 - Data in the payload of a *Get Bind Entries by Index* command.

Field Name	Field Description	Type	Value	M/O
Start Entry Index	Indicates the index of the first entry that should be provided.	U16	0x0000 - 0xFFFF	M
Number of Entries	Indicates how many entries should be provided, including the first entry.	U8	0x00 - 0xFF	M

Table 50 - Data ordering of the payload of a *Get Bind Entries by Index* command.

8	7	6	5	4	3	2	1	Octet
Start Entry Index (MSB)								1
Start Entry Index (LSB)								2
Number of Entries								3

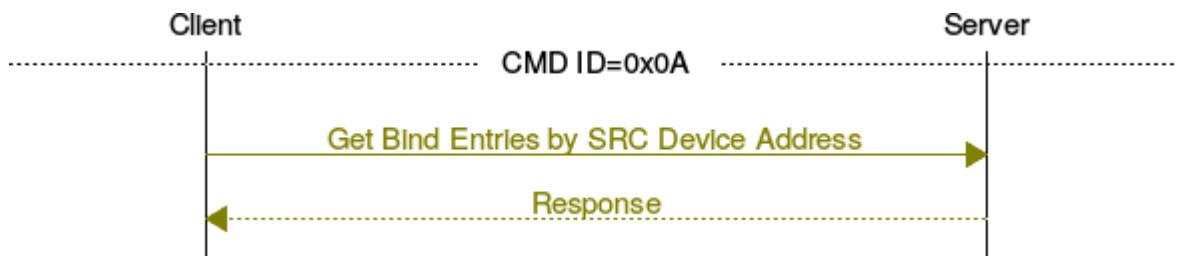
Table 51 - Data in the payload of a Response to a *Get Bind Entries by Index* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x02 - Fail: Invalid argument 0x20 - Fail: Read session not established 0x21 - Fail: Entries table was modified 0xFF - Fail: Unknown reason	M
Number of Entries	Total number of entries present on the response payload data.	U8	0x00 - 0xFF	M
Bind Entries	For bind entries see Table 23 and Table 24.	-	-	M

Table 52 - Data ordering of the payload of a Response to a *Get Bind Entries by Index* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
Number of Entries								2
Bind Entry i								3
...								⋮
Bind Entry i+n								4 + n

6.2.5.10 Get Bind Entries by SRC Device Address

**Figure 22 - Bind Management: *Get Bind Entries by SRC Device Address* command.**

This command, allows a device (implementing bind client interface) to retrieve bind entries, present on D'0:U'0, by specifying a source device address; all bind entries that match the specified source address will be returned. Before issuing this command, the client device must have a read session (6.2.5.7) successfully initialized.

The command must provide the information described in Table 53, organized according to Table 54.

The response must provide the information described in Table 55, organized according to Table 56.

Table 53 - Data in the payload of a *Get Bind Entries by SRC Device Address* command.

Field Name	Field Description	Type	Value	M/O
SRC Device Address	Address of the source device for matching with bind entries.	U15	0x0000 - 0x7FFF	M

Table 54 - Data ordering of the payload of a *Get Bind Entries by SRC Device Address* command.

8	7	6	5	4	3	2	1	Octet
x	SRC Device Address (MSB)							
	SRC Device Address (LSB)							

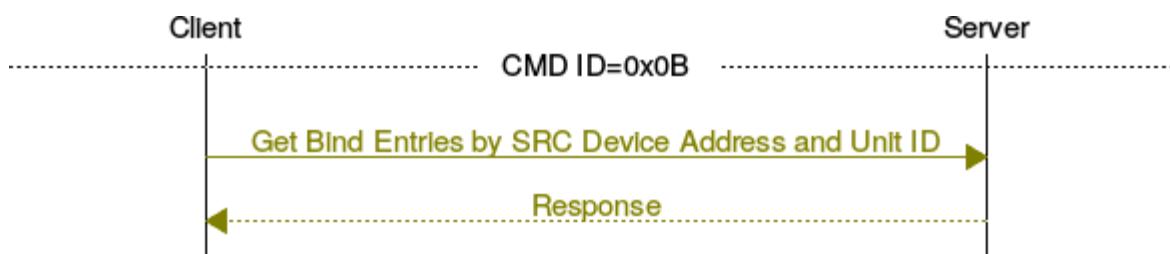
Table 55 - Data in the payload of a Response to a *Get Bind Entries by SRC Device Address* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x02 - Fail: Invalid argument 0x03 - Fail: Not supported 0x20 - Fail: Read session not established 0x21 - Fail: Entries table was modified 0xFF - Fail: Unknown reason	M
Number of Entries	Total number of entries present on the response payload data.	U8	0x00 - 0xFF	M
Bind Entries	For bind entries see Table 23 and Table 24.	-	-	M

Table 56 - Data ordering of the payload of a Response to a *Get Bind Entries by SRC Device Address* command.

8	7	6	5	4	3	2	1	Octet
					Response Code			1
					Number of Entries			2
					Bind Entry i			3
					...			⋮
					Bind Entry i+n			4 + n

6.2.5.11 Get Bind Entries by SRC Device Address and Unit ID

**Figure 23 - Bind Management: *Get Bind Entries by SRC Device Address and Unit ID* command.**

This command, allows a device (implementing bind client interface) to retrieve bind entries, present on D'0:U'0, by specifying a source device address and unit ID; all bind entries that match the specified source address and unit ID will be returned. Before issuing this command, the client device must have a read session (6.2.5.7) successfully initialized.

The command must provide the information described in Table 57, organized according to Table 58.

The response must provide the information described in Table 59, organized according to Table 60.

Table 57 - Data in the payload of a *Get Bind Entries by SRC Device Address and Unit ID* command.

Field Name	Field Description	Type	Value	M/O
Src Device Address	Address of the source device for matching with bind entries.	U15	0x0000 - 0x7FFF	M
Src Unit ID	Identifier of the source unit for matching with bind entries.	U8	0x00 - 0xFF	M

Table 58 - Data ordering of the payload of a *Get Bind Entries by SRC Device Address and Unit ID* command.

8	7	6	5	4	3	2	1	Octet
x	SRC Device Address (MSB)							1
	SRC Device Address (LSB)							2
	SRC Unit ID							3

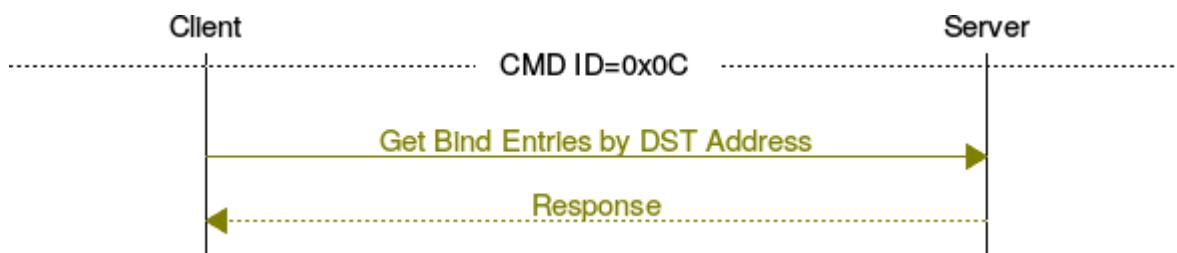
Table 59 - Data in the payload of a Response to a *Get Bind Entries by SRC Device Address and Unit ID* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x02 - Fail: Invalid argument 0x03 - Fail: Not supported 0x20 - Fail: Read session not established 0x21 - Fail: Entries table was modified 0xFF - Fail: Unknown reason	M
Number of Entries	Total number of entries present on the response payload data.	U8	0x00 - 0xFF	M
Bind Entries	For bind entries see Table 23 and Table 24.	-	-	M

Table 60 - Data ordering of the payload of a Response to a *Get Bind Entries by SRC Device Address and Unit ID* command.

8	7	6	5	4	3	2	1	Octet
	Response Code							1
	Number of Entries							2
	Bind Entry i							3
	...							:
	Bind Entry i+n							4 + n

6.2.5.12 Get Bind Entries by DST Address

**Figure 24 - Bind Management: *Get Bind Entries by DST Address* command.**

This command, allows a device (implementing bind client interface) to retrieve bind entries, present on D'0:U'0, by specifying a destination device address; all bind entries that match the specified destination address will be returned. Before issuing this command, the client device must have a read session (6.2.5.7) successfully initialized.

The command must provide the information described in Table 61, organized according to Table 62.

The response must provide the information described in Table 63, organized according to Table 64.

Table 61 - Data in the payload of a *Get Bind Entries by DST Address* command.

Field Name	Field Description	Type	Value	M/O
DST Address	Destination address of device or group for matching with bind entries.	U15	0x0000 - 0x7FFF	M
DST Address Type	Type of <i>DST Address</i> .	U1	0x00 - Individual 0x01 – Group	M

Table 62 - Data ordering of the payload of a *Get Bind Entries by DST Address* command.

8	7	6	5	4	3	2	1	Octet
DST Address Type	DST Address (MSB)							1
	DST Address (LSB)							2

Table 63 - Data in the payload of a Response to a *Get Bind Entries by DST Address* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x02 - Fail: Invalid argument 0x03 - Fail: Not supported 0x20 - Fail: Read session not established 0x21 - Fail: Entries table was modified 0xFF - Fail: Unknown reason	M
Number of Entries	Total number of entries present on the response payload data.	U8	0x00 - 0xFF	M
Bind Entries	For bind entries see Table 23 and Table 24.	-	-	M

Table 64 - Data ordering of the payload of a Response to a *Get Bind Entries by DST Address* command.

8	7	6	5	4	3	2	1	Octet
								Response Code
								1
								Number of Entries
								2
								Bind Entry i
								3
								...
								Bind Entry i+n
								4 + n

6.2.5.13 Get Bind Entries by DST Address and Unit ID

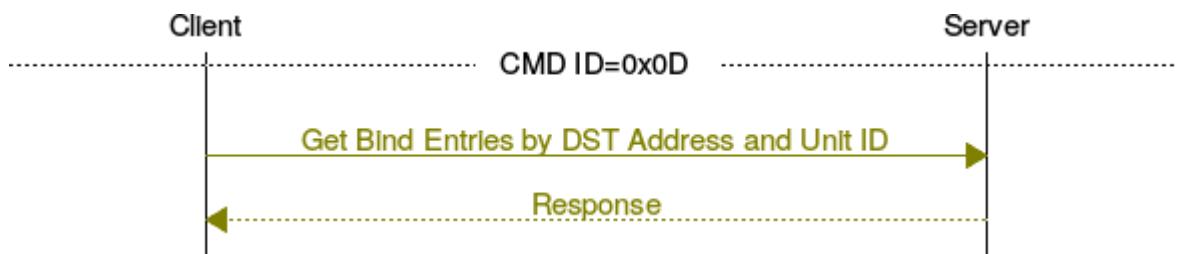


Figure 25 - Bind Management: *Get Bind Entries by DST Address and Unit ID* command.

This command, allows a device (implementing bind client interface) to retrieve bind entries, present on D'0:U'0, by specifying a destination device address and unit ID; all bind entries that match the specified destination address and unit ID will be returned. Before issuing this command, the client device must have a read session (6.2.5.7) successfully initialized.

The command must provide the information described in Table 65, organized according to Table 66.

The response must provide the information described in Table 67, organized according to Table 68.

Table 65 - Data in the payload of a *Get Bind Entries by DST Address and Unit ID* command.

Field Name	Field Description	Type	Value	M/O
DST Address	Destination address of device or group for matching with bind entries.	U15	0x0000 - 0x7FFF	M
DST Address Type	Type of <i>DST Address</i> .	U1	0x00 - Individual 0x01 – Group	M
DST Unit ID	Identifier of the destination unit for matching with bind entries.	U8	0x00 - 0xFF	M

Table 66 - Data ordering of the payload of a *Get Bind Entries by DST Address and Unit ID* command.

8	7	6	5	4	3	2	1	Octet							
DST Address Type	DST Address (MSB)							1							
DST Address (LSB)								2							
DST Unit ID								3							

Table 67 - Data in the payload of a Response to a *Get Bind Entries by DST Address and Unit ID* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x02 - Fail: Invalid argument 0x03 - Fail: Not supported 0x20 - Fail: Read session not established 0x21 - Fail: Entries table was modified 0xFF - Fail: Unknown reason	M
Number of Entries	Total number of entries present on the response payload data.	U8	0x00 - 0xFF	M
Bind Entries	For bind entries see Table 23 and Table 24.	-	-	M

Table 68 - Data ordering of the payload of a Response to a *Get Bind Entries by DST Address and Unit ID* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
Number of Entries								2
Bind Entry i								3
...								⋮
Bind Entry i+n								4 + n

6.3 Group Management Interface

This interface defines the commands and attributes to be used by a device implementing Group Management Service.

6.3.1 Server Interface Data structures

A group entry has the information described in Table 69 and a group member entry the information shown in Table 70.

The device D'0:U'0 that chooses to implement this interface is therefore required to store group entries for each created group and group member entries for each unit belonging to a group.

Table 69 – Information stored in a group entry.

Field Name	Field Description	Type	Value		M/O
Group Address	Address of the group. This serves as the group's unique identifier.	U15	0x0001 - 0x7FFF		M
Group Name	Human readable group name.	Length ¹³	U8	0x00 - 0x20	M
		Character String	U8	0x20 - 0xFE (each U8)	
Number of Members	Number of members, device-unit pairs, present in the group.	U16	0x0000 - 0xFFFF		M

Table 70 – Information stored in a group member entry.

Field Name	Field Description	Type	Value		M/O
Device Address	Address of a device member of the group.	U15	0x0000 - 0x7FFE		M
Unit ID	Identifier of a unit member of the group and residing in the device identified by <i>Device Address</i> .	U8	0x00 - 0xFE		M

6.3.2 Server Attributes

Table 71 – Group Management interface attributes.

Attribute ID	Attribute Name	Attribute Type	Attribute Values	Attribute Access	M/O
0x01	Number Of Groups	U8	0x00 - 0xFF	Read Only	M

6.3.2.1 Number of Groups

Number of Groups attribute records the number of groups that exist on any given time.

6.3.3 Client Attributes

None.

¹³ *Length* is the first byte of the data array; therefore it precedes the character string.

6.3.4 Server to Client Commands

None.

6.3.5 Client to Server Commands

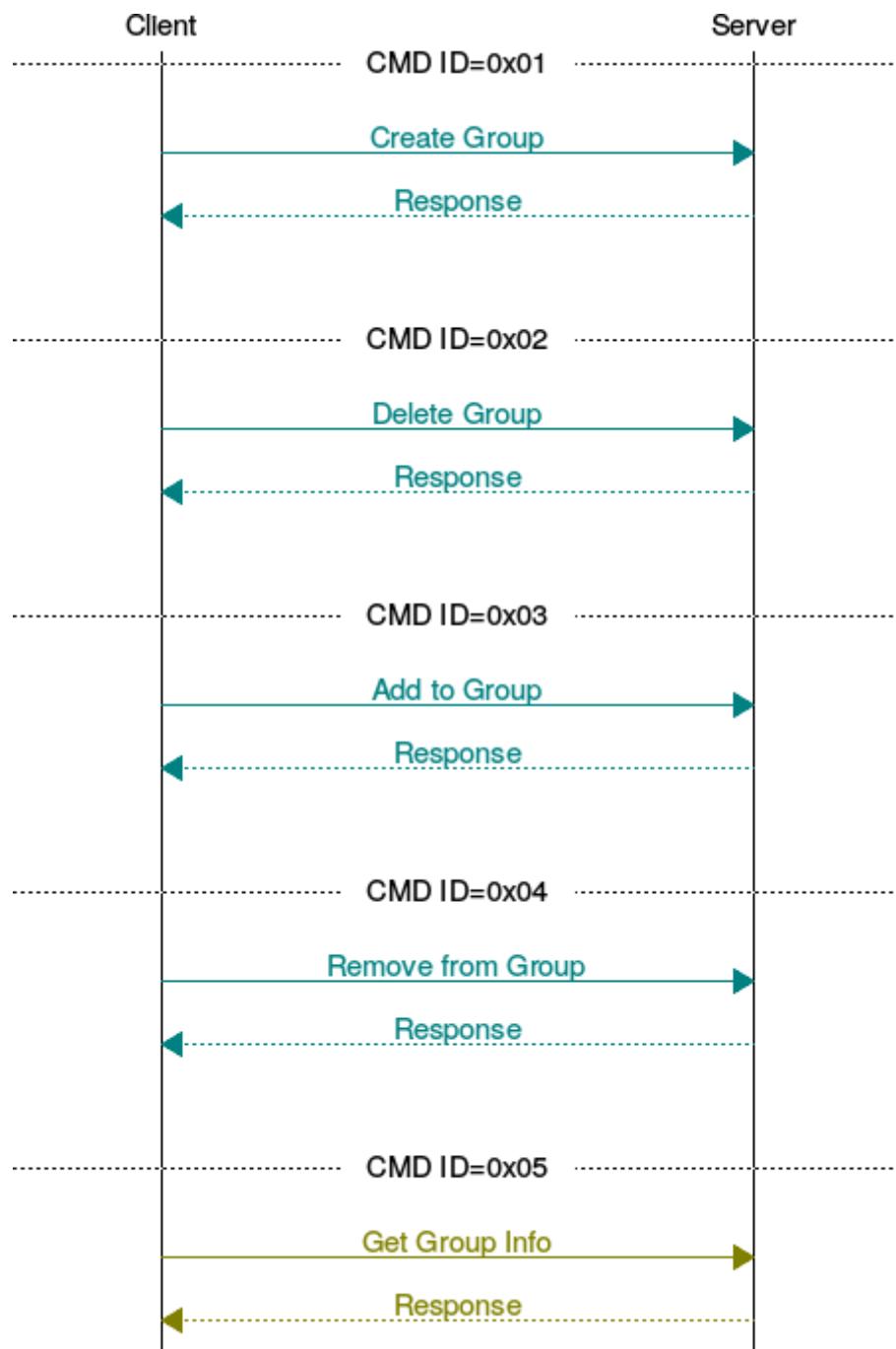
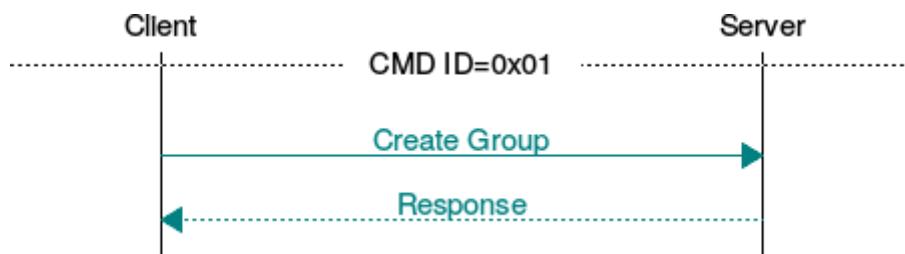


Figure 26 - Group Management interface Client commands.

Table 72 - Implementation status of Group Management Interface Client commands.

Command	Reference	Client Role	Server Role	Response
Create Group	6.3.5.1	M	M	M
Delete Group	6.3.5.2	M	M	M
Add to Group	6.3.5.3	M	M	M
Remove from Group	6.3.5.4	M	M	M
Get Group Info	6.3.5.5	O	O	M

6.3.5.1 Create Group

**Figure 27 - Group Management: *Create Group* command.**

This command, sent to D'0:U'0, allows a device (implementing group management client interface) to create an empty units group. In doing so, the device will provide a human readable name for the group. In return a valid, and unique, HF network group address will be provided in the commands response. Issuing this command will not automatically add the issuer to the created group, it will only inform D'0:U'0 to create the required data structures to maintain group and group membership information. To make a unit part of a group an *Add to Group* command (6.3.5.3) must be issued.

The command must provide the information described in Table 73, organized according to Table 74.

The response must provide the information described in Table 75, organized according to Table 76.

Table 73 - Data in the payload of a *Create Group* command.

Field Name	Field Description	Type	Value		M/O
Group Name	Human readable name intended for the group being created.	Length ¹⁴	U8	0x00 - 0x20	M
		Character String	U8	0x20 - 0xFE (each U8)	

¹⁴ Length is the first byte of the data array; therefore it precedes the character string.

Table 74 - Data ordering of the payload of a *Create Group* command.

8	7	6	5	4	3	2	1	Octet
Group Name Length								1
Group Name Character i								2
...								:
Group Name Character i+n								3 + n

Table 75 - Data in the payload of a Response to a *Create Group* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0xFE - Fail: Not enough resources 0xFF - Fail: Unknown reason	M
Group Address	Address of the created group. This serves as a unique identifier for the group.	U15	0x0001 - 0x7FFF	M

Table 76 - Data ordering of the payload of a Response to a *Create Group* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
x	Group Address (MSB)							
Group Address (LSB)								3

6.3.5.2 Delete Group

**Figure 28 - Group Management: *Delete Group* command.**

This command, sent to D'0:U'0, allows a device (implementing group management client interface) to delete a group by providing its address. This procedure removes all group and group membership information, stored in D'0:U'0. In addition, this group address becomes available for attribution to any new groups, created with *Create Group* command (6.3.5.1).

Any messages sent to a previously deleted group, therefore currently non-existent, will silently fail to be delivered to the intended final destination.

The command must provide the information described in Table 77, organized according to Table 78.

The response is a single byte – Response Code – with one of the values described in Table 79.

Table 77 - Data in the payload of a *Delete Group* command.

Field Name	Field Description	Type	Value	M/O
Group Address	Address of the group to remove/destroy.	U15	0x0001 - 0x7FFF	M

Table 78 - Data ordering of the payload of a *Delete Group* command.

8	7	6	5	4	3	2	1	Octet
x	Group Address (MSB)							1
	Group Address (LSB)							2

Table 79 - Data in the payload of a Response to a *Delete Group* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0xFF - Fail: Unknown reason	M

6.3.5.3 Add to Group

**Figure 29 - Group Management: *Add to Group* command.**

This command, sent to D'0:U'0, allows a device (implementing group management client interface) to add a unit to an existing group. The device can add its own units to a group or units from any other HF device, provided it has been successfully authorized to do so. Units that are part of a group will be able to receive messages sent to that group. Units sending such messages need to know the group address to which transmit to, but are not required to be a member of it.

The special group address 0x7FFF will match all existing groups currently created and present in D'0:U'0. This allows for a unit to be added to all existing groups with a single command.

The command must provide the information described in Table 80, organized according to Table 81.

The response is a single byte – Response Code – with one of the values described in Table 82.

Table 80 - Data in the payload of an *Add to Group* command.

Field Name	Field Description	Type	Value	M/O
Group Address	Address of the already created group, to each the specified unit will be added. To match all existing groups use the special address 0x7FFF.	U15	0x0001 - 0x7FFF	M
Device Address	Address of the device whose unit will be added to the group identified by <i>Group Address</i> .	U15	0x0000 - 0x7FFE	M
Unit ID	Identifier of the unit to add to the group identified by <i>Group Address</i> .	U8	0x00 - 0xFF	M

Table 81 - Data ordering of the payload of an *Add to Group* command.

8	7	6	5	4	3	2	1	Octet
x								1
								2
x								3
								4
								5

Table 82 - Data in the payload of a Response to an *Add to Group* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0xFE - Fail: Not enough resources 0xFF - Fail: Unknown reason	M

6.3.5.4 Remove from Group

**Figure 30 - Group Management: *Remove from Group* command.**

This command, sent to D'0:U'0, allows a device (implementing group management client interface) to remove a unit from a group. The device can remove its own units from a group or units belonging to any other HF device, provided it has been successfully authorized to do so. Units removed from a group will cease to receive messages addressed to that group, but can send messages to it.

The special group address 0x7FFF will match all existing groups currently created and present in D'0:U'0. This allows for a unit to be removed from all existing groups with a single command.

The command must provide the information described in Table 83, organized according to Table 84.

The response is a single byte – Response Code – with one of the values described in Table 85.

Table 83 - Data in the payload of a *Remove from Group* command.

Field Name	Field Description	Type	Value	M/O
Group Address	Address of the group, from which the specified unit will be removed. To match all existing groups use the special address 0x7FFF.	U15	0x0001 - 0x7FFF	M
Device Address	Address of the device whose unit will be removed from the group identified by <i>Group Address</i> .	U15	0x0000 - 0x7FFE	M
Unit ID	Identifier of the unit to remove from the group identified by <i>Group Address</i> .	U8	0x00 - 0xFE	M

Table 84 - Data ordering of the payload of a *Remove from Group* command.

8	7	6	5	4	3	2	1	Octet
x								1
								2
x								3
								4
								5

Table 85 - Data in the payload of a Response to a *Remove from Group* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0xFF - Fail: Unknown reason	M

6.3.5.5 Get Group Info

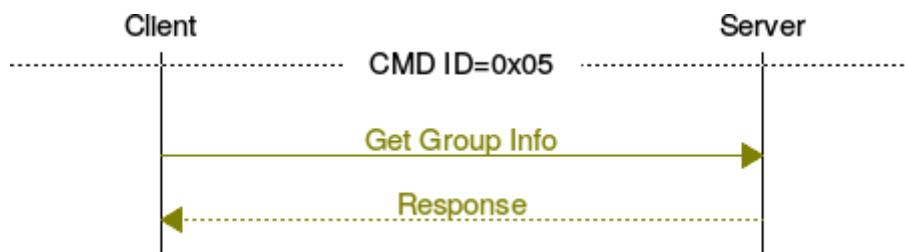


Figure 31 - Group Management: *Get Group Info* command.

This command, sent to D'0:U'0, allows a device (implementing group management client interface) to obtain the human readable name and the list of members of a specified group.

The command must provide the information described in Table 86, organized according to Table 87.

The response must provide the information described in Table 88, organized according to Table 89.

Table 86 - Data in the payload of a *Get Group Info* command.

Field Name	Field Description	Type	Value	M/O
Group Address	Address of the group, from which information is to be retrieved.	U15	0x0001 - 0x7FFF	M

Table 87 - Data ordering of the payload of a *Get Group Info* command.

8	7	6	5	4	3	2	1	Octet
x								Group Address (MSB)
								Group Address (LSB)

Table 88 - Data in the payload of a Response to a *Get Group Info* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0xFE - Fail: Not enough resources 0xFF - Fail: Unknown reason	M
Group Name	Human readable group name.	Length ¹⁵	U8	0x00 - 0x20
		Character String	U8	0x20 - 0xFE (each U8)
Number of Members	Number of members, device-unit pairs, present in the group.	U16	0x0000 - 0xFFFF	M
Device Address	Address of a device member of the group.	U15	0x0000 - 0x7FFE	M
Unit ID	Identifier of a unit member of the group and residing in the device identified by <i>Device Address</i> .	U8	0x00 - 0xFE	M

¹⁵ *Length* is the first byte of the data array; therefore it precedes the character string.

Table 89 - Data ordering of the payload of a Response to a *Create Group* command.

8	7	6	5	4	3	2	1	Octet							
Response Code								1							
Group Name Length								2							
Group Name Character i								3							
...								⋮							
Group Name Character i+n								4 + n							
Number of Members (MSB)								5 + n							
Number of Members (LSB)								6 + n							
x	Device Address (MSB) i							7 + n							
Device Address (LSB) i								8 + n							
Unit ID i								9 + n							
...								⋮							
x	Device Address (MSB) i+n							10 + n							
Device Address (LSB) i+n								11 + n							
Unit ID i+n								12 + n ¹⁶							

¹⁶ The value n represents an accumulation of an arbitrary number of bytes.

6.4 Identify Device Interface

This interface defines the commands and attributes to be used by a device implementing Identify Device Service.

6.4.1 Server Attributes

None.

6.4.2 Client Attributes

None.

6.4.3 Server to Client Commands

Table 90 - Implementation status of Identify Management Interface Server commands.

Command	Reference	Client Role	Server Role	Response
Identification	6.4.3.1	O	O	N/A

6.4.3.1 Identification

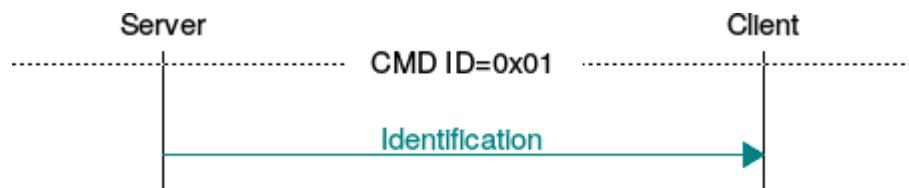


Figure 32 - Identify Device: *Identification* command.

This command is sent to a client interface and triggered by some physical means (like the push of a button) on the sending device. The message must contain some unique identifier, UID number, label, etc. to be displayed on some GUI the client device has.

The command must provide the information described in Table 91, organized according to Table 92.

Table 91 - Data in the payload of *Identification* command.

Field Name	Field Description	Type	Value		M/O
Identifier	Some kind of identifier that easily allows for differentiation between several devices.	Length ¹⁷	U8	0x00 - 0x20	M
		Characters	U8	0x20 - 0xFE (each U8)	

¹⁷ Length is the first byte of the data array; therefore it precedes the character string.

Table 92 - Data ordering of the payload of Identification command.

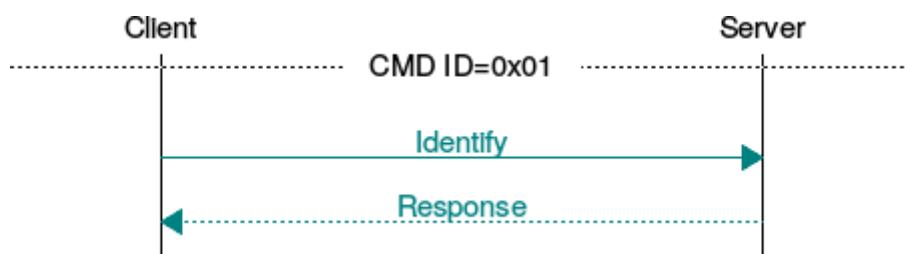
8	7	6	5	4	3	2	1	Octet
Identifier Length								1
Identifier Character i								2
...								:
Identifier Character i+n								3 + n

6.4.4 Client to Server Commands

Table 93 - Implementation status of Identify Management Interface Client commands.

Command	Reference	Client Role	Server Role	Response
Identify	6.4.4.1	O	O	O

6.4.4.1 Identify

**Figure 33 - Identify Device: *Identify* command.**

This command, sent to a server interface, will trigger a physical notification of some sort on the receiving device. Such notification will surely depend on the hardware characteristics of the device. It can be, for example, a led that lights up or blinks, the sound from a buzzer, etc.

With this command, the requesting device will also specify the time interval during which it would like the notification to be active. The receiving device should try to comply with the requested interval but it is not obligated to do so, since power restrictions may impose other limits. Therefore, both devices should be aware that the specified time interval is a reference not a guarantee.

The command must carry a single byte with the information described in Table 94.

The optional response, if implemented, should conform to the General Response Format described in the HF-Protocol document [REF 2].

Table 94 - Data in the payload of an *Identify* command.

Field Name	Field Description	Type	Value	M/O
Time Interval	Time interval during which the physical notification should be active.	U8	0x00 - 0xFF	M

6.5 Device Information Interface

This interface defines the commands and attributes to be used by a device implementing Device Information Service.

Each attribute can be accessed and/or set by any Get/Set Attribute message defined in the HF-Protocol document [REF 2].

6.5.1 Server Attributes

Table 95 - Device Information Interface: Attributes.

Attribute ID	Attribute Name	Attribute Type		Attribute Values	Attribute Access	M/O
0x01	HF Core Release version	U8		0x00 - 0xFF	Read Only	M
0x02	Profile Release version	U8		0x00 - 0xFF	Read Only	M
0x03	Interface Release version	U8		0x00 - 0xFF	Read Only	M
0x04	Extra Capabilities	U8		0x00 - 0xFF	Read Only	M
0x05	Minimum Sleep Time	U32		0x00000000 - 0xFFFFFFFF	Read Only	O
0x06	Actual Response Time (Paging Interval)	U32		0x00000000 - 0xFFFFFFFF	Read Only	O
0x07	Application version	Length ¹⁸	U8	0x00 - 0x20	Read Only	O ¹⁹
		Character String	U8	0x20 - 0xFE (each U8)		
0x08	Hardware version	Length	U8	0x00 - 0x20	Read Only	O ¹⁹
		Character String	U8	0x20 - 0xFE (each U8)		
0x09	EMC	U16		0x0000 - 0xFFFF	Read Only	O
0x0A	RFPI/ IPUI	U8*5		0x00 - 0xFF(each U8)	Read Only	O
0x0B	Manufacturer Name	Length	U8	0x00 - 0x20	Read Only	O ¹⁹
		Character String	U8	0x20 - 0xFE (each U8)		
0x0C	Location	Length	U8	0x00 - 0x20	Read / Write	O
		Character String	U8	0x20 - 0xFE (each U8)		
0x0D	Device Enable	U8		0x00 - 0x01	Read / Write	O
0x0E	Friendly Name	See 6.5.1.14.		-	Read / Write	O

¹⁸ Length is the first byte of the data array; therefore it precedes the character string.

¹⁹ These attributes must be implemented in devices implementing SUOTA (6.16) interface as client role.

Attribute ID	Attribute Name	Attribute Type		Attribute Values	Attribute Access	M/O
0x0F	Device Unique Identifier (UID)	Type	U8	0x00 - 0x03	Read Only	O
		Length	U8	0x00 - 0xFF		
		Character String	U8	0x00 - 0xFF (each U8)		
0x10	Device Serial Number	Length	U8	0x00 - 0xFF	Read Only	O
		Character String	U8	0x20 - 0xFE (each U8)		

6.5.1.1 HF Core Release version

HF Core Release version attribute indicates the version of the HF Core Services and Interfaces release implemented on the device and should be set to the major number of the release version for the corresponding document.

For example, if the device implements services/interfaces according to HF Core Services and Interfaces v2.0.0, then this attribute should be set to 2.

6.5.1.2 Profile Release version

Profile Release version attribute indicates the version of the HF Profile Release implemented on the device and should be set to the major of the HF-Profile document [REF 5] release number.

For example, if the device implements profiles according to HF Profiles v2.0.0, then this attribute should be set to 2.

6.5.1.3 Interface Release version

Interface Release version attribute indicates the version of the HF Interface Release implemented on the device and should be set to the major of the HF-Interface document [REF 4] release number.

For example, if the device implements interfaces according to HF Interfaces v2.0.0, then this attribute should be set to 2.

6.5.1.4 Extra Capabilities

Extra Capabilities attribute informs upon the paging and broadcast capabilities of the device. This information is provided by each individual bit with their order presented in Table 96 and their meaning in Table 97.

Table 96 - Extra Capabilities: Bit ordering.

8	7	6	5	4	3	2	1	Octet
x	x	x	x	x	x	Broadcast	Paging	1

Table 97 - Extra Capabilities: Bit values meaning.

Capability	Bit values meaning
Paging	0 – Device is not page able 1 – Device is page able
Broadcast	0 – Device cannot broadcast 1 – Device can broadcast

6.5.1.5 Minimum Sleep Time

Minimum Sleep Time attribute defines the minimum sleeping period, in milliseconds, a device supports.

In case ULE transport layer is used, this parameter defines the minimum interval between pagings (only for page able devices). Upon registration the device should update this field on D'0:U'0.

6.5.1.6 Actual Response Time (Paging Interval)

Actual response Time (Paging Interval) attribute, in case ULE transport layer is used, defines the actual paging interval, in milliseconds, to be used by a page able device.

This parameter is set by D'0:U'0 which should take into consideration the device's *Minimum Sleep Time*.

6.5.1.7 Application version

Application version attribute is used to indicate the software version of the application running in the device.

6.5.1.8 Hardware version

Hardware version attribute is used to indicate the hardware version of the device.

6.5.1.9 EMC (Equipment Manufacturer Code)

EMC attribute is a container for a manufacturer code allocated by ETSI with an upper limit of 65535. Its usage is manufacturer dependent but the value zero shall not be used.

6.5.1.10 RFPI/IPUI

RFPI/IPUI attribute can be used to indicate the unique identifier associated with any DECT device. When this field is used, the field Device Unique Identifier (UID) (see 6.5.1.15) must contain the same information.

The distinction between RFPI and IPUI is done by checking the device's HF address; in case the device is the HF concentrator (D'0) this field stores a RFPI, otherwise it stores an IPUI.

6.5.1.11 Manufacturer Name

Manufacturer Name attribute is used to store/indicate the device's manufacturer. Useful for Graphical User Interface (GUI) purposes.

6.5.1.12 Location

Location attribute is used to indicate the location of a device in a house. Useful for GUI purposes.

6.5.1.13 Device Enable

Device Enable attribute is used to enable/disable all units in a device, except U'0. U'0 can never be disabled.

6.5.1.14 Friendly Name

Friendly Name attribute is a container for user friendly and meaningful names for a device's units. The name defined for U'0 is considered to be the device's name.

This attribute contains the information defined in Table 98, organized according to Table 99.

Table 98 - Data in the *Friendly Name* attribute.

Field Name	Field Description	Type	Value	M/O
Number of Unit Names	Number of units that currently have friendly names defined.	U8	0x00 - 0xFF	M
Unit ID	Unit identifier to whom <i>Name</i> field refers to.	U8	0x00 - 0xFE	M

Unit Name Length	Length, in byte, of <i>Name</i> field.	U8	0x00 - 0x20	M
Unit Name	Actual unit's name.	-	0x20 - 0xFE (each U8)	M

Table 99 - Data ordering of the data in the *Friendly Name* Attribute.

8	7	6	5	4	3	2	1	Octet
Number of Units Names								1
Unit ID i								2
Unit i Name Length								3
Unit i Name (MSB)								4
...								⋮
Unit i Name (LSB)								5 + n
...								⋮
Unit ID i+n								6 + n
Unit i+n Name Length								7 + n
Unit i+n Name (MSB)								8 + n
...								⋮
Unit i+n Name (LSB)								9 + n

6.5.1.15 Device Unique Identifier (UID)

Device UID attribute is used to uniquely identify a device. It has an associated type according to Table 100.

Table 100 - Device Unique Identifier: Types.

Type	Meaning	Description
0x00	None	Indicates that no unique identifier is assigned to the device.
0x01	IPUI/RFPI	A 5 byte size, globally unique identifier assigned by a manufacturer to any DECT Portable Part or Fixed Part. See also 6.5.1.10.
0x02	IEEE MAC-48	A 6 byte size, globally unique identifier assigned by a manufacturer to any network interface involved in communications on the physical network segment.
0x03	URI	A variable size string of characters that uniquely identifies a name or a web resource. For example, <code>1stname.lastname@domain.com</code> , <code>username@sip.domain.com</code> or <code>+97239223322@sip.domain.com</code> .

6.5.1.16 Device Serial Number

Device Serial Number attribute is used to store/indicate the device's serial number. Its value is manufacturer dependent.

6.6 Attribute Reporting Interface

This interface defines the commands and attributes to be used by a device implementing Attribute Reporting Service. In order to provide flexibility to implementations (especially in low resource devices), all Attribute Reporting commands (even mandatory ones) can have a minimal implementation by simply replying with either *0x03 - Fail: Not supported* or *0xFE - Fail: Not enough resources* response codes to any received command. However, a profile can mandate the implementation of commands; in such case the referred response codes cannot be used and the device needs to correctly parse the command according to its specification.

Each report is identifiable by a unique²⁰ number called *Report ID*, which can take any value in the range 0x00 to 0x7E or the special value 0x7F. This special value signifies “all existing entries” and is only used by *Delete Report* (6.6.5.5). Some of the specified commands use a concept named *Attribute Pack*, which is a means to group attributes together under an UID. This allows for the registration of several attribute reporting rules with a single byte, according to the special values specified in Table 101.

Depending on the issued command, an attribute pack will apply to the attributes of a device, unit or interface. Since a device aggregates units and units contain one or several interfaces, issuing a command to apply to the attributes of a device will implicitly apply to the attributes of all interfaces in all units on that device. Similarly, issuing a command to apply to the attributes of a unit will implicitly apply the attributes of all interfaces of that unit. Finally there are commands that apply to the attributes of a specified interface, in an also specified device-unit pair.

Table 101 - List of Special Attribute Pack UIDs and their meaning.

Attribute Pack		
UID	Name	Description
0x00	Mandatory	Register all mandatory attributes.
0xFE	All	Register all mandatory and optional attributes.
0xFF	Dynamic	Dynamically register attributes. When this attribute pack UID is used a list of attributes must be provided.

6.6.1 Server Interface Data structures

Any device that implements this service as a server role is required to maintain attribute reporting rules, which are divided into periodic and event type rules. There should be one rule for each attribute request.

6.6.1.1 Periodic Driven Reports

A periodic attribute reporting rule consists of an entry with the information described in Table 102 and organized (for messaging purposes) as shown in Table 103. However, for the special *Attribute Pack ID* of 0xFF that allows for dynamic setting of rules on a random set of attributes, the additional information described in Table 106 must be stored and organized (for messaging purposes) according to Table 107.

²⁰ Unique only at the device with server role implementation level that is, a *Report ID* is not unique across different devices implementing this interface’s server role.

Table 102 - Information stored in each Periodic attribute reporting rule.

Field Name	Field Description	Type	Value	M/O
Report Type	Distinguishes if the report is periodic or event driven.	U1	0x00 - Periodic 0x01 - Event	M
Report ID	Report identifier, unique per device with server role implementation.	U7	0x00 - 0x7E	M
Address Type	Indicates the type of <i>Receiver Device Address</i> .	U1	0x00 - Individual 0x01 – Group	M
Receiver Device Address	Address of device that requested the notification.	U15	0x0000 - 0x7FFE	M
Receiver Unit ID	Identifier of the unit that implements Attribute Reporting Client interface and has requested the notification.	U8	0x00 - 0xFE	M
Periodic Interval Value	Specifies the time, in seconds, which dictates the sending of a notification.	U32	0x00000000 - 0xFFFFFFFF	M
Number of Report Entries	Number of currently defined entries in this report.	U8	0x00 - 0xFF	M
Periodic Report Entry	See Table 104 and Table 105.	-	-	M

Table 103 - Byte order and organization of a Periodic attribute reporting rule.

8	7	6	5	4	3	2	1	Octet
Report Type	Report ID							1
Address Type	Receiver Device Address (MSB)							2
	Receiver Device Address (LSB)							3
	Receiver Unit ID							4
	Periodic Interval Value (MSB)							5
	...							⋮
	Periodic Interval Value (LSB)							6 + n
	Number of Report Entries							7 + n
	Periodic Report Entry 1							8 + n
	...							⋮
	Periodic Report Entry n							9 + n

Table 104 - Information stored in each Periodic Report Entry.

Field Name	Field Description	Type	Value	M/O
Unit ID	Identifier of the unit to which the attribute to report, belongs to.	U8	0x00 - 0xFE	M
Interface Type	Type of the interface to which the attribute to report, belongs to.	U1	0x00 - Client 0x01 - Server	M
Interface UID	Identifier of the interface to which the attribute to report, belongs to.	U15	0x0000 - 0x7FFE	M
Attribute Pack UID	See Table 101 in section 6.6 for detailed information.	U8	0x00 - 0xFF	M

Table 105 - Byte order and organization of a Periodic Report Entry, for any *Attribute Pack ID* except 0xFF - Dynamic.

8	7	6	5	4	3	2	1	Octet
Unit ID								1
Interface Type	Interface UID (MSB)							
Interface UID (LSB)								3
Attribute Pack ID								4

Table 106 - Additional data stored in each Periodic Report Entry, for the special *Attribute Pack ID* 0xFF - Dynamic.

Field Name	Field Description	Type	Value	M/O
Number of Attributes	Number of attributes, present in the command, for which reporting rules must be created.	U8	0x00 - 0xFF	M
Attribute ID	Identifier of an attribute to report.	U8	0x00 - 0xFF	M

Table 107 - Byte order and organization of a Periodic Report Entry, for the special *Attribute Pack ID* 0xFF - Dynamic.

8	7	6	5	4	3	2	1	Octet
Unit ID								1
Interface Type	Interface UID (MSB)							
Interface UID (LSB)								3
Attribute Pack ID = 0xFF								4
Number of Attributes								5
Attribute ID								6
...								:
Attribute ID								7 + n

6.6.1.2 Event Driven Reports

An event attribute reporting rule consists of an entry with the information described in Table 108 and organized (for messaging purposes) as shown in Table 109. However, for the special *Attribute Pack ID* of 0xFF that allows for dynamic setting of rules on a random set of attributes, the additional information described in Table 113 must be stored and organized (for messaging purposes) according to Table 114.

Table 108 - Information stored in each Event attribute reporting rule.

Field Name	Field Description	Type	Value	M/O
Report Type	Distinguishes if the report is periodic or event driven.	U1	0x00 - Periodic 0x01 - Event	M
Report ID	Report identifier, unique per device with server role implementation.	U7	0x00 - 0x7E	M
Address Type	Indicates the type of <i>Receiver Device Address</i> .	U1	0x00 - Individual 0x01 – Group	M
Receiver Device Address	Address of device that requested the notification.	U15	0x0000 - 0x7FFE	M
Receiver Unit ID	Identifier of the unit that implements Attribute Reporting Client interface and has requested the notification.	U8	0x00 - 0xFE	M
Number of Report Entries	Number of currently defined entries in this report.	U8	0x00 - 0xFF	M
Event Report Entry	See Table 110 and Table 112.	-	-	M

Table 109 - Byte order and organization of an Event attribute reporting rule.

8	7	6	5	4	3	2	1	Octet
Report Type	Report ID							1
Address Type	Receiver Device Address (MSB)							2
Receiver Device Address (LSB)								3
Receiver Unit ID								4
Number of Report Entries								5
Event Report Entry 1								6
...								:
Event Report Entry n								7 + n

Table 110 - Information stored in each Event Report Entry.

Field Name	Field Description	Type	Value	M/O
Unit ID	Identifier of the unit to which the attribute to report, belongs to.	U8	0x00 - 0xFE	M
Interface Type	Type of the interface to which the attribute to report, belongs to.	U1	0x00 - Client 0x01 - Server	M
Interface UID	Identifier of the interface to which the attribute to report, belongs to.	U15	0x0000 - 0x7FFE	M
Attribute Pack UID	See Table 101 in section 6.6 for detailed information.	U8	0x00 - 0xFF	M
Type Of Reporting	Reporting Type.	U8	0x00 - Change of value (COV) 0x01 - High Threshold (HT) 0x02 - Low Threshold (LT) 0x03 - Equal	M
Reporting Data Field	See Table 111 for data field specification.	-	-	M

Table 111 - Specification of each Type of Reporting and their data fields.

Type Of Reporting	Type description	Data Field		
		Type	Value	
0x00 - COV	A notification will be sent whenever the attribute value is modified. Data field values between 0x01 and 0xFF are used as percentage of change, meaning a notification will only be sent if attribute changed more than X percentage. The special value 0x00, triggers a notification independently of how much attribute changed.	U8	0x00 - 0xFF	
0x01 - HT	A notification will be sent when the attribute value gets higher than “high threshold value”. Data field starts with one byte indicating the size (in bytes) of “high threshold value”, followed by the value itself stored MSB first.	Number of bytes in value	U8	0x00 - 0xFF
		-	-	High threshold value
0x02 - LH	A notification will be sent when the attribute value gets lower than “low threshold value”. Data field starts with one byte indicating the size (in bytes) of “low threshold value”, followed by the value itself stored MSB first.	Number of bytes in value	U8	0x00 - 0xFF
		-	-	Low threshold value
0x03 - Equal	A notification will be sent when the attribute value equals the defined “comparison value”. Data field starts with one byte indicating the size (in bytes) of “comparison value”, followed by the value itself stored MSB first.	Number of bytes in value	U8	0x00 - 0xFF
		-	-	Comparison Value

Table 112 - Byte order and organization of an Event Report Entry, for any *Attribute Pack ID* except 0xFF - Dynamic.

8	7	6	5	4	3	2	1	Octet
Unit ID								1
Interface Type	Interface UID (MSB)							
Interface UID (LSB)								3
Attribute Pack ID								4
Type of Reporting								5
Reporting Data Field (MSB)								6
...								:
Reporting Data Field (LSB)								7 + n

Table 113 - Additional data stored in each Event Report Entry, for the special *Attribute Pack ID* 0xFF - Dynamic.

Field Name	Field Description	Type	Value	M/O
Number of Attributes	Number of attributes, present in the command, for which reporting rules must be created.	U8	0x00 - 0xFF	M
Attribute ID	Identifier of an attribute to report.	U8	0x00 - 0xFF	M

Table 114 - Byte order and organization of an Event Report Entry, for the special *Attribute Pack ID* 0xFF - Dynamic.

8	7	6	5	4	3	2	1	Octet
Unit ID								1
Interface Type	Interface UID (MSB)							
Interface UID (LSB)								3
Attribute Pack ID = 0xFF								4
Number of Attributes								5
Attribute ID								6
Type of Reporting								7
Reporting Data Field (MSB)								8
...								:
Reporting Data Field (LSB)								9 + n
...								:
Attribute ID								10 + n
Type of Reporting								11 + n
Reporting Data Field (MSB)								12 + n
...								:
Reporting Data Field (LSB)								13 + n

6.6.2 Server Attributes

Table 115 - Attribute Reporting interface attributes.

Attribute ID	Attribute Name	Attribute Type	Attribute Values	Attribute Access	M/O
0x01	Number Of Reports	U16	0x0000 - 0xFFFF	Read Only	O
0x02	Number Of Periodic Reports	U16	0x0000 - 0xFFFF	Read Only	O
0x03	Number Of Event Reports	U16	0x0000 - 0xFFFF	Read Only	O

6.6.2.1 Number of Reports

Number of Reports attribute records the total number of reports registered on any given time.

6.6.2.2 Number of Periodic Reports

Number of Periodic Reports attribute records the total number of periodic reports registered on any given time.

6.6.2.3 Number of Event Reports

Number of Event Reports attribute records the total number of event reports registered on any given time.

6.6.3 Client Attributes

None.

6.6.4 Server to Client Commands

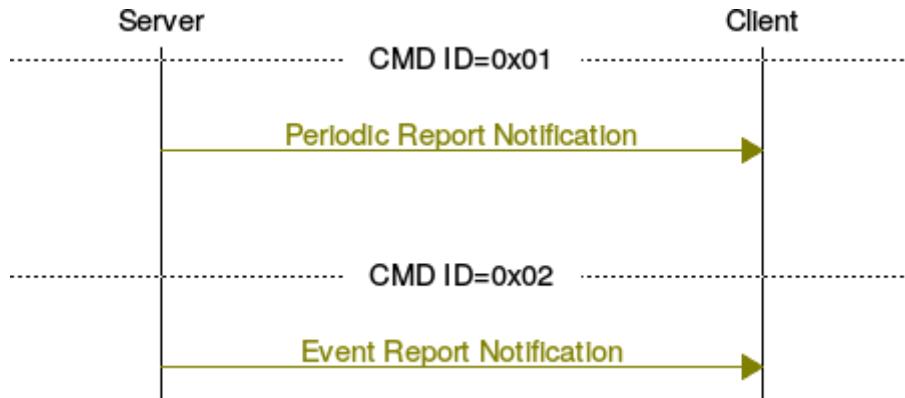


Figure 34 - Attribute Reporting Interface Server: Commands.

Table 116 - Implementation status of Attribute Reporting Interface Server commands.

Command	Reference	Client Role	Server Role	Response
Periodic Report Notification	6.6.4.1	M	M	N/A
Event Report Notification	6.6.4.2	M	M	N/A

6.6.4.1 Periodic Report Notification

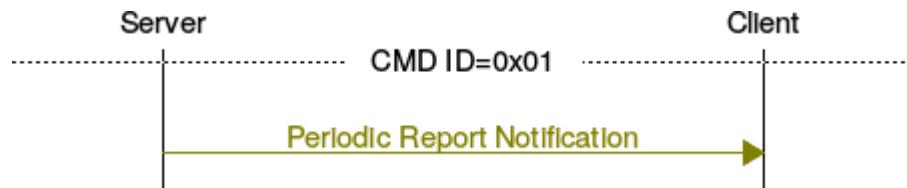


Figure 35 - Attribute Reporting Interface Server: *Periodic Report Notification* command.

This command is used by the server to send the information associated with a periodic report whenever its associated time expires and is, consequently, reset. The information is sent to a client unit whose address is on the previously created Periodic Report Rule (6.6.1.1).

The command must provide the information described in Table 117, organized according to Table 118.

Table 117 - Data in the payload of a *Periodic Report Notification* command.

Field Name	Field Description	Type	Value	M/O
Report Type	Distinguishes if the report is periodic or event driven.	U1	0x00 - Periodic 0x01 - Event	M
Report ID	Report identifier, unique per device with server role implementation.	U7	0x00 - 0x7E	M
Content of Periodic Report Entry	See Table 119 and Table 120.	-	-	M

Table 118 - Data ordering of the payload of a *Periodic Report Notification* command.

8	7	6	5	4	3	2	1	Octet							
Report Type	Report ID							1							
Content of Periodic Report Entry 1								2							
...								:							
Content of Periodic Report Entry n								3 + n							

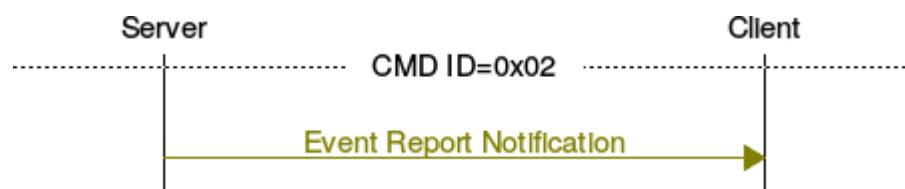
Table 119 - Information in a Content of Periodic Report Entry structure.

Field Name	Field Description	Type	Value	M/O
Unit ID	Identifier of the unit to which the attribute to report, belongs to.	U8	0x00 - 0xFE	M
Interface Type	Type of the interface to which the attribute to report, belongs to.	U1	0x00 - Client 0x01 - Server	M
Interface UID	Identifier of the interface to which the attribute to report, belongs to.	U15	0x0000 - 0x7FFE	M
Number of Attributes	Number of attributes in the report.	U8	0x00 - 0xFF	M
Attribute ID	Identifier of the attribute.	U8	0x00 - 0xFF	M
Attribute Value	Content stored in the attribute identified by <i>Attribute ID</i> field.	-	Size dependent	M

Table 120 - Byte order and organization of a Content of Periodic Report Entry structure.

8	7	6	5	4	3	2	1	Octet
Unit ID								1
Interface Type	Interface UID (MSB)							
Interface UID (LSB)								3
Number of Attributes								4
Attribute ID								5
Attribute Value (MSB)								6
...								:
Attribute Value (LSB)								7 + n
...								:
Attribute ID								8 + n
Attribute Value (MSB)								9 + n
...								:
Attribute Value (LSB)								10 + n

6.6.4.2 Event Report Notification

**Figure 36 - Attribute Reporting Interface Server: *Event Report Notification* command.**

This command is used by the server to send the information associated with an event report whenever its associated event triggers. The information is sent to a client unit whose address is on the previously created Event Report Rule (6.6.1.2).

The command must provide the information described in Table 121, organized according to Table 122.

Table 121 - Data in the payload of an *Event Report Notification* command.

Field Name	Field Description	Type	Value	M/O
Report Type	Distinguishes if the report is periodic or event driven.	U1	0x00 - Periodic 0x01 - Event	M
Report ID	Report identifier, unique per device with server role implementation.	U7	0x00 - 0x7E	M
Content of Event Report Entry	See Table 123 and Table 124.	-	-	M

Table 122 - Data ordering of the payload of an *Event Report Notification* command.

8	7	6	5	4	3	2	1	Octet							
Report Type	Report ID							1							
Content of Event Report Entry 1								2							
...								:							
Content of Event Report Entry n								3 + n							

Table 123 - Information in a Content of Event Report Entry structure.

Field Name	Field Description	Type	Value	M/O
Unit ID	Identifier of the unit to which the attribute to report, belongs to.	U8	0x00 - 0xFE	M
Interface Type	Type of the interface to which the attribute to report, belongs to.	U1	0x00 - Client 0x01 - Server	M
Interface UID	Identifier of the interface to which the attribute to report, belongs to.	U15	0x0000 - 0x7FFE	M
Number of Attributes	Number of attributes in the report.	U8	0x00 - 0xFF	M
Attribute ID	Identifier of the attribute.	U8	0x00 - 0xFF	M
Type Of Reporting	Reporting Type.	U8	0x00 - Change of value (COV) 0x01 - High Threshold (HT) 0x02 - Low Threshold (LT) 0x03 - Equal	M
Attribute Value	Content stored in the attribute identified by <i>Attribute ID</i> field.	-	Size dependent	M

Table 124 - Byte order and organization of a Content of Event Report Entry structure.

8	7	6	5	4	3	2	1	Octet
								Unit ID
Interface Type	Interface UID (MSB)							
	Interface UID (LSB)							
	Number of Attributes							
	Attribute ID							
	Type of Reporting							
	Attribute Value (MSB)							
	...							
	Attribute Value (LSB)							
	...							
	Attribute ID							
	Type of Reporting							
	Attribute Value (MSB)							
	...							
	Attribute Value (LSB)							

6.6.5 Client to Server Commands

This interface defines the client to server mandatory commands depicted in Figure 37.

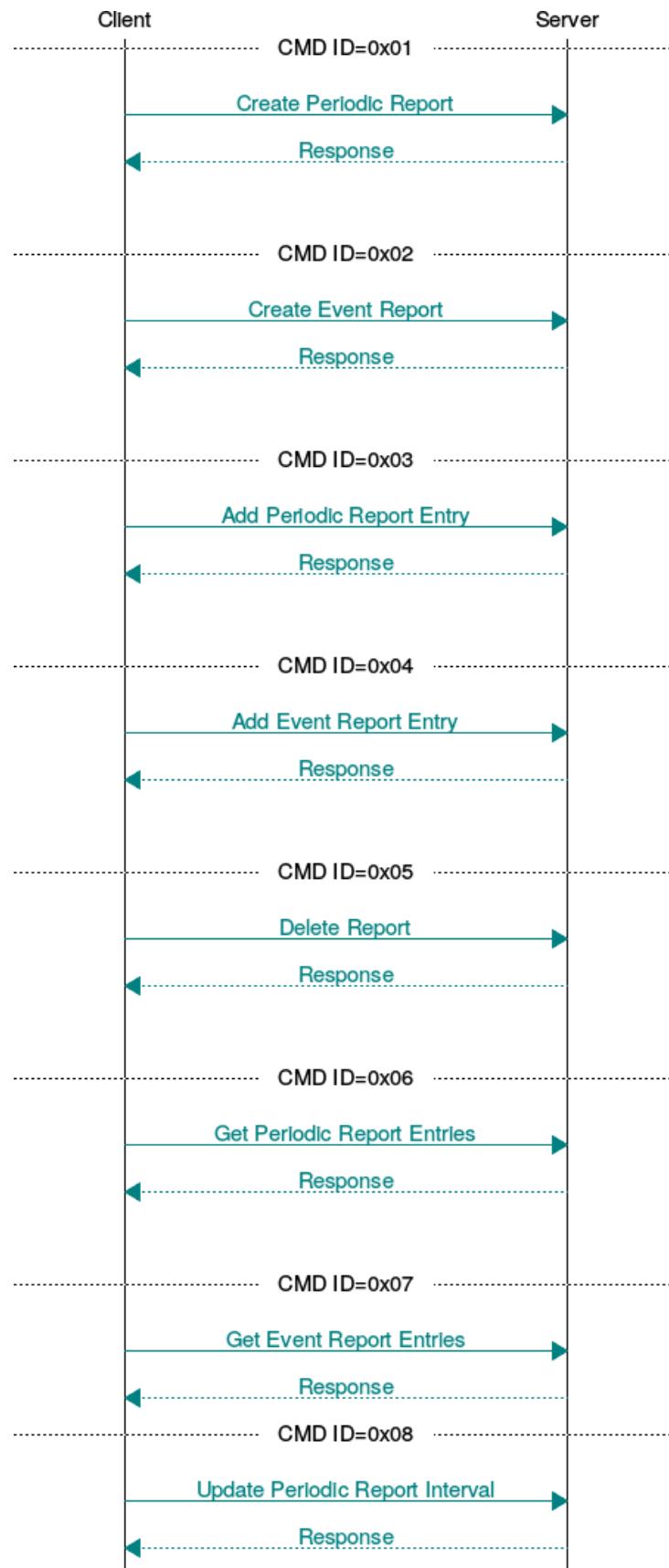
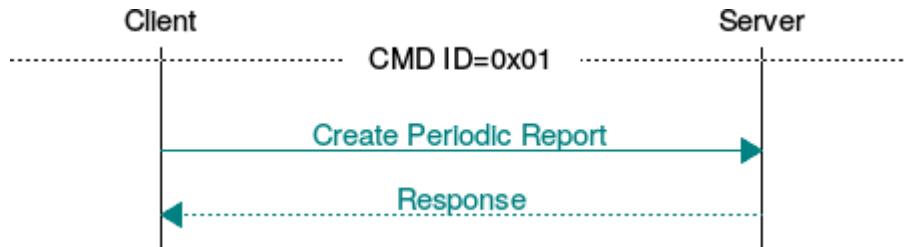


Figure 37 - Attribute Reporting Interface Client commands.

Table 125 - Implementation status of Attribute Reporting Interface Client commands.

Command	Reference	Client Role	Server Role	Response
Create Periodic Report	6.6.5.1	M	M	M
Create Event Report	6.6.5.2	M	M	M
Add Periodic Report Entry	6.6.5.3	M	M	O
Add Event Report Entry	6.6.5.4	M	M	O
Delete Report	6.6.5.5	M	M	M
Get Periodic Report Entries	6.6.5.6	O	O	M
Get Event Report Entries	6.6.5.7	O	O	M
Update Periodic Report Interval	6.6.5.8	O	O	O

6.6.5.1 Create Periodic Report

**Figure 38 - Attribute Reporting: *Create Periodic Report* command.**

This command allows a unit (implementing the attribute reporting client interface) to create an empty report with the periodic time interval specified and the full address of which device-unit pair should receive notifications. In return, if the operation is successful, a *Report ID* is obtained which allows for adding (6.6.5.3) attributes to be monitored according to the characteristics of the created report.

The periodic time interval is to be used on a free running timer that should start when the first attribute is added to the report; subsequent attribute additions should not reset the timer.

The command must provide the information described in Table 126, organized according to Table 127.

The response must provide the information described in Table 128, organized according to Table 129. If the Response Code indicates a failure then no other fields are required to be present and the response has only 1 byte in total.

Table 126 - Data in the payload of a *Create Periodic Report* command.

Field Name	Field Description	Type	Value	M/O
Address Type	Indicates the type of <i>Receiver Device Address</i> .	U1	0x00 - Individual 0x01 – Group	M
Receiver Device Address	Address of device that requested the notification.	U15	0x0000 - 0x7FFE	M
Receiver Unit ID	Identifier of the unit that implements Attribute Reporting Client interface and has requested the notification.	U8	0x00 - 0xFE	M
Periodic Interval Value	Specifies the time, in seconds, which dictates the sending of a notification.	U32	0x00000000 - 0xFFFFFFFF	M

Table 127 - Data ordering of the payload of a *Create Periodic Report* command.

8	7	6	5	4	3	2	1	Octet
Address Type	Receiver Device Address (MSB)							1
	Receiver Device Address (LSB)							2
	Receiver Unit ID							3
	Periodic Interval Value (MSB)							4
	...							⋮
	Periodic Interval Value (LSB)							5 + n

Table 128 - Data in the payload of a Response to a *Create Periodic Report* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0x03 - Fail: Not supported 0xFE - Fail: Not enough resources 0xFF - Fail: Unknown reason	M
Report Type	Distinguishes if the report is periodic or event driven.	U1	0x00 - Periodic 0x01 - Event	M
Report ID	Report identifier, unique per device with server role implementation.	U7	0x00 - 0x7E	M

Table 129 - Data ordering of the payload of a Response to a *Create Periodic Report* command.

8	7	6	5	4	3	2	1	Octet
Report Type	Response Code							1
	Report ID							2

6.6.5.2 Create Event Report

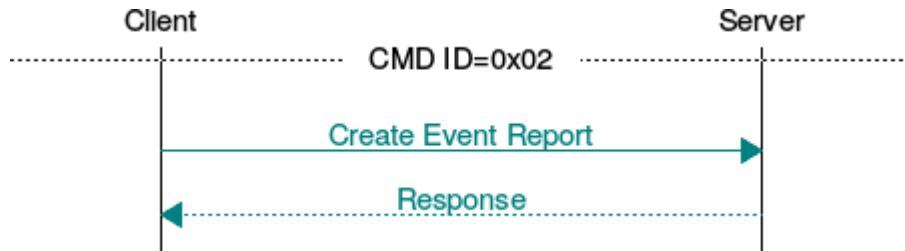


Figure 39 - Attribute Reporting: *Create Event Report* command.

This command allows a unit (implementing the attribute reporting client interface) to create an empty report with the full address of which device-unit pair should receive notifications, specified. In return, if the operation is successful, a *Report ID* is obtained which allows for adding (6.6.5.4) attributes to be monitored according to specified characteristics.

The command must provide the information described in Table 130, organized according to Table 131.

The response must provide the information described in Table 132, organized according to Table 133. If the Response Code indicates a failure then no other fields are required to be present and the response has only 1 byte in total.

Table 130 - Data in the payload of a *Create Event Report* command.

Field Name	Field Description	Type	Value	M/O
Address Type	Indicates the type of <i>Receiver Device Address</i> .	U1	0x00 - Individual 0x01 – Group	M
Receiver Device Address	Address of device that requested the notification.	U15	0x0000 - 0x7FFE	M
Receiver Unit ID	Identifier of the unit that implements Attribute Reporting Client interface and has requested the notification.	U8	0x00 - 0xFE	M

Table 131 - Data ordering of the payload of a *Create Event Report* command.

8	7	6	5	4	3	2	1	Octet	
Address Type	Receiver Device Address (MSB)								1
Receiver Device Address (LSB)									2
Receiver Unit ID									3

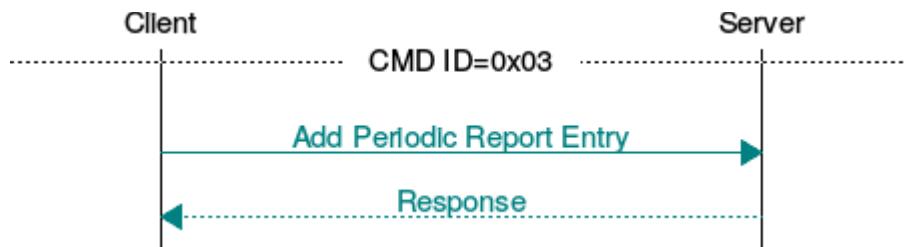
Table 132 - Data in the payload of a Response to a *Create Event Report* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0x03 - Fail: Not supported 0xFE - Fail: Not enough resources 0xFF - Fail: Unknown reason	M
Report Type	Distinguishes if the report is periodic or event driven.	U1	0x00 - Periodic 0x01 - Event	M
Report ID	Report identifier, unique per device with server role implementation.	U7	0x00 - 0x7E	M

Table 133 - Data ordering of the payload of a Response to a *Create Event Report* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
Report Type	Report ID							

6.6.5.3 Add Periodic Report Entry

**Figure 40 - Attribute Reporting: *Add Periodic Report Entry* command.**

This command allows a unit (implementing the attribute reporting client interface) to add attributes to an already existent report (by using that report's ID). Added attributes will be sent (6.6.4.1), to the report receiver, with the time cadence specified on the report. Note that this time cadence does not reset when a new attribute is added, so the first notification may be sent earlier than expected.

The command must provide the information described in Table 134, organized according to Table 135.

The response must provide the information described in Table 136, organized according to Table 137.

Table 134 - Data in the payload of an *Add Periodic Report Entry* command.

Field Name	Field Description	Type	Value	M/O
Report Type	Distinguishes if the report is periodic or event driven.	U1	0x00 - Periodic 0x01 - Event	M
Report ID	Report identifier, unique per device with server role implementation.	U7	0x00 - 0x7E	M
Number of Report Entries	Number of currently defined entries in this report.	U8	0x00 - 0xFF	M
Periodic Report Entry	See Table 104 and Table 105, also if required, see Table 106 and Table 107. All tables are in section 6.6.1.1.	-	-	M

Table 135 - Data ordering of the payload of an *Add Periodic Report Entry* command.

8	7	6	5	4	3	2	1	Octet
Report Type	Report ID							1
	Number of Report Entries							2
	Periodic Report Entry 1							3
	...							:
	Periodic Report Entry n							4 + n

Table 136 - Data in the payload of a Response to an *Add Periodic Report Entry* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0x03 - Fail: Not supported 0xFE - Fail: Not enough resources 0xFF - Fail: Unknown reason	M
Report Type	Distinguishes if the report is periodic or event driven.	U1	0x00 - Periodic 0x01 - Event	M
Report ID	Report identifier, unique per device with server role implementation.	U7	0x00 - 0x7E	M

Table 137 - Data ordering of the payload of a Response to an *Add Periodic Report Entry* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
Report Type	Report ID							2

6.6.5.4 Add Event Report Entry

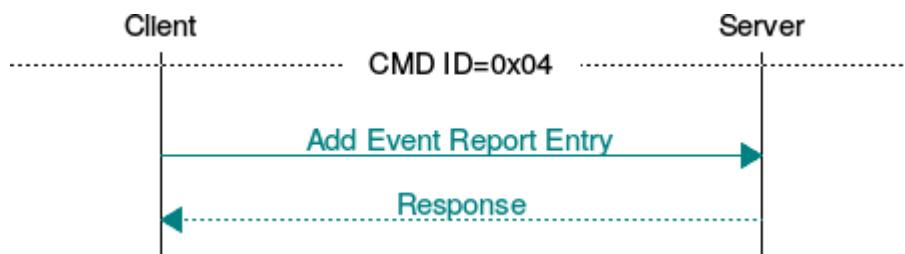


Figure 41 - Attribute Reporting: Add Event Report Entry command.

This command allows a unit (implementing the attribute reporting client interface) to add attributes to an already existent report (by using that report's ID). Each added attribute will specify the type of monitoring to which it should be submitted. Whenever the attribute value triggers its associated event, such value will be sent (6.6.4.2) to the report receiver.

The command must provide the information described in Table 138, organized according to Table 139.

The response must provide the information described in Table 140, organized according to Table 141.

Table 138 - Data in the payload of an *Add Event Report Entry* command.

Field Name	Field Description	Type	Value	M/O
Report Type	Distinguishes if the report is periodic or event driven.	U1	0x00 - Periodic 0x01 - Event	M
Report ID	Report identifier, unique per device with server role implementation.	U7	0x00 - 0x7E	M
Number of Report Entries	Number of currently defined entries in this report.	U8	0x00 - 0xFF	M
Event Report Entry	See Table 110 and Table 112, also if required, see Table 113 and Table 114. All tables are in section 6.6.1.2.	-	-	M

Table 139 - Data ordering of the payload of an *Add Event Report Entry* command.

8	7	6	5	4	3	2	1	Octet
Report Type	Report ID							1
	Number of Report Entries							2
	Event Report Entry 1							3
	...							:
	Event Report Entry n							4 + n

Table 140 - Data in the payload of a Response to an *Add Event Report Entry* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0x03 - Fail: Not supported 0xFE - Fail: Not enough resources 0xFF - Fail: Unknown reason	M
Report Type	Distinguishes if the report is periodic or event driven.	U1	0x00 - Periodic 0x01 - Event	M
Report ID	Report identifier, unique per device with server role implementation.	U7	0x00 - 0x7E	M

Table 141 - Data ordering of the payload of a Response to an *Add Event Report Entry* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
Report Type	Report ID							

6.6.5.5 Delete Report

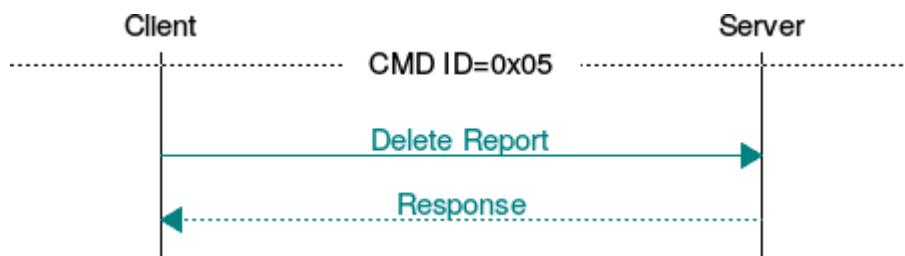


Figure 42 - Attribute Reporting: Delete Report command.

This command allows a unit (implementing the attribute reporting client interface) to delete an existent report. The server will clear any information related to the deleted report and the device-unit pair which received the notifications will cease to receive them. If the special *Report ID* value - 0x7F - is used then all reports of type *Report Type* currently registered are deleted.

The command must provide the information described in Table 142, organized according to Table 143.

The response must provide the information described in Table 144, organized according to Table 145.

Table 142 - Data in the payload of a Delete Report command.

Field Name	Field Description	Type	Value	M/O
Report Type	Distinguishes if the report is periodic or event driven.	U1	0x00 - Periodic 0x01 - Event	M
Report ID	Report identifier, unique per device with server role implementation.	U7	0x00 - 0x7F	M

Table 143 - Data ordering of the payload of a Delete Report command.

8	7	6	5	4	3	2	1	Octet
Report Type	Report ID							

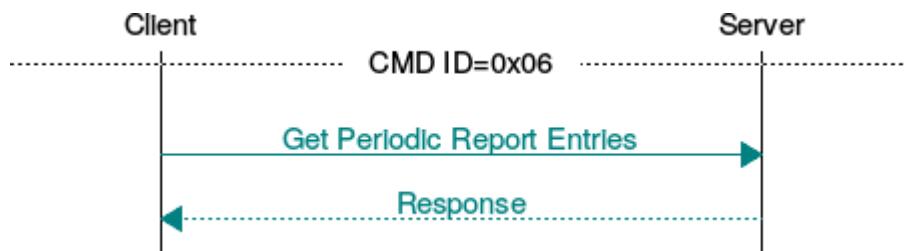
Table 144 - Data in the payload of a Response to a Delete Report command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0x03 - Fail: Not supported 0xFE - Fail: Not enough resources 0xFF - Fail: Unknown reason	M
Report Type	Distinguishes if the report is periodic or event driven.	U1	0x00 - Periodic 0x01 - Event	M
Report ID	Report identifier, unique per device with server role implementation.	U7	0x00 - 0x7F	M

Table 145 - Data ordering of the payload of a Response to a *Delete Report* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
Report Type	Report ID						2	

6.6.5.6 Get Periodic Report Entries

**Figure 43 - Attribute Reporting: *Get Periodic Report Entries* command.**

This command allows a unit (implementing the attribute reporting client interface) to obtain all periodic reporting rules from the specified report.

The command must provide the information described in Table 146, organized according to Table 147.

The response must provide the information described in Table 148, organized according to Table 149.

Table 146 - Data in the payload of a *Get Periodic Report Entries* command.

Field Name	Field Description	Type	Value	M/O
Report Type	Distinguishes if the report is periodic or event driven.	U1	0x00 - Periodic 0x01 - Event	M
Report ID	Report identifier, unique per device with server role implementation.	U7	0x00 - 0x7E	M

Table 147 - Data ordering of the payload of a *Get Periodic Report Entries* command.

8	7	6	5	4	3	2	1	Octet
Report Type	Report ID						1	

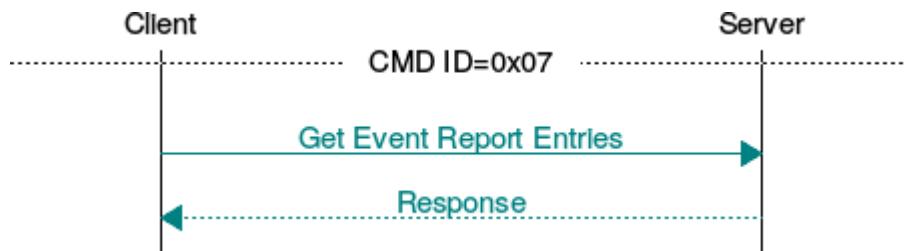
Table 148 - Data in the payload of a Response to a *Get Periodic Report Entries* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0x03 - Fail: Not supported 0xFFE - Fail: Not enough resources 0xFF - Fail: Unknown reason	M
Periodic Report Rule	See Table 102 and Table 103 from section 6.6.1.1.	-	-	M

Table 149 - Data ordering of the payload of a Response to a *Get Periodic Report Entries* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
Periodic Report Rule (See Table 102 and Table 103 from section 6.6.1.1)								2

6.6.5.7 Get Event Report Entries

**Figure 44 - Attribute Reporting: *Get Event Report Entries* command.**

This command allows a unit (implementing the attribute reporting client interface) to obtain all event reporting rules from the specified report.

The command must provide the information described in Table 150, organized according to Table 151.

The response must provide the information described in Table 152, organized according to Table 153.

Table 150 - Data in the payload of a *Get Event Report Entries* command.

Field Name	Field Description	Type	Value	M/O
Report Type	Distinguishes if the report is periodic or event driven.	U1	0x00 - Periodic 0x01 - Event	M
Report ID	Report identifier, unique per device with server role implementation.	U7	0x00 - 0x7E	M

Table 151 - Data ordering of the payload of a *Get Event Report Entries* command.

8	7	6	5	4	3	2	1	Octet	
Report Type	Report ID								1

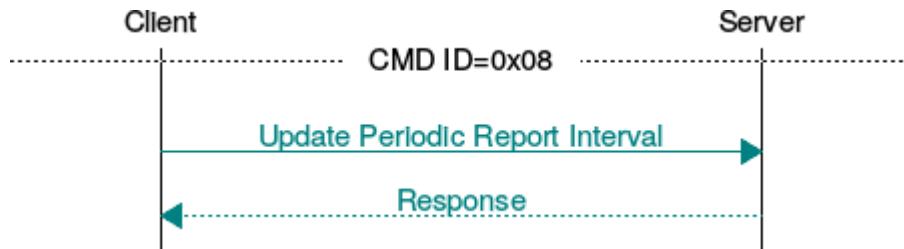
Table 152 - Data in the payload of a Response to a *Get Event Report Entries* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0x03 - Fail: Not supported 0xFE - Fail: Not enough resources 0xFF - Fail: Unknown reason	M
Event Report Rule	See Table 108 and Table 109 from section 6.6.1.2.	-	-	M

Table 153 - Data ordering of the payload of a Response to a *Get Event Report Entries* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
Event Report Rule (See Table 108 and Table 109 from section 6.6.1.2)								2

6.6.5.8 Update Periodic Report Interval

**Figure 45 - Attribute Reporting: *Update Periodic Report Interval* command.**

This command allows a unit (implementing the attribute reporting client interface) to update the interval of an existent report.

The command must provide the information described in Table 154, organized according to Table 155

The response must provide the information described in Table 156, organized according to Table 157.

Table 154 - Data in the payload of an *Update Periodic Report Interval* command.

Field Name	Field Description	Type	Value	M/O
Report Type	Distinguishes if the report is periodic or event driven.	U1	0x00 – Periodic 0x01 – Event	M
Report ID	Report identifier, unique per device with server role implementation.	U7	0x00 – 0x7F	M
Periodic Interval Value	Specifies the time, in seconds, which dictates the sending of a notification.	U32	0x00000000 – 0xFFFFFFFF	M

Table 155 - Data ordering of the payload of an *Update Periodic Report Interval* command.

8	7	6	5	4	3	2	1	Octet
Report Type	Report ID							1
	Periodic Interval Value (MSB)							2
	Periodic Interval Value							3
	Periodic Interval Value							4
	Periodic Interval Value (LSB)							5

Table 156 - Data in the payload of a Response to an *Update Periodic Report Interval* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0x03 - Fail: Not supported 0xFE - Fail: Not enough resources 0xFF - Fail: Unknown reason	M

Table 157 - Data ordering of the payload of a Response to an *Update Periodic Report Interval* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1

6.7 Batch Program Management Interface

This interface defines the commands and attributes to be used by a device implementing Batch Program Management Service.

Each batch program is identifiable by a unique²¹ number called *Program ID*, which can take any value in the range 0x00 to 0xFE or the special value 0xFF. This special value signifies “any available identifier” and is only used by *Define Program* (6.7.5.1).

6.7.1 Server Interface Data Structures

Any device that implements this service as a server role is required to store batch program entries.

A batch program entry has the information described in Table 158, organized (for messaging purposes) as shown in Table 159.

An action entry the information shown in Table 160, organized (for messaging purposes) as shown in Table 161.

Table 158 - Information stored for each Batch Program entry.

Field Name	Field Description	Type	Value	M/O
Program ID	Program identifier, unique per device with server role implementation.	U8	0x00 - 0xFE ²²	M
Program Name	Human readable name intended for the program being created.	Length ²³	U8	0x00 - 0x20
		Character String	U8	0x20 - 0xFE (each U8)
Number of Action Entries	Number of action entries defined by the program.	U8	0x01 - 0xFF	M
Action Entry	See Table 160 and Table 161.	-	-	M

Table 159 - Byte order and organization of a Batch Program Entry.

8	7	6	5	4	3	2	1	Octet
Program ID								1
Program Name Length								2
Program Name Character i								3
...								:
Program Name Character i+n								4 + n
Number of Actions								5 + n
Action Entry 1								6 + n
...								:
Action Entry n								7 + n ²⁴

²¹ Unique only at the device with server role implementation level that is, a *Program ID* is not unique across different devices implementing this interface’s server role.

²² The possible values can be further restricted by the *Maximum Number of Entries* server attribute (see 6.7.2.1).

²³ *Length* is the first byte of the data array; therefore it precedes the character string.

²⁴ The value **n** represents an accumulation of an arbitrary number of bytes.

Table 160 - Information stored in each Action entry.

Field Name	Field Description	Type	Value
Unit ID	Identifier of the Unit to which this action is to be sent.	U8	0x00 - 0xFE
Message Type	Message Type Code that specifies how <i>Interface Member</i> is to be interpreted (for details see [REF 2])	U8	0x01 - Command 0x06 - Set Attribute Request
Interface Type	Type of the interface to which this action is to be sent.	U1	0x00 - Client 0x01 - Server
Interface UID	Identifier of the interface to which this action is to be sent.	U15	0x0000 - 0x7FFE
Interface Member	Identifier of the interface member that is to be invoked (Command ID) or acted upon (Attribute ID), depending on the specified <i>Message Type</i> .	U8	0x00 – 0xFF
Payload Size	Size of the attribute or of the payload of the command, as defined in the interface with <i>Interface UID</i> .	U16	0x0000 - 0x01E5
Payload	Content of the attribute or of the payload of the command, as defined in the interface with <i>Interface UID</i> .	-	-

Table 161 - Byte order and organization of an Action Entry.

8	7	6	5	4	3	2	1	Octet
								1
								2
Interface Type								3
								4
								5
							Payload Size (MSB)	6
								7
								8
								:
								9 + n

6.7.2 Server Attributes

Table 162 - Batch Program Management Interface: Server Attributes.

Attribute ID	Attribute Name	Attribute Type	Attribute Values	Attribute Access	M/O
0x01	Maximum Number of Entries	U8	0x00 - 0xFF	Read Only	M
0x02	Number of Entries	U8	0x00 - 0xFF	Read Only	M

6.7.2.1 Maximum Number of Entries

Maximum Number of Entries attribute specifies the maximum number of programs that can be defined.

6.7.2.2 Number of Entries

Number of Entries attribute records the number of defined programs on any given time.

6.7.3 Client Attributes

None.

6.7.4 Server to Client Commands

None.

6.7.5 Client to Server Commands

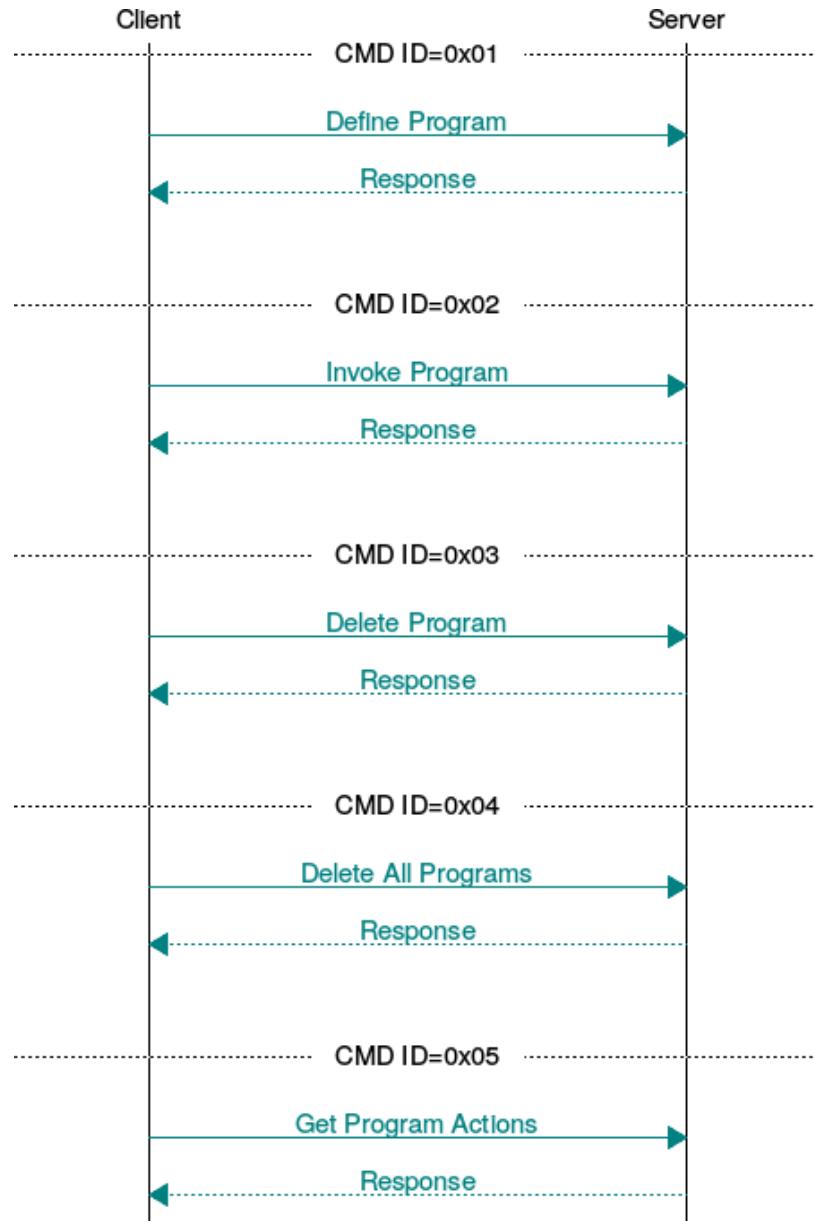


Figure 46 - Batch Program Management Interface: Client commands.

Table 163 - Implementation status of Batch Program Management Interface Client commands.

Command	Reference	Client Role	Server Role	Response
Define Program	6.7.5.1	M	M	M
Invoke Program	6.7.5.2	M	M	M
Delete Program	6.7.5.3	M	M	M
Delete All Programs	6.7.5.4	M	M	M
Get Program Actions	6.7.5.5	O	M	M

6.7.5.1 Define Program

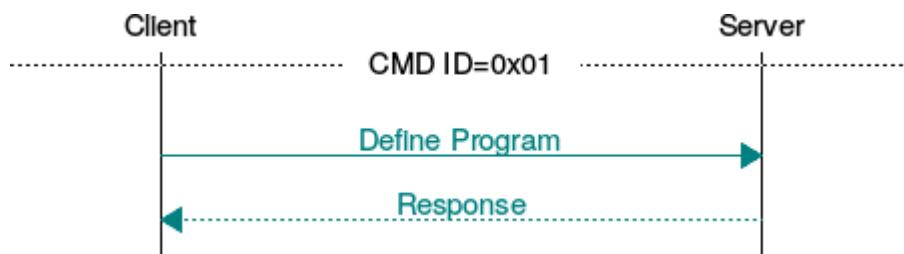


Figure 47 - Batch Program Management Interface: *Define Program* command.

This command allows a batch program to be defined. A batch program is an aggregation of one or more actions. An action is either a *Command*²⁵ or a *Set Attribute Request*²⁶, with its respective payload (when applicable) bundled together with both the Unit ID and the Interface ID to which it is destined to. If the special *Program ID* value – 0xFF – is used, the program being defined will be stored on the next available identifier.

The command must provide the information described in Table 164, organized according to Table 165.

The response must provide the information described in Table 166, organized according to Table 167. If the Response Code indicates a failure then no other fields are required to be present and the response has only 1 byte in total.

Table 164 - Data in the payload of a *Define Program* command.

Field Name	Field Description	Type	Value	M/O
Program ID	Program identifier or special value (0xFF)	U8	0x00 - 0xFF	M
Program Name	Human readable name intended for the program being created.	Length ²⁷	0x00 - 0x20	M
		Character String	0x20 - 0xFE (each U8)	
Number of Action Entries	Number of action entries defined by the program.	U8	0x01 - 0xFF	M
Action Entry	See Table 160 and Table 161, specified in section 6.7.1.	-	-	M

²⁵ Message Type Code 0x01 as defined in [REF 2].

²⁶ Message Type Code 0x06 as defined in [REF 2].

²⁷ Length is the first byte of the data array; therefore it precedes the character string.

Table 165 - Data ordering of the payload of a *Define Program* command.

8	7	6	5	4	3	2	1	Octet
Program ID								1
Program Name Length								2
Program Name Character i								3
...								:
Program Name Character i+n								4 + n
Number of Actions								5 + n
Action Entry 1								6 + n
...								:
Action Entry n								7 + n ²⁸

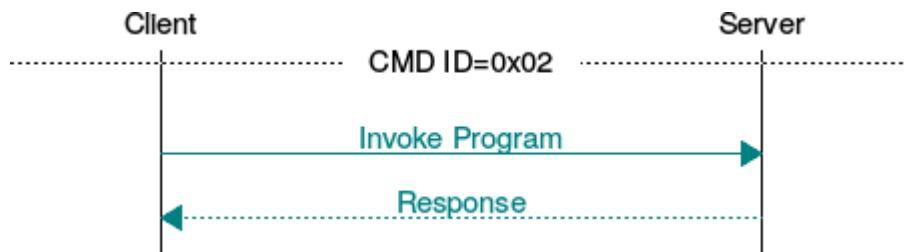
Table 166 - Data in the payload of a Response to a *Define Program* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0xFE - Fail: Not enough resources 0xFF - Fail: Unknown reason	M
Program ID	Program identifier, unique per device with server role implementation.	U8	0x00 - 0xFE	M

Table 167 - Data ordering of the payload of a Response to a *Define Program* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
Program ID								2

6.7.5.2 Invoke Program

**Figure 48 - Batch Program Management Interface: *Invoke Program* command.**

This command allows a previously defined program to be invoked. All actions defined by the specified program will be executed resulting in one or more messages being sent.

²⁸ The value **n** represents an accumulation of an arbitrary number of bytes.

The command must carry a single byte with the information described in Table 168.

The response must provide the information described in Table 169, organized according to Table 170. If the Response Code indicates a failure then no other fields are required to be present and the response has only 1 byte in total.

Table 168 - Data in the payload of an *Invoke Program* command.

Field Name	Field Description	Type	Value	M/O
Program ID	Program identifier, unique per device with server role implementation.	U8	0x00 - 0xFE	M

Table 169 - Data in the payload of a Response to an *Invoke Program* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0xFE - Fail: Not enough resources 0xFF - Fail: Unknown reason	M
Program ID	Program identifier, unique per device with server role implementation.	U8	0x00 - 0xFE	M

Table 170 - Data ordering of the payload of a Response to an *Invoke Program* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
Program ID								2

6.7.5.3 Delete Program

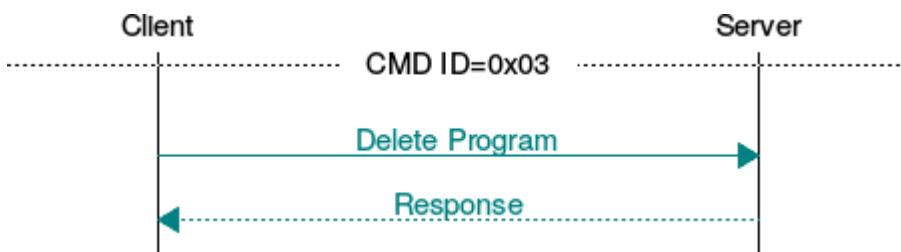


Figure 49 - Batch Program Management Interface: *Delete Program* command.

This command allows for a defined program to be deleted. The server implementation will clear any information related to the deleted program and its associated action entries, releasing resources that were previously allocated. As a consequence the *Program ID* of the deleted program is made available for new programs.

The command must carry a single byte with the information described in Table 171.

The response must provide the information described in Table 172, organized according to Table 173. If the Response Code indicates a failure then no other fields are required to be present and the response has only 1 byte in total.

Table 171 - Data in the payload of a *Delete Program* command.

Field Name	Field Description	Type	Value	M/O
Program ID	Program identifier, unique per device with server role implementation.	U8	0x00 - 0xFE	M

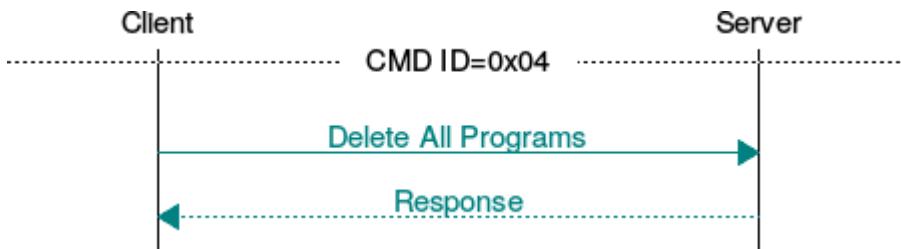
Table 172 - Data in the payload of a Response to a *Delete Program* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0xFF - Fail: Unknown reason	M
Program ID	Program identifier, unique per device with server role implementation.	U8	0x00 - 0xFE	M

Table 173 - Data ordering of the payload of a Response to a *Delete Program* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
Program ID								2

6.7.5.4 Delete All Programs

**Figure 50 - Batch Program Management Interface: *Delete All Programs* command.**

This command allows for all defined programs to be deleted. The server implementation will clear all information related to existing programs and their associated action entries, releasing all the allocated resources.

The command has no payload, but its response has a single byte – Response Code – with one of the values described in Table 174.

Table 174 - Data in the payload of a Response to a *Delete All Programs* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0xFF - Fail: Unknown reason	M

6.7.5.5 Get Program Actions

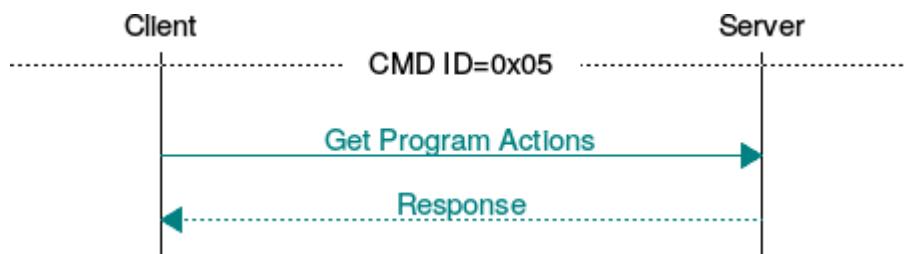


Figure 51 - Batch Program Management Interface: *Get Program Actions* command.

This command allows for all action entries, from the specified program, to be obtained.

The command must carry a single byte with the information described in Table 175.

The response must provide the information described in Table 176, organized according to Table 177.

Table 175 - Data in the payload of a *Get Program Actions* command.

Field Name	Field Description	Type	Value	M/O
Program ID	Program identifier, unique per device with server role implementation.	U8	0x00 - 0xFE	M

Table 176 - Data in the payload of a Response to a *Get Program Actions* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0xFE - Fail: Not enough resources 0xFF - Fail: Unknown reason	M
Program Entry	See Table 158 specified in section 6.7.1.	-	-	M

Table 177 - Data ordering of the payload of a response to *Get Program Actions* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
Program Entry (See Table 159 from section 6.7.1)								2

6.8 Event Scheduling Interface

This interface defines the commands and attributes to be used by a device implementing Event Scheduling Service.

Each event entry is identifiable by a unique²⁹ number called *Event ID*, which can take any value in the range 0x00 to 0xFE or the special value 0xFF. This special value signifies “any available identifier” and is only used by *Define Event* (6.8.5.2).

The only way to free the resources used by an event entry, even if it has already expired, is to issue either a *Delete Event* (6.8.5.5) or a *Delete All Events* (6.8.5.6) command.

6.8.1 Server Interface Data Structures

Any device that implements this service as a server role is required to store event scheduling entries.

An event entry has the information described in Table 178, organized (for messaging purposes) as shown in Table 179.

Table 178 - Information stored for each Event entry.

Field Name	Field Description	Type	Value	M/O
Event ID	Event identifier, unique per device with server role implementation.	U8	0x00 - 0xFE ³⁰	M
Event Status	Indicates the current state (disabled/enabled) of the event.	U1	0x00 - Disabled 0x01 - Enabled	M
-	Reserved for future use.	U7	0x00	M
Start Date	Date of the first invocation of the batch program with identifier <i>Program ID</i> . This date follows the format specified by the Time Service’s interface (see 6.12)	U32	0x00000000 - 0xFFFFFFFF	M
End Date	Date after which the batch program with identifier <i>Program ID</i> will stop being invoked. This date follows the format specified by the Time Service’s interface (see 6.12)	U32	0x00000000 - 0xFFFFFFFF	M
Repeat Interval	Time interval (specified in number of seconds) that defines the periodicity at which the invocation of the batch program with identifier <i>Program ID</i> should be repeated, after its first invocation at <i>Start Date</i> .	U32	0x00000000 - 0xFFFFFFFF	M
Program ID	Identifier of the batch program to be invoked (see 6.7 for details on batch programs)	U8	0x00 - 0xFE	M

²⁹ Unique only at the device with server role implementation level that is, an *Event ID* is not unique across different devices implementing this interface’s server role.

³⁰ The possible values can be further restricted by the *Maximum Number of Entries* server attribute (see 6.8.2.1).

Table 179 - Byte order and organization of an Event Entry.

8	7	6	5	4	3	2	1	Octet
Event ID								1
Event Status	Reserved							
Start Date (MSB)								3
Start Date								4
Start Date								5
Start Date (LSB)								6
End Date (MSB)								7
End Date								8
End Date								9
End Date (LSB)								10
Repeat Interval (MSB)								11
Repeat Interval								12
Repeat Interval								13
Repeat Interval (LSB)								14
Program ID								15

6.8.2 Server Attributes

Table 180 - Event Scheduling Interface: Server Attributes.

Attribute ID	Attribute Name	Attribute Type	Attribute Values	Attribute Access	M/O
0x01	Maximum Number of Entries	U8	0x00 - 0xFF	Read Only	M
0x02	Number of Entries	U8	0x00 - 0xFF	Read Only	M
0x03	Scheduler Status	U8	0x00 - Disabled 0x01 - Enabled	Read Only	M

6.8.2.1 Maximum Number of Entries

Maximum Number of Entries attribute specifies the maximum number of events that can be scheduled.

6.8.2.2 Number of Entries

Number of Entries attribute records the number of scheduled events on any given time.

6.8.2.3 Scheduler Status

Scheduler Status attribute indicates if the entire scheduler is enabled or disabled. When disabled none of the scheduled events will be processed. The default scheduler status is enabled.

6.8.3 Client Attributes

None.

6.8.4 Server to Client Commands

None.

6.8.5 Client to Server Commands

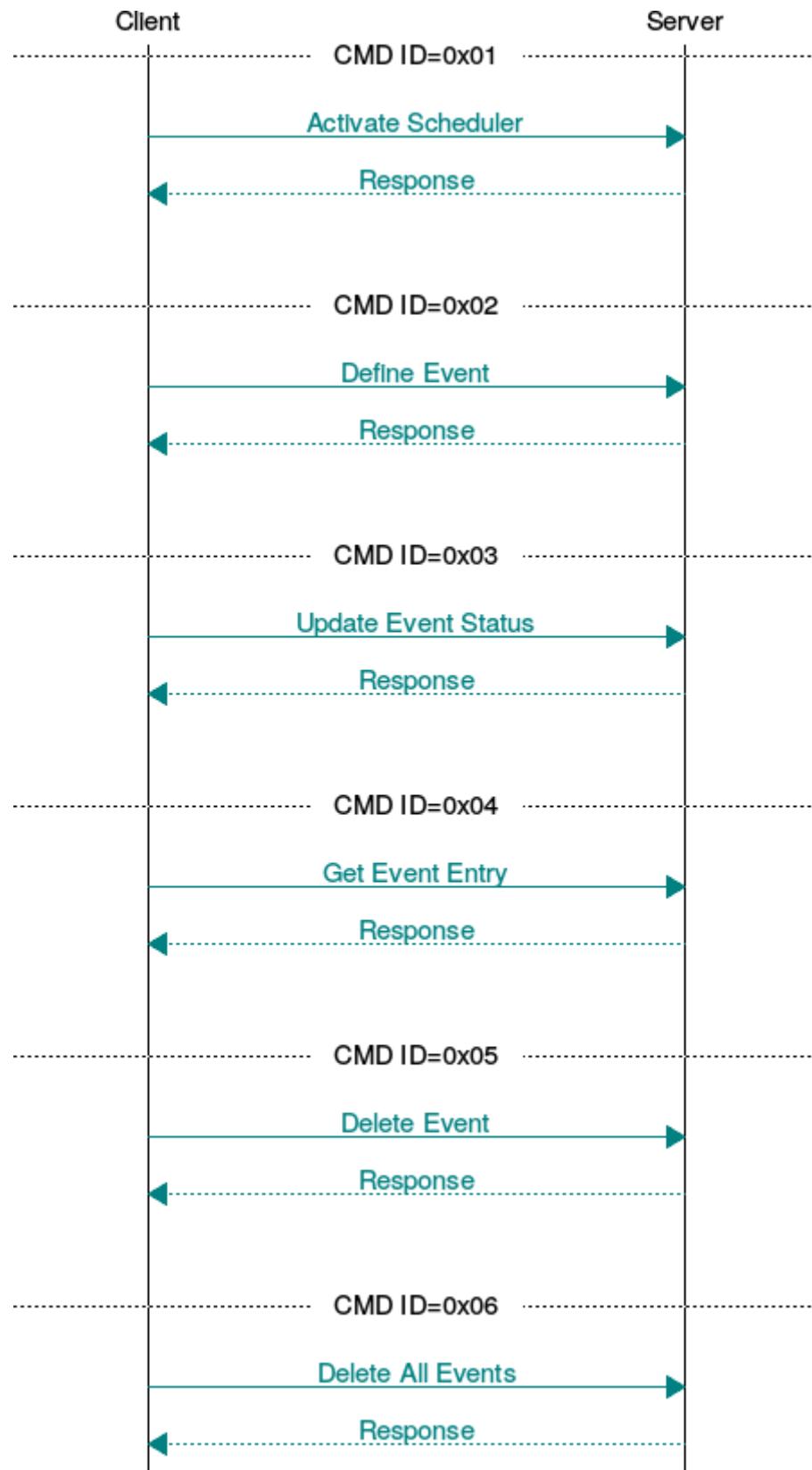
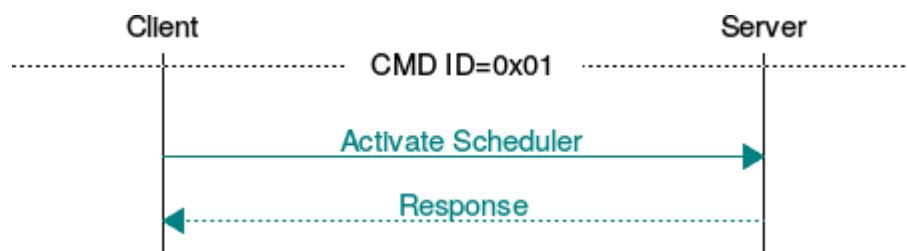


Figure 52 - Event Scheduling Interface: Client commands.

Table 181 - Implementation status of Event Scheduling Interface Client commands.

Command	Reference	Client Role	Server Role	Response
Activate Scheduler	6.8.5.1	M	M	M
Define Event	6.8.5.2	M	M	M
Update Event Status	6.8.5.3	O	M	M
Get Event Entry	6.8.5.4	O	M	M
Delete Event	6.8.5.5	M	M	M
Delete All Events	6.8.5.6	O	M	M

6.8.5.1 Activate Scheduler

**Figure 53 - Event Scheduling Interface: *Activate Scheduler* command.**

This command allows for the entire scheduler to be disabled or enabled, providing the means to inhibit the processing of all scheduled events with a single command.

The command must carry a single byte with the information described in Table 182.

The response is a single byte – Response Code – with one of the values described in Table 184.

Table 182 - Data in the payload of an *Activate Scheduler* command.

Field Name	Field Description	Type	Value	M/O
Scheduler Status	Operational state in which to put the scheduler into.	U8	0x00 - Disabled 0x01 - Enabled	M

Table 183 - Data ordering of the payload of a Response to an *Activate Scheduler* command.

8	7	6	5	4	3	2	1	Octet
Scheduler Status								1

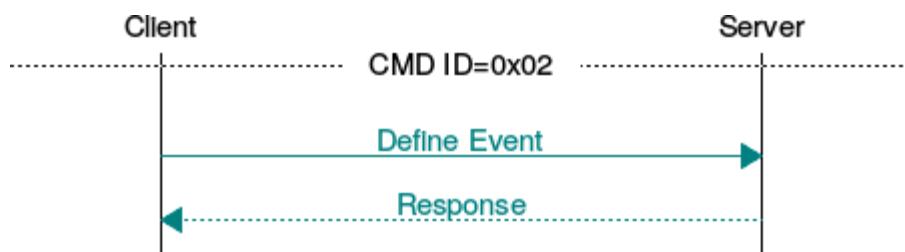
Table 184 - Data in the payload of a Response to an *Activate Scheduler* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0xFF - Fail: Unknown reason	M

Table 185 - Data ordering of the payload of a Response to an *Activate Scheduler* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1

6.8.5.2 Define Event

**Figure 54 - Event Scheduling Interface: *Define Event* command.**

This command allows an event to be defined. If the Scheduler is enabled (see 6.8.2.3 and 6.8.5.1) the defined event will invoke a specific batch program (see 6.7) at *Start Time*, repeat it at each *Repeat Interval* until *End Time* is reached.

When the special *Event ID* value – 0xFF – is used, the event being defined will be stored on the next available identifier.

The command must provide the information described in Table 186, organized according to Table 187.

The response must provide the information described in Table 188, organized according to Table 189. If the Response Code indicates a failure then no other fields are required to be present and the response has only 1 byte in total.

Table 186 - Data in the payload of a *Define Event* command.

Field Name	Field Description	Type	Value	M/O
Event ID	Event identifier, unique per device with server role implementation.	U8	0x00 - 0xFE	M
Event Status	Indicates the current state (disabled/enabled) of the event.	U1	0x00 - Disabled 0x01 - Enabled	M
-	Reserved for future use.	U7	0x00	M
Start Date	Date of the first invocation of the batch program with identifier <i>Program ID</i> . This date follows the format specified by the Time Service's interface (see 6.12)	U32	0x00000000 - 0xFFFFFFFF	M
End Date	Date after which the batch program with identifier <i>Program ID</i> will stop being invoked. This date follows the format specified by the Time Service's interface (see 6.12)	U32	0x00000000 - 0xFFFFFFFF	M
Repeat Interval	Time interval (specified in number of seconds) that defines the periodicity at which the invocation of the batch program with identifier <i>Program ID</i> should be repeated, after its first invocation at <i>Start Date</i> .	U32	0x00000000 - 0xFFFFFFFF	M
Program ID	Identifier of the batch program to be invoked (see 6.7 for details on batch programs)	U8	0x00 - 0xFE	M

Table 187 - Data ordering of the payload of a *Define Event* command.

8	7	6	5	4	3	2	1	Octet
Event ID								1
Event Status	Reserved							
Start Date (MSB)								3
Start Date								4
Start Date								5
Start Date (LSB)								6
End Date (MSB)								7
End Date								8
End Date								9
End Date (LSB)								10
Repeat Interval (MSB)								11
Repeat Interval								12
Repeat Interval								13
Repeat Interval (LSB)								14
Program ID								15

Table 188 - Data in the payload of a Response to a *Define Event* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0xFE - Fail: Not enough resources 0xFF - Fail: Unknown reason	M
Event ID	Event identifier, unique per device with server role implementation.	U8	0x00 - 0xFE	M

Table 189 - Data ordering of the payload of a Response to a *Define Event* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
Event ID								2

6.8.5.3 Update Event Status

**Figure 55 - Event Scheduling Interface: *Update Event Status* command.**

This command allows the state of a specific event to be modified, either enabling or disabling it.

The command must provide the information described in Table 190, organized according to Table 191.

The response must provide the information described in Table 192, organized according to Table 193. If the Response Code indicates a failure then no other fields are required to be present and the response has only 1 byte in total.

Table 190 - Data in the payload of an *Update Event Status* command.

Field Name	Field Description	Type	Value	M/O
Event ID	Event identifier, unique per device with server role implementation.	U8	0x00 - 0xFE	M
Event Status	Indicates the state (disabled/enabled) to which to set the event.	U1	0x00 - Disabled 0x01 - Enabled	M

Table 191 - Data ordering of the payload of an *Update Event Status* command.

8	7	6	5	4	3	2	1	Octet
Event ID								1
Event Status	Reserved							

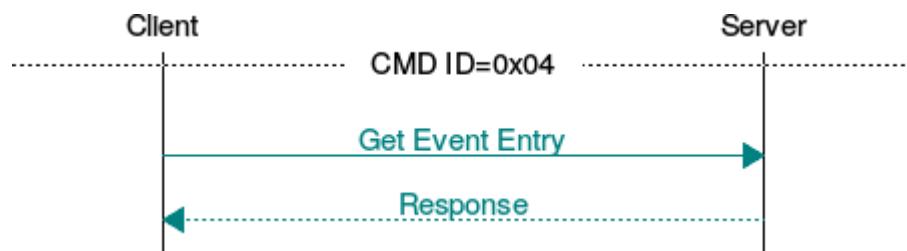
Table 192 - Data in the payload of a Response to an *Update Event Status* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0xFF - Fail: Unknown reason	M
Event ID	Event identifier, unique per device with server role implementation.	U8	0x00 - 0xFE	M

Table 193 - Data ordering of the payload of a Response to an *Update Event Status* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
Event ID								2

6.8.5.4 Get Event Entry

**Figure 56 - Event Scheduling Interface: *Get Event Entry* command.**

This command allows for a specific event entry to be obtained.

The command must carry a single byte with the information described in Table 194.

The response must provide the information described in Table 195, organized according to Table 196.

Table 194 - Data in the payload of a *Get Event Entry* command.

Field Name	Field Description	Type	Value	M/O
Event ID	Event identifier, unique per device with server role implementation.	U8	0x00 - 0xFE	M

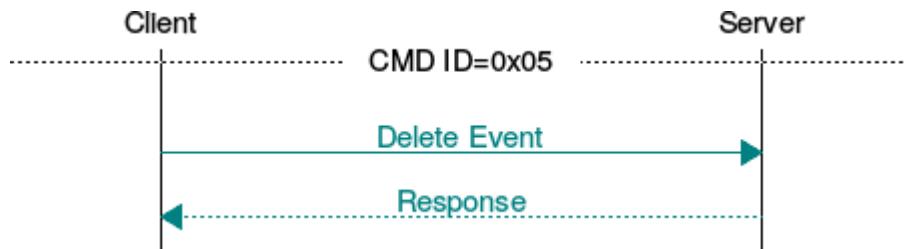
Table 195 - Data in the payload of a Response to a *Get Event Entry* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0xFE - Fail: Not enough resources 0xFF - Fail: Unknown reason	M
Event Entry	See Table 178 specified in section 6.8.1.	-	-	M

Table 196 - Data ordering of the payload of a response to *Get Event Entry* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
Event Entry (See Table 179 from section 6.8.1)								2

6.8.5.5 Delete Event

**Figure 57 - Event Scheduling Interface: *Delete Event* command.**

This command allows for a defined event to be deleted. The server implementation will clear any information related to the deleted event, releasing resources that were previously allocated. As a consequence the *Event ID* of the deleted event is made available for new events.

The command must carry a single byte with the information described in Table 197.

The response must provide the information described in Table 198, organized according to Table 199. If the Response Code indicates a failure then no other fields are required to be present and the response has only 1 byte in total.

Table 197 - Data in the payload of a *Delete Event* command.

Field Name	Field Description	Type	Value	M/O
Event ID	Event identifier, unique per device with server role implementation.	U8	0x00 - 0xFE	M

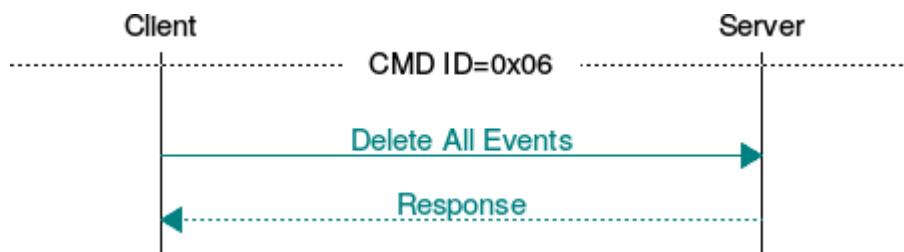
Table 198 - Data in the payload of a Response to a *Delete Event* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0xFF - Fail: Unknown reason	M
Event ID	Event identifier, unique per device with server role implementation.	U8	0x00 - 0xFE	M

Table 199 - Data ordering of the payload of a Response to a *Delete Event* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
Event ID								2

6.8.5.6 Delete All Events

**Figure 58 - Event Scheduling Interface: *Delete All Events* command.**

This command allows for all defined events to be deleted. The server implementation will clear all information related to existing events, releasing all the allocated resources.

The command has no payload, but its response has a single byte – Response Code – with one of the values described in Table 200.

Table 200 - Data in the payload of a Response to a *Delete All Programs* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0xFF - Fail: Unknown reason	M

Table 201 - Data ordering of the payload of a Response to a *Delete All Programs* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1

6.9 Weekly Scheduling Interface

This interface defines the commands and attributes to be used by a device implementing Weekly Scheduling Service.

Each weekly event entry is identifiable by a unique³¹ number called *Weekly Event ID*, which can take any value in the range 0x00 to 0xFE or the special value 0xFF. This special value signifies “any available identifier” and is only used by *Define Weekly Event* (6.9.5.2).

6.9.1 Server Interface Data Structures

Any device that implements this service as a server role is required to store weekly event scheduling entries.

A weekly event entry has the information described in Table 202, organized (for messaging purposes) as shown in Table 203.

Table 202 - Information stored for each Weekly Event entry.

Field Name	Field Description	Type	Value	M/O
Weekly Event ID	Weekly event identifier, unique per device with server role implementation.	U8	0x00 - 0xFE ³²	M
Weekly Event Status	Indicates the current state (disabled/enabled) of the weekly event.	U1	0x00 - Disabled 0x01 - Enabled	M
-	Reserved for future use.	U7	0x00	M
Day of Week	Day of the week at which the batch program with identifier <i>Program ID</i> , will be invoked	U8	0x00 - Monday 0x01 - Tuesday 0x02 - Wednesday 0x03 - Thursday 0x04 - Friday 0x05 - Saturday 0x06 - Sunday	M
Hour of Day	Hour of the day at which the batch program with identifier <i>Program ID</i> , will be invoked	U8	0x00 - 0x17	M
Minute of Hour	Minute of the hour at which the batch program with identifier <i>Program ID</i> , will be invoked	U8	0x00 - 0x3B	M
Program ID	Identifier of the batch program to be invoked (see 6.7 for details on batch programs)	U8	0x00 - 0xFE	M

³¹ Unique only at the device with server role implementation level that is, a *Weekly Event ID* is not unique across different devices implementing this interface’s server role.

³² The possible values can be further restricted by the *Maximum Number of Entries* server attribute (see 6.9.2.1).

Table 203 - Byte order and organization of a Weekly Event Entry.

8	7	6	5	4	3	2	1	Octet
Weekly Event ID								1
Weekly Event Status	Reserved							
Day of Week								3
Hour of Day								4
Minute of Hour								5
Program ID								6

6.9.2 Server Attributes

Table 204 - Weekly Scheduling Interface: Server Attributes.

Attribute ID	Attribute Name	Attribute Type	Attribute Values	Attribute Access	M/O
0x01	Maximum Number of Entries	U8	0x00 - 0xFF	Read Only	M
0x02	Number of Entries	U8	0x00 - 0xFF	Read Only	M
0x03	Weekly Scheduler Status	U8	0x00 - Disabled 0x01 - Enabled	Read Only	M

6.9.2.1 Maximum Number of Entries

Maximum Number of Entries attribute specifies the maximum number of weekly events that can be scheduled.

6.9.2.2 Number of Entries

Number of Entries attribute records the number of weekly scheduled events on any given time.

6.9.2.3 Weekly Scheduler Status

Weekly Scheduler Status attribute indicates if the entire weekly scheduler is enabled or disabled. When disabled none of the scheduled weekly events will be processed. The default weekly scheduler status is enabled.

6.9.3 Client Attributes

None.

6.9.4 Server to Client Commands

None.

6.9.5 Client to Server Commands

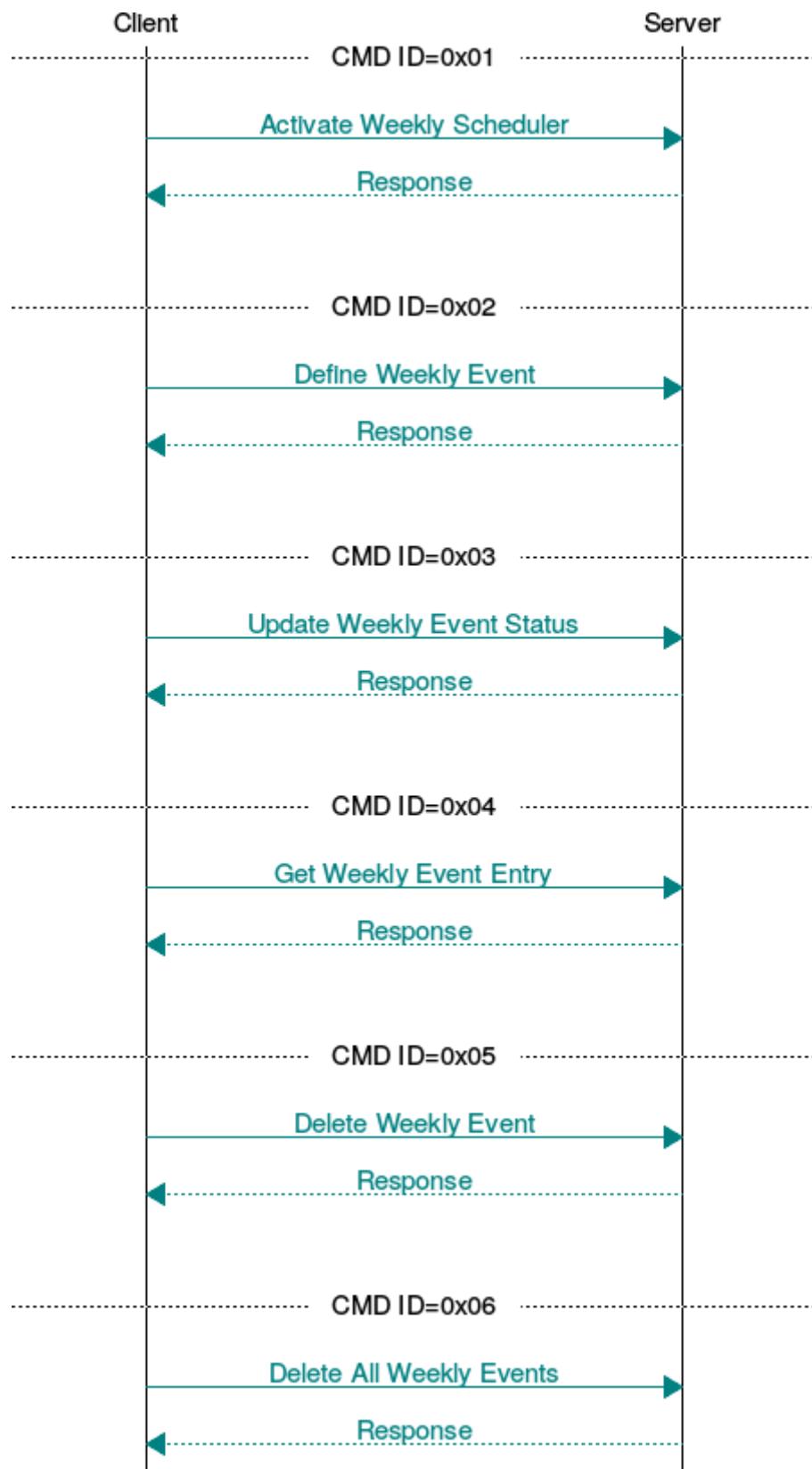
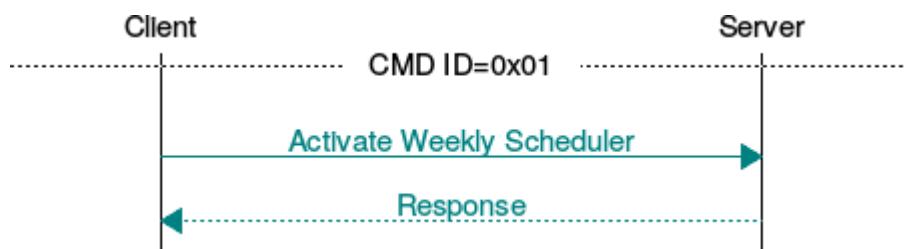


Figure 59 - Weekly Scheduling Interface: Client commands.

Table 205 - Implementation status of Weekly Scheduling Interface Client commands.

Command	Reference	Client Role	Server Role	Response
Activate Weekly Scheduler	6.9.5.1	M	M	M
Define Weekly Event	6.9.5.2	M	M	M
Update Weekly Event Status	6.9.5.3	O	M	M
Get Weekly Event Entry	6.9.5.4	O	M	M
Delete Weekly Event	6.9.5.5	M	M	M
Delete All Weekly Events	6.9.5.6	O	M	M

6.9.5.1 Activate Weekly Scheduler

**Figure 60 - Weekly Scheduling Interface: *Activate Weekly Scheduler* command.**

This command allows for the entire weekly scheduler to be disabled or enabled, providing the means to inhibit the processing of all weekly scheduled events with a single command.

The command must carry a single byte with the information described in Table 206.

The response is a single byte – Response Code – with one of the values described in Table 207.

Table 206 - Data in the payload of an *Activate Weekly Scheduler* command.

Field Name	Field Description	Type	Value	M/O
Weekly Scheduler Status	Operational state in which to put the weekly scheduler into.	U8	0x00 - Disabled 0x01 - Enabled	M

Table 207 - Data in the payload of a Response to an *Activate Weekly Scheduler* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0xFF - Fail: Unknown reason	M

6.9.5.2 Define Weekly Event

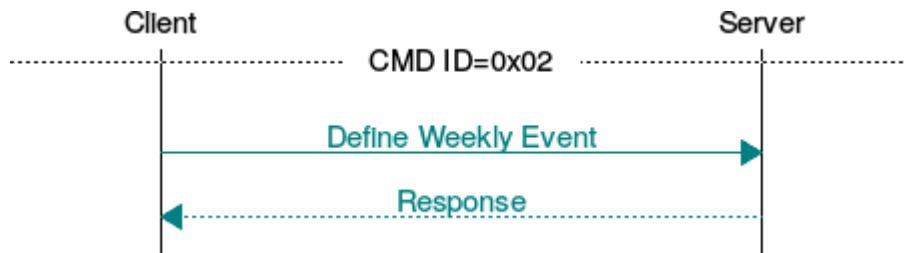


Figure 61 - Weekly Scheduling Interface: *Define Weekly Event* command.

This command allows a weekly event to be defined. If the weekly scheduler is enabled (see 6.9.2.3 and 6.9.5.1) the defined event will invoke a specific batch program (see 6.7) every *Day of Week*, at the specified *Hour of Day* and *Minute of Hour*.

When the special *Weekly Event ID* value – 0xFF – is used, the weekly event being defined will be stored on the next available identifier.

The command must provide the information described in Table 208, organized according to Table 209.

The response must provide the information described in Table 210, organized according to Table 211. If the Response Code indicates a failure then no other fields are required to be present and the response has only 1 byte in total.

Table 208 - Data in the payload of a *Define Weekly Event* command.

Field Name	Field Description	Type	Value	M/O
Event ID	Event identifier, unique per device with server role implementation.	U8	0x00 - 0xFE	M
Event Status	Indicates the current state (disabled/enabled) of the event.	U1	0x00 - Disabled 0x01 - Enabled	M
-	Reserved for future use.	U7	0x00	M
Day of Week	Day of the week at which the batch program with identifier <i>Program ID</i> , will be invoked	U8	0x00 - Monday 0x01 - Tuesday 0x02 - Wednesday 0x03 - Thursday 0x04 - Friday 0x05 - Saturday 0x06 - Sunday	M
Hour of Day	Hour of the day at which the batch program with identifier <i>Program ID</i> , will be invoked	U8	0x00 - 0x17	M
Minute of Hour	Minute of the hour at which the batch program with identifier <i>Program ID</i> , will be invoked	U8	0x00 - 0x3B	M
Program ID	Identifier of the batch program to be invoked (see 6.7 for details on batch programs)	U8	0x00 - 0xFE	M

Table 209 - Data ordering of the payload of a *Define Weekly Event* command.

8	7	6	5	4	3	2	1	Octet
Event ID								1
Event Status	Reserved							
Day of Week								3
Hour of Day								4
Minute of Hour								5
Program ID								6

Table 210 - Data in the payload of a Response to a *Define Weekly Event* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0xFE - Fail: Not enough resources 0xFF - Fail: Unknown reason	M
Weekly Event ID	Weekly event identifier, unique per device with server role implementation.	U8	0x00 - 0xFE	M

Table 211 - Data ordering of the payload of a Response to a *Define Weekly Event* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
Weekly Event ID								2

6.9.5.3 Update Weekly Event Status

**Figure 62 - Weekly Scheduling Interface: *Update Weekly Event Status* command.**

This command allows the state of a specific weekly event to be modified, either enabling or disabling it.

The command must provide the information described in Table 212, organized according to Table 213.

The response must provide the information described in Table 214, organized according to Table 215. If the Response Code indicates a failure then no other fields are required to be present and the response has only 1 byte in total.

Table 212 - Data in the payload of an *Update Weekly Event Status* command.

Field Name	Field Description	Type	Value	M/O
Weekly Event ID	Weekly event identifier, unique per device with server role implementation.	U8	0x00 - 0xFE	M
Weekly Event Status	Indicates the state (disabled/enabled) to which to set the weekly event.	U1	0x00 - Disabled 0x01 - Enabled	M

Table 213 - Data ordering of the payload of an *Update Weekly Event Status* command.

8	7	6	5	4	3	2	1	Octet
Event ID								1
Event Status	Reserved							2

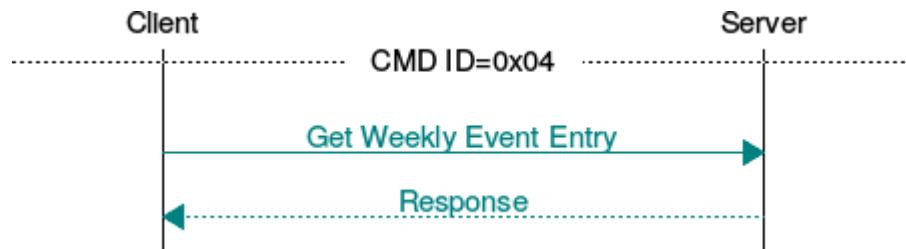
Table 214 - Data in the payload of a Response to an *Update Weekly Event Status* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0xFF - Fail: Unknown reason	M
Weekly Event ID	Weekly event identifier, unique per device with server role implementation.	U8	0x00 - 0xFE	M

Table 215 - Data ordering of the payload of a Response to an *Update Event Status* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
Weekly Event ID								2

6.9.5.4 Get Weekly Event Entry

**Figure 63 - Weekly Scheduling Interface: *Get Weekly Event Entry* command.**

This command allows for a specific weekly event entry to be obtained.

The command must carry a single byte with the information described in Table 216.

The response must provide the information described in Table 217, organized according to Table 218.

Table 216 - Data in the payload of a *Get Weekly Event Entry* command.

Field Name	Field Description	Type	Value	M/O
Weekly Event ID	Weekly event identifier, unique per device with server role implementation.	U8	0x00 - 0xFE	M

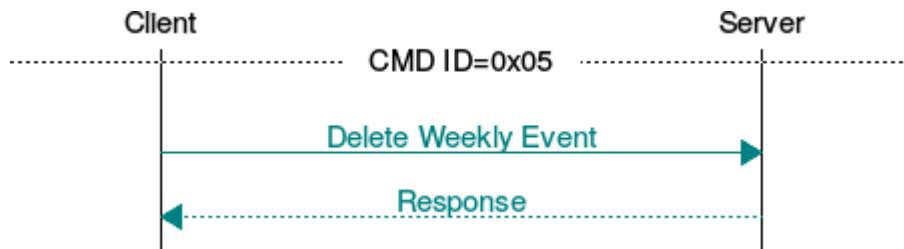
Table 217 - Data in the payload of a Response to a *Get Weekly Event Entry* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0xFE - Fail: Not enough resources 0xFF - Fail: Unknown reason	M
Weekly Event Entry	See Table 202 specified in section 6.9.1.	-	-	M

Table 218 - Data ordering of the payload of a response to *Get Weekly Event Entry* command.

8	7	6	5	4	3	2	1	Octet
								1
								2

6.9.5.5 Delete Weekly Event

**Figure 64 - Weekly Scheduling Interface: *Delete Weekly Event* command.**

This command allows for a defined weekly event to be deleted. The server implementation will clear any information related to the deleted weekly event, releasing resources that were previously allocated. As a consequence the *Weekly Event ID* of the deleted weekly event is made available for new weekly events.

The command must carry a single byte with the information described in Table 219.

The response must provide the information described in Table 221, organized according to Table 222. If the Response Code indicates a failure then no other fields are required to be present and the response has only 1 byte in total.

Table 219 - Data in the payload of a *Delete Weekly Event* command.

Field Name	Field Description	Type	Value	M/O
Weekly Event ID	Weekly event identifier, unique per device with server role implementation.	U8	0x00 - 0xFE	M

Table 220 - Data ordering of the payload of a Response to a *Delete Weekly Event* command.

8	7	6	5	4	3	2	1	Octet
Weekly Event ID								2

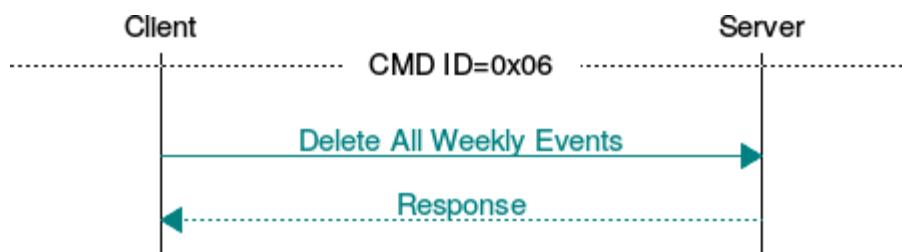
Table 221 - Data in the payload of a Response to a *Delete Weekly Event* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0xFF - Fail: Unknown reason	M
Weekly Event ID	Weekly event identifier, unique per device with server role implementation.	U8	0x00 - 0xFE	M

Table 222 - Data ordering of the payload of a Response to a *Delete Weekly Event* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
Weekly Event ID								2

6.9.5.6 Delete All Weekly Events

**Figure 65 - Weekly Scheduling Interface: *Delete All Weekly Events* command.**

This command allows for all defined weekly events to be deleted. The server implementation will clear all information related to existing weekly events, releasing all the allocated resources.

The command has no payload, but its response has a single byte – Response Code – with one of the values described in Table 223.

Table 223 - Data in the payload of a Response to a *Delete All Weekly Events* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Ok 0x01 - Fail: Not authorized 0x02 - Fail: Invalid argument 0xFF - Fail: Unknown reason	M

Table 224 - Data ordering of the payload of a Response to a *Delete All Weekly Events* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1

6.10 Group Table Interface

This interface allows adding, removing, retrieving entries from the group table maintain by the device. Group table entry is an association of a group address to a unit identifier.

A device that supports group-cast must implement the Server side of this interface in its unit 0.

The manipulation of the table MUST done by a HF Concentrator only, i.e. the client side commands of the interface coming from a device that is not a HF Concentrator will be ignored.

A HF Device can add another device to a group using the Group Management service defined in the core services document.

Please see the Multicast section of the relevant interworking document for further requirements HF devices are subject to when implementing this interface.

6.10.1 Server Attributes

Table 225 – Group Table Interface Server: Attributes.

Attribute ID	Attribute Name	Attribute Type	Attribute Values	Attribute Access	M/O
0x01	Number of entries	U8	0x00 - 0xFF	Read Only	M
0x02	Maximum Number of Entries	U8	0x01 - 0xFF	Read Only	M

6.10.2 Number of entries

This attributes hold how many entries are currently in the Group Table. This attributed is affected from Add/Remove entry commands

6.10.3 Maximum Number of Entries

This attributes hold what is the maximum entries that the device can hold in its group table.

If this interface is implemented by the device it must support at least 1 entry.

6.10.4 Client Attributes

None.

6.10.5 Server to Client Commands

None.

6.10.6 Client to Server Commands

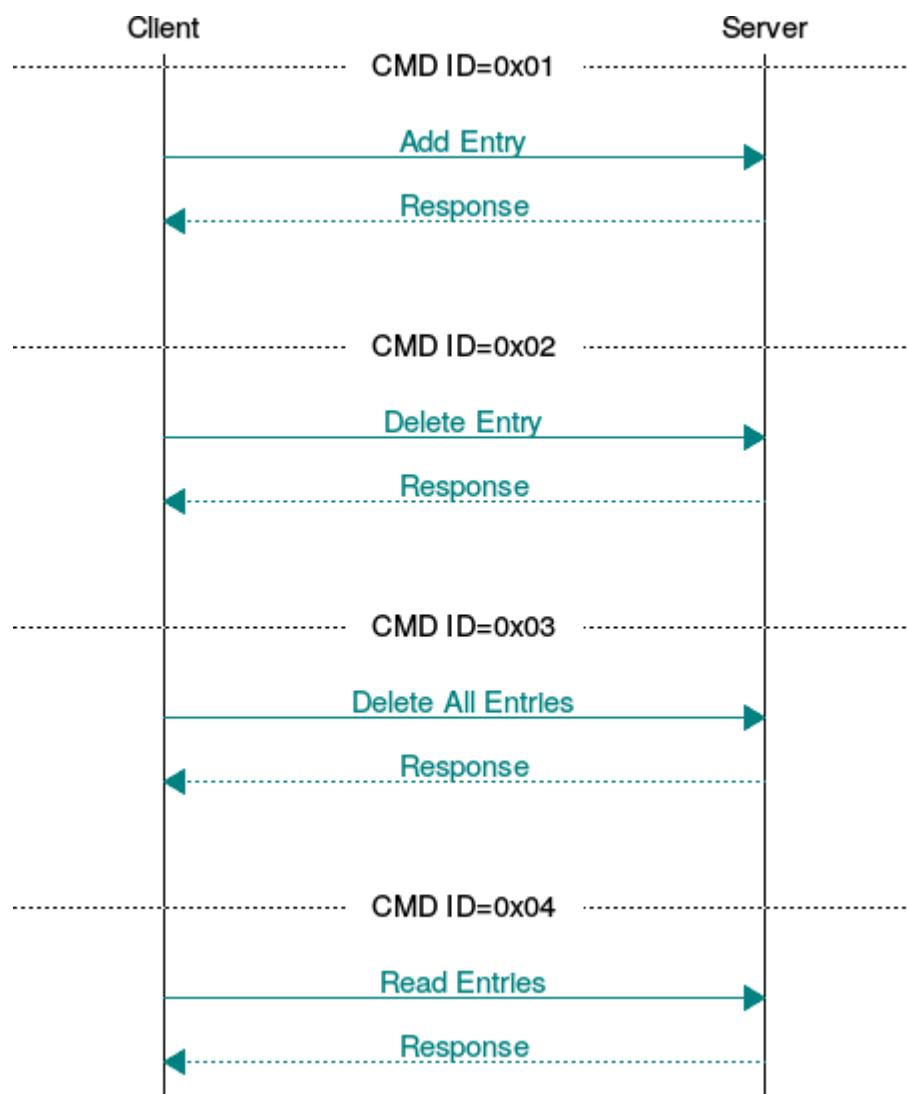


Figure 66 - Group Table Interface: Client commands.

Table 226 - Implementation status of Group Table Interface Client commands.

Command	Reference	Client Role	Server Role	Response
Add Entry	6.10.6.1	M	M	M
Delete Entry	6.10.6.2	M	M	M
Delete All Entries	6.10.6.3	M	M	M
Read Entries	6.10.6.4	M	M	M

6.10.6.1 Add Entry

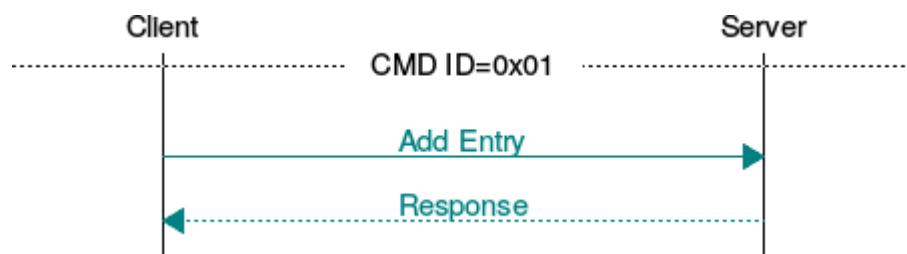


Figure 67 - Group Table Interface: *Add Entry* command.

This command is sent to a server implementing the Group Table interface in its unit 0, and adds an entry that associates a group address to one or more units within the device.

When the device receives group-cast message to a specific group address, it shall forward it to any unit associate with this group address.

Table 227 - Data in Payload of the *Add Entry* command

Field Name	Field Description	Type	Value	M/O
Group Address	The Group Address	U15	0x0001 - 0x7FFF	M
Unit ID	The Unit ID	U8	0x00 - 0xFF	M

Table 228 - Data Ordering of Payload of an *Add Entry* Command

Octet	8	7	6	5	4	3	2	1
x	Group Address (MSB)							1
	Group Address (LSB)							2
	Unit ID							3

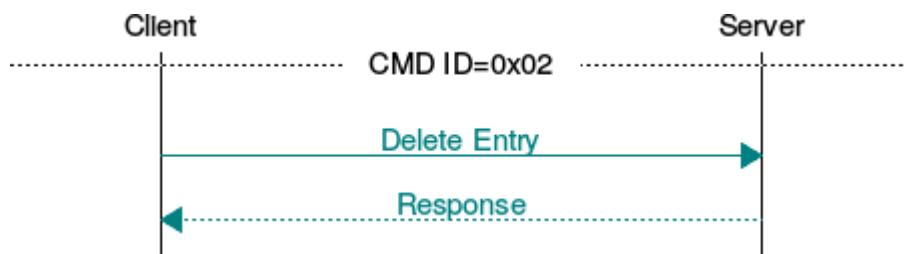
Table 229 - Data in Payload of the Response to an *Add Entry* command

Field Name	Field Description	Type	Value	M/O
Response code	Value that indicates the state of the command reception/processing	U8	One of the General response allowed in the standard (0 == OK)	M
Group Address	The Group Address	U15	0x0001 - 0x7FFF	M
Unit ID	The Unit Id	U8	0x00 - 0xFF	M

Table 230 - Data ordering of the payload of a Response to an *Add Entry* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
x	Group Address (MSB)							
Group Address (LSB)								3
Unit ID								4

6.10.6.2 Delete Entry

**Figure 68 - Group Table Interface: Delete Entry command.**

This command is sent to a server implementing the Group Table interface in its unit 0, to delete an entry (association of a group address to a unit id) within the device.

Table 231 - Data in Payload of the *Delete Entry* command

Field Name	Field Description	Type	Value	M/O
Group Address	The Group Address	U15	0x0001 - 0x7FFF	M
Unit ID	The Unit ID	U8	0x00 - 0xFF	M

Table 232 - Data Ordering of Payload of a *Delete Entry* Command

8	7	6	5	4	3	2	1	Octet
x	Group Address (MSB)							
Group Address (LSB)								2
Unit ID								3

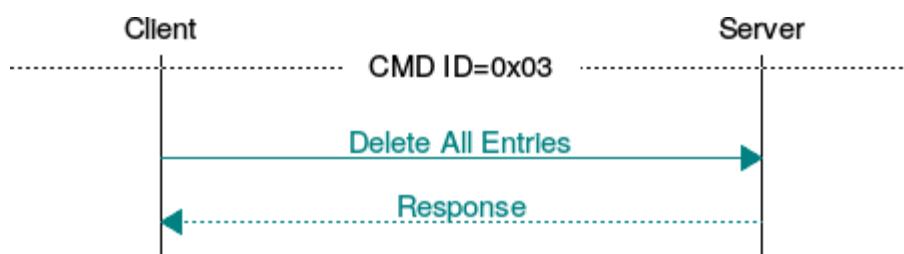
Table 233 - Data in Payload of the Response to a *Delete Entry* command

Field Name	Field Description	Type	Value	M/O
Response code	Value that indicates the state of the command reception/processing	U8	One of the General response allowed in the standard (0 == OK)	M
Group ID	The Group Id	U15	0x0001 - 0x7FFF	M
Unit ID	The Unit Id	U8	0x01 - 0xFF	M

Table 234 - Data ordering of the payload of a Response to a Delete Entry command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
x	Group Address (MSB)							
Group Address (LSB)								3
Unit ID								4

6.10.6.3 Delete All Entries

**Figure 69 - Group Table Interface: Delete All Entries command.**

This command is sent to a server implementing the Group Table interface in its unit 0, and deletes all the entries in the group table within the device.

This command had no payload.

Table 235 - Data in Payload of the Response to *Delete All Entries* command

Field Name	Field Description	Type	Value	M/O
Response code	Value that indicates the state of the command reception/processing	U8	One of the General response allowed in the standard (0 == OK)	M

Table 236 - Data Ordering in Payload of the Response to *Delete All Entries* command

8	7	6	5	4	3	2	1	Octet
Response Code								1

6.10.6.4 Read Entries

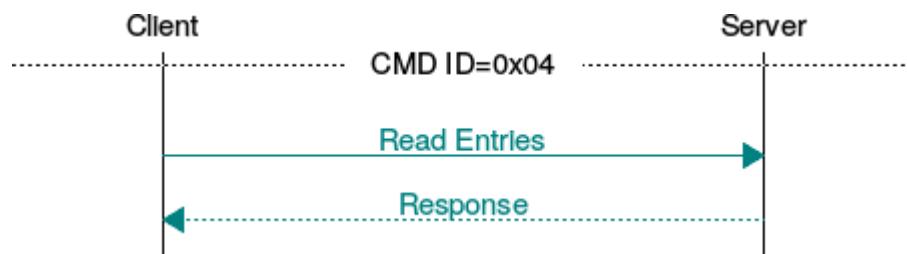


Figure 70 - Group Table Interface: Delete Entry command.

This command is sent to a server implementing the Group Table interface in its unit 0, and retrieve one or more entries from the group table.

Table 237 - Payload of the *Read Entries* command

Field Name	Field Description	Type	Value	M/O
Start Index	Index (0 based) of first entry in the table	U8	0x00 – 0xFF	M
Number of Entries	Number of entries	U8	0x01 – 0xFF	M

Table 238 - Data Ordering of Payload of *Read Entries* Command

8	7	6	5	4	3	2	1	Octet
Start Index								1
Number of entries								2

Table 239 - Data in Payload of the Response to *Read Entries* command

Field Name	Field Description	Type	Value	M/O
Response code	Value that indicates the state of the command reception/processing	U8	One of the General response allowed in the standard (0 == OK)	M
Start Index	Index (0 based) of first entry in the table	U8	0x00 - 0xFF	M
Number of Entries	Number of entries actually returned	U8	0x01 - 0xFF	M
Entries	Each entry is group address and unit id.	U15+U8		M

Table 240 – Data Ordering of Payload of *Read Entries* Response Command

8	7	6	5	4	3	2	1	Octet							
Response Code								1							
Start Index								2							
Number of entries								3							
x	Group Address (MSB)							4							
Group Address (LSB)								5							
Unit ID								6							
...								...							
x	Group Address (MSB)							K							
Group Address (LSB)								K+1							
Unit ID								K+2							

6.11 Tamper Interface

This interface defines the commands and attributes to be used by a device implementing Tampering Alert Service.

6.11.1 Server Attributes

Table 241 - Tamper Interface Server: Attributes.

Attribute ID	Attribute Name	Attribute Type	Attribute Values	Attribute Access	M/O
0x01	State	U8	0x00 - 0x01	Read Only	M

6.11.1.1 State

State attribute indicates the current state of the tampering detection. It can take one of two values:

- the value 0x00 indicates tampering is not being detected.
- the value 0x01 indicates tampering is currently being detected.

6.11.2 Client Attributes

None.

6.11.3 Server to Client Commands

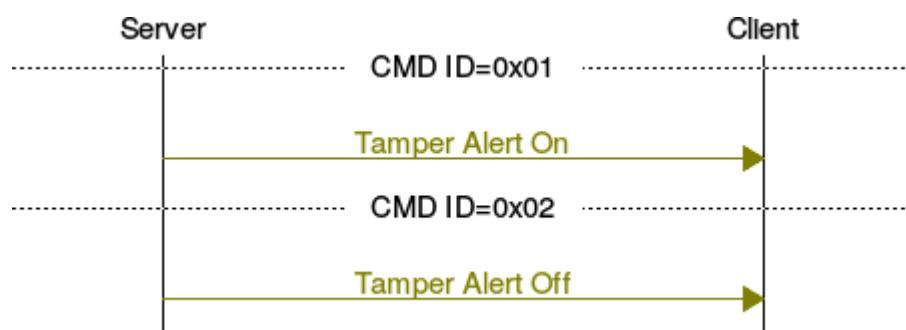


Figure 71 - Tamper Interface: Server Commands.

Table 242 - Implementation status of Tamper Interface Server commands.

Command	Reference	Client Role	Server Role	Response
Tamper Alert On	6.11.3.1	M	M	N/A
Tamper Alert Off	6.11.3.2	M	M	N/A

6.11.3.1 Tamper Alert On

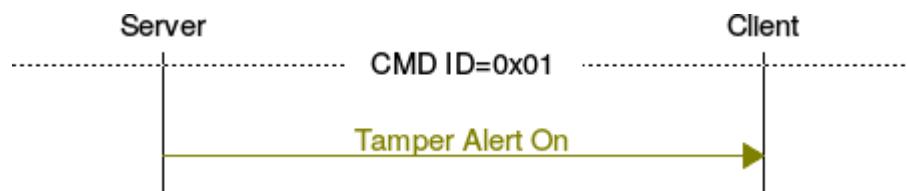


Figure 72 - Tamper Interface: *Tamper Alert On* command.

Command used to notify that tampering is being detected. It should be sent as soon has tampering is detected.

This command has no payload.

6.11.3.2 Tamper Alert Off

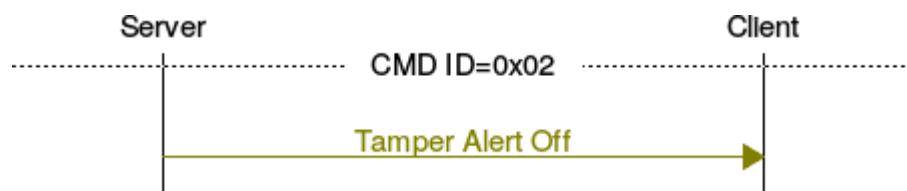


Figure 73 - Tamper Interface: *Tamper Alert Off* command.

Command used to notify that tampering stopped being detected.

This command has no payload.

6.11.4 Client to Server Commands

None.

6.12 Time Interface

This interface defines the commands and attributes to be used by a device implementing Time Service.

6.12.1 Server Attributes

Table 243 - Time Interface Server: Attributes.

Attribute ID	Attribute Name	Attribute Type	Attribute Values	Attribute Access	M/O
0x01	Time	U32	0x00000000 - 0xFFFFFFFF	Read / Write	M

6.12.1.1 Time

Time attribute stores the number of seconds of time since 0 hours, 0 minutes, 0 seconds, on the 1st of January, 2000 UTC. The special value 0xFFFFFFFF indicates an invalid time.

6.12.2 Client Attributes

None.

6.12.3 Server to Client Commands

None.

6.12.4 Client to Server Commands

None.

6.13 Power Interface

This interface defines the commands and attributes to be used by a device implementing Power Service.

6.13.1 Server Attributes

Table 244 - Power Interface: Attributes.

Attribute ID	Attribute Name	Attribute Type		Attribute Values	Attribute Access	M/O
0x01	Mains Voltage	U16		0x0000 - 0xFFFF	Read Only	O
0x02	Mains Frequency	U8		0x00 - 0xFF	Read Only	O
0x03	Mains Alarms	U8 (bitmask)		0x00 - 0xFF	Read / Write	O
0x04	Mains Minimum Voltage Threshold	U16		0x0000 - 0xFFFF	Read / Write	O
0x05	Mains Maximum Voltage Threshold	U16		0x0000 - 0xFFFF	Read / Write	O
0x06	Mains Voltage Dwell Trip Point	U16		0x0000 - 0xFFFF	Read / Write	O
0x07	Battery Voltage	U8		0x00 - 0xFF	Read Only	O
0x08	Battery Manufacturer	Length	U8	0x00 - 0x0F	Read / Write	O
		Character String	U8	0x20 - 0xFE (each U8)		
0x09	Battery Type	U8		0x00 - 0xFF	Read / Write	O
0x0A	Battery AHr Rating	U16		0x0000 - 0xFFFF	Read / Write	O
0x0B	Battery Cell Quantity	U8		0x00 - 0xFF	Read / Write	O
0x0C	Battery Rated Voltage	U8		0x00 - 0xFF	Read / Write	O
0x0D	Battery Status	U8		0x00 - 0x01	Read Only	O
0x0E	Battery Minimum Voltage Threshold	U8		0x00 - 0xFF	Read / Write	O
0x0F	Battery Maximum Voltage Threshold	U8		0x00 - 0xFF	Read / Write	O
0x10	Battery Charge Level	U8		0x00 - 0xFF	Read Only	O
0x11	Battery Minimum Charge Level	U8		0x00 - 0xFF	Read / Write	O
0x12	Battery Maximum Charge Level	U8		0x00 - 0xFF	Read / Write	O
0x13	Power Source	U8		0x00 - 0xFF	Read Only	O

6.13.1.1 Mains Voltage

Mains Voltage attribute stores the currently measured Root Mean Square (RMS) Voltage value if mains is AC, or the DC Voltage value if mains is DC. Either measurement is stored as units of 100mV.

6.13.1.2 Mains Frequency

Mains Frequency attribute stores the measured frequency divided by two, with a maximum resolution of 2 Hz, allowing the measurement of frequencies from 2 Hz up to 506 Hz. It may store special values as follows:

- 0x00, indicates frequency is too low to be measured or is a DC power supply;
- 0x01 - 0xFD, range of valid frequency values;
- 0xFE, indicates frequency is too high to be measured;
- 0xFF, indicates frequency could not be measured.

6.13.1.3 Mains Alarms

Mains Alarms attribute allows alarms to be enabled, triggered by some characteristic related with mains voltage. This value is a bitmask with bits having the following meaning:

- Bit 0 (least significant bit), if set to “1” an alarm will be generated if mains voltage is too low;
- Bit 1, if set to “1” an alarm will be generated if mains voltage is too high;
- Bits 2-7 are reserved for future usage.

6.13.1.4 Mains Minimum Voltage Threshold

Mains Minimum Voltage Threshold attribute specifies a value, in units of 100mV, to trigger an alarm if *Mains Voltage* is less than it. Such value must be less than *Mains Maximum Voltage Threshold*. The special value 0x0000 can be used to suppress the generation of alarms. An application implementing this attribute should operate in the following manner; When *Mains Voltage* goes below *Mains Minimum Voltage Threshold* a timer with *Mains Voltage Dwell Trip Point* duration (in seconds) must be started, if after the timer expires the violation still holds true then an alarm is generated, otherwise it is not.

6.13.1.5 Mains Maximum Voltage Threshold

Mains Maximum Voltage Threshold attribute specifies a value, in units of 100mV, to trigger an alarm if *Mains Voltage* is greater than it. Such value must be greater than *Mains Minimum Voltage Threshold*. The special value 0xFFFF can be used to suppress the generation of alarms. An application implementing this attribute should operate in the following manner; When *Mains Voltage* goes above *Mains Maximum Voltage Threshold* a timer with *Mains Voltage Dwell Trip Point* duration (in seconds) must be started, if after the timer expires the violation still holds true then an alarm is generated, otherwise it is not.

6.13.1.6 Mains Voltage Dwell Trip Point

Mains Voltage Dwell Trip Point attribute specifies duration in seconds, during which *Mains Voltage* may violate some specified thresholds before an alarm being generated. The special value 0xFFFF disables any alarm generation.

6.13.1.7 Battery Voltage

Battery Voltage attribute stores the currently measured battery voltage, in units of 100mV. It may store special values as follows:

- 0x00 - 0xFE range of valid voltage values;
- 0xFF indicates an invalid voltage value.

6.13.1.8 Battery Manufacturer

Battery Manufacturer attribute stores the battery’s manufacturer name with a maximum of 16 characters.

6.13.1.9 Battery Type

Battery Type attribute indicates the battery's type, restricted to the specified enumeration of values:

- 0x00 - Not present;
- 0x01 - Built-in;
- 0x02 - Other;
- 0x03 - AA;
- 0x04 - AAA;
- 0x05 - C;
- 0x06 - D;
- 0x07 - CR2;
- 0x08 - CR123A
- 0x09 - CR2032;
- 0x0A - CR2450;
- 0x0B - ER14250;
- 0x0C - 0xFE range reserved for future features;
- 0xFF - Unknown battery type.

6.13.1.10 Battery AHr Rating

Battery AHr Rating attribute stores the measured battery's ampere-hour rating, in units of 10mAHR.

6.13.1.11 Battery Cell Quantity

Battery Cell Quantity attribute indicates the number of battery cells currently on the device.

6.13.1.12 Battery Rated Voltage

Battery Rated Voltage attribute indicates, in units of 100mV, the manufacturer rated battery voltage.

6.13.1.13 Battery Status

Battery Status attribute indicates if battery is low or not, according to the following definitions:

- 0x00 indicates battery is ok;
- 0x01 indicates battery is low.

6.13.1.14 Battery Minimum Voltage Threshold

Battery Minimum Voltage Threshold attribute stores the low voltage notification threshold, specified in units of 100mV. If *Battery Voltage* is less than this threshold a notification must be generated.

6.13.1.15 Battery Maximum Voltage Threshold

Battery Maximum Voltage Threshold attribute stores the high voltage notification threshold, specified in units of 100mV. If *Battery Voltage* is higher than this threshold a notification must be generated.

6.13.1.16 Battery Charge Level

Battery Charge Level attribute indicates the battery's current charge level, as a percentage of full charge.

6.13.1.17 Battery Minimum Charge Level

Battery Minimum Charge Level attribute stores the low charge level notification threshold, specified as a percentage of full charge. If *Battery Charge Level* is less than this threshold a notification must be generated.

6.13.1.18 Battery Maximum Charge Level

Battery Maximum Charge Level attribute stores the high charge level notification threshold, specified as a percentage of full charge. If *Battery Charge Level* is higher than this threshold a notification must be generated.

6.13.1.19 Power Source

Power Source attribute stores information regarding the source(s) of power available on the device. Information is stored in nibble format according to **Table 245**, each nibble can have one of the values in **Table 246**.

Table 245 - Power Interface: Power Source information organization

8	7	6	5	4	3	2	1	Octet
Secondary Power Source				Primary Power Source				1

Table 246 - Power Interface: Power Source Codes.

Power Source Code	Description
0x00	Non existent
0x01	Battery
0x02	DC Source
0x03 - 0x7F	Reserved

6.13.2 Client Attributes

None.

6.13.3 Server to Client Commands

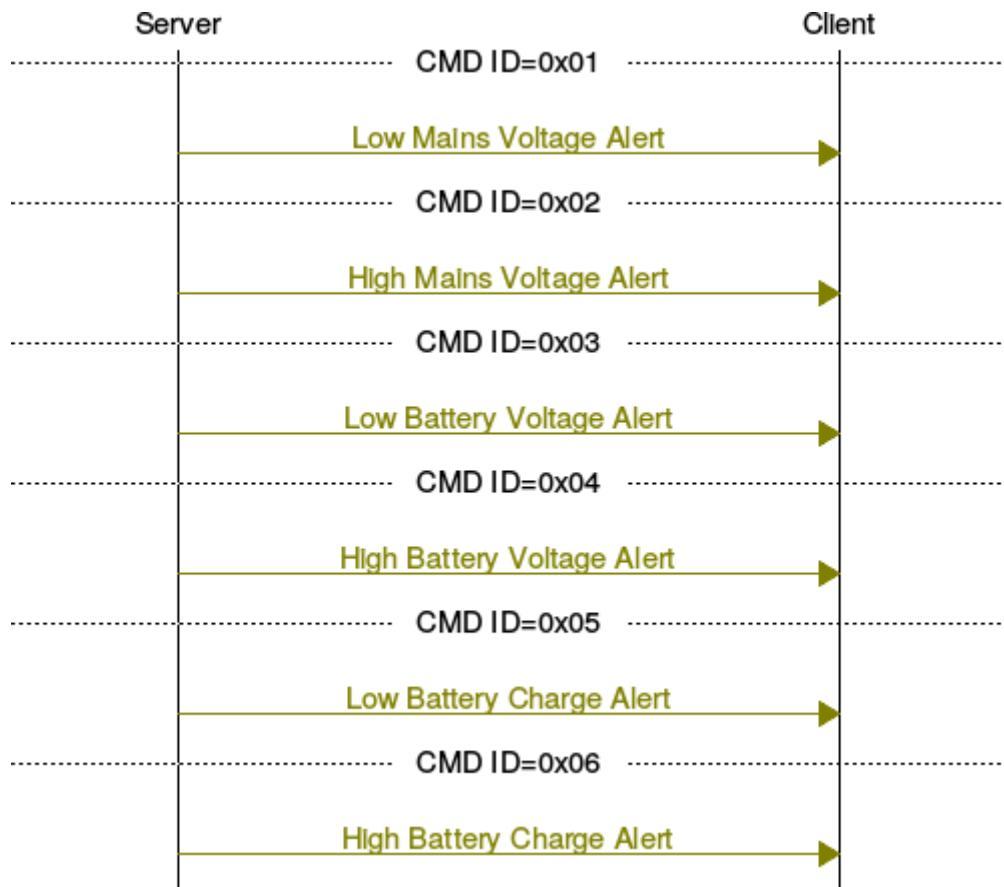


Figure 74 - Power Interface: Server Commands.

Table 247 - Implementation status of Power Interface Server commands.

Command	Reference	Client Role	Server Role	Response
Low Mains Voltage Alert	6.13.3.1	O	O	N/A
High Mains Voltage Alert	6.13.3.2	O	O	N/A
Low Battery Voltage Alert	6.13.3.3	O	O	N/A
High Battery Voltage Alert	6.13.3.4	O	O	N/A
Low Battery Charge Alert	6.13.3.5	O	O	N/A
High Battery Charge Alert	6.13.3.6	O	O	N/A

6.13.3.1 Low Mains Voltage Alert

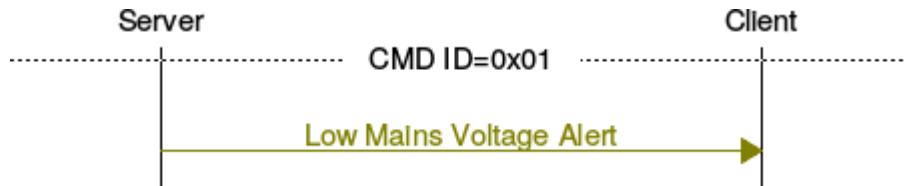


Figure 75 - Power Interface: *Low Mains Voltage Alert* command.

This command sent to a client implementation of the Power interface, indicates it has been detected *Mains Voltage* to be below *Mains Minimum Voltage Threshold*. Implementing this command is optional.

This command has no payload.

6.13.3.2 High Mains Voltage Alert

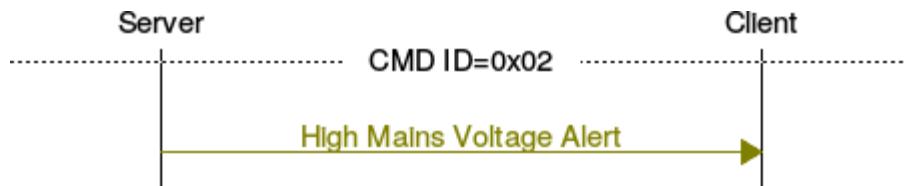


Figure 76 - Power Interface: *High Mains Voltage Alert* command.

This command sent to a client implementation of the Power interface, indicates it has been detected *Mains Voltage* to be above *Mains Maximum Voltage Threshold*. Implementing this command is optional.

This command has no payload.

6.13.3.3 Low Battery Voltage Alert

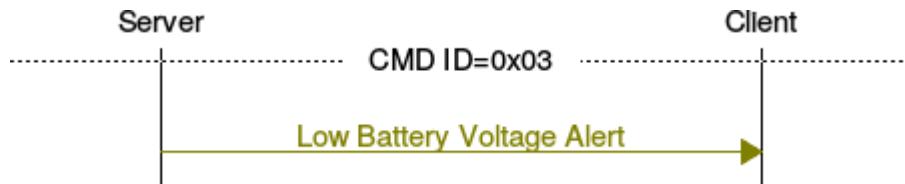


Figure 77 - Power Interface: *Low Battery Voltage Alert* command.

This command sent to a client implementation of the Power interface, indicates it has been detected *Battery Voltage* to be below *Battery Minimum Voltage Threshold*. Implementing this command is optional.

This command has no payload.

6.13.3.4 High Battery Voltage Alert

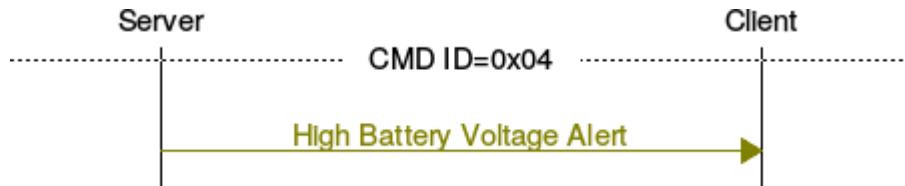


Figure 78 - Power Interface: High Battery Voltage Alert command.

This command sent to a client implementation of the Power interface, indicates it has been detected *Battery Voltage* to be above *Battery Maximum Voltage Threshold*. Implementing this command is optional.

This command has no payload.

6.13.3.5 Low Battery Charge Alert

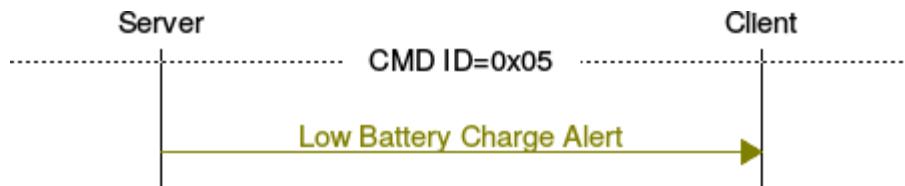


Figure 79 - Power Interface: Low Battery Charge Alert command.

This command sent to a client implementation of the Power interface, indicates it has been detected *Battery Charge Level* to be below *Battery Minimum Charge Level*. Implementing this command is optional.

This command has no payload.

6.13.3.6 High Battery Charge Alert

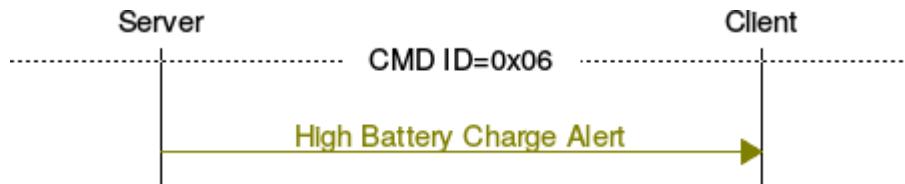


Figure 80 - Power Interface: High Battery Charge Alert command.

This command sent to a client implementation of the Power interface, indicates it has been detected *Battery Charge Level* to be above *Battery Maximum Charge Level*. Implementing this command is optional.

This command has no payload.

6.13.4 Client to Server Commands

None.

6.14 RSSI Interface

This interface defines the commands and attributes to be used by a device implementing RSSI Service.

6.14.1 Server Attributes

Table 248 - Simple RSSI Interface Server, Attributes

Attribute ID	Attribute Name	Attribute Type	Attribute Values	Attribute Access	M/O
0x01	RSSI	S8	-128 to +127	Read	M

6.14.1.1 RSSI

This attributes holds the RSSI value in dBm.

6.14.2 Client Attributes

None.

6.14.3 Server to Client Commands

None.

6.14.4 Client to Server Commands

None.

6.15 Keep Alive Interface

This interface defines the commands and attributes to be used by a device implementing Keep Alive Service.

6.15.1 Server Attributes

Table 249 - Keep Alive Interface Server: Attributes.

Attribute ID	Attribute Name	Attribute Type	Attribute Values	Attribute Access	M/O
0x01	Interval	U32	0x00000000 - 0xFFFFFFFF	Read / Write	M

6.15.1.1 Interval

Interval attribute stores the number of milliseconds of time that should trigger the sending of the *I am Alive* command. This value can be set, from a minimum of zero, which will disable the sending of the *I am Alive* command, to a maximum of a little over 49 days.

6.15.2 Client Attributes

None.

6.15.3 Server to Client Commands

Table 250 - Implementation status of Keep Alive Interface Server commands.

Command	Reference	Client Role	Server Role	Response
I am Alive	6.15.3.1	M	M	N/A

6.15.3.1 I am Alive

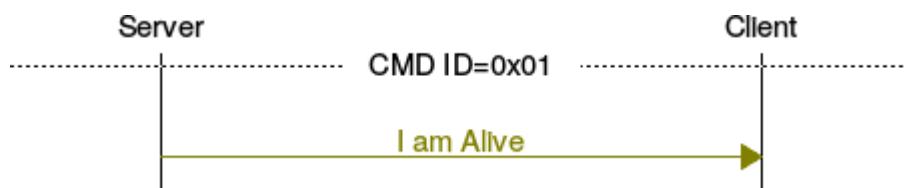


Figure 81 - Keep Alive Interface: *I am Alive* command.

This command sent to a client implementation of the Keep Alive interface, informs that the device is alive and running.

This command has no payload.

6.15.4 Client to Server Commands

None.

6.16 SUOTA Interface

This interface defines the commands and attributes to be used by a device implementing SUOTA Service.

Any device implementing this interface with the Client role must implement three optional attributes from Device Information (6.5) interface, namely *Application version*, *Hardware version* and *Manufacturer Name* attributes as indicated in Table 95.

After a successful update procedure, the device should re-send a Device Management Interface Register Device (6.1.5.1) command to update the HF concentrator of any new functionality the device may now implement.

6.16.1 Server Attributes

None.

6.16.2 Client Attributes

None.

6.16.3 Server to Client Commands

Table 251 - Implementation status of SUOTA Interface Server commands.

Command	Reference	Client Role	Server Role	Response
New Version Available	6.16.3.1	M	M	M

6.16.3.1 New Version Available

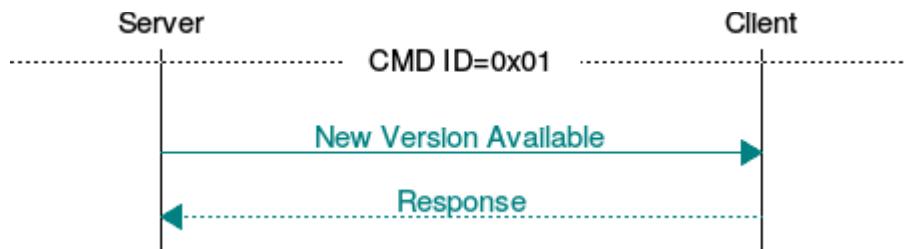


Figure 82 - SUOTA Interface: New Version Available command.

This command, sent to one or several client implementations of SUOTA interface, notifies them about the existence of new software. The client is required to process the notification and send a response indicating if an update procedure will be initiated or not. If such update is initiated it must be according to the procedure detailed in the Interworking document.

The command must provide the information described in Table 252, organized according to Table 253.

The response is a single byte – Response Code – with one of the values described in Table 254.

Table 252 – Data in the payload of a *New Version Available* command.

Field Name	Field Description	Type	Value	M/O
Software version	The new software version.	Length	U8	0x00 - 0x20
		Character String	U8	0x20 - 0xFE (each U8)
Hardware version	The minimum hardware version supported by the new software version.	Length	U8	0x00 - 0x20
		Character String	U8	0x20 - 0xFE (each U8)
URL	The URL from where the new software version might be obtained.	Length	U8	0x00 - 0xFF
		Character String	U8	0x20 - 0xFE (each U8)

Table 253 – Data ordering of the payload of a *New Version Available* command.

Octet
8
7
6
5
4
3
2
1
Octet
Software version - Length
1
Software version - First character
2
...
3 + n
Software version - Last character
4 + n
Hardware version - Length
5 + n
Hardware version - First character
...
6 + n
Hardware version - Last character
7 + n
URL - Length
8 + n
URL - First character
...
9 + n ³³
URL - Last character

Table 254 – Data in the payload of a Response to a *New Version Available* command.

Field Name	Field Description	Type	Value	M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - Upgrade initiated 0x11 - Invalid software 0x12 - Unsupported hardware 0x13 - Battery too low 0xFF - Fail: Unknown Error	M

Table 255 – Data ordering of the payload of a Response to a *New Version Available* command.

Octet
8
7
6
5
4
3
2
1
Octet
Response Code
1

³³ The value **n** represents an accumulation of an arbitrary number of bytes.

6.16.4 Client to Server Commands

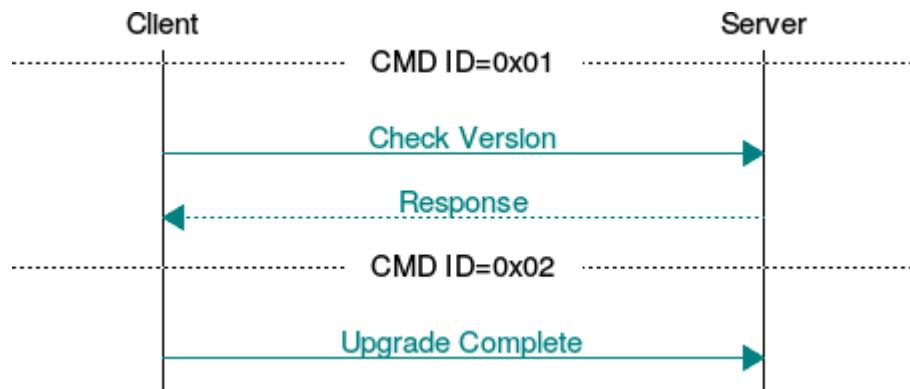


Figure 83 - SUOTA Interface, Client commands.

Table 256 - Implementation status of SUOTA Interface Server commands.

Command	Reference	Client Role	Server Role	Response
Check Version	6.16.4.1	O	M	M
Upgrade Complete	6.16.4.2	M	M	N/A

6.16.4.1 Check Version

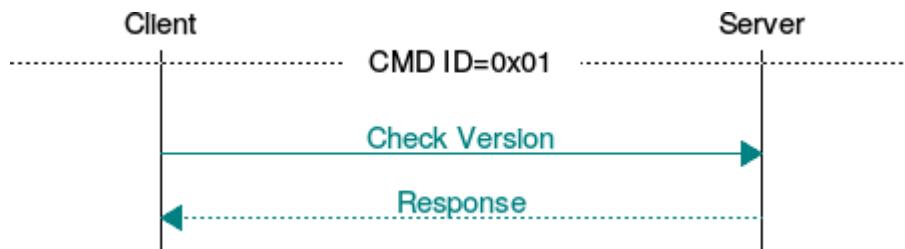


Figure 84 - SUOTA Interface: *Check Version* command.

This command, sent to a server implementation of SUOTA interface, checks the existence of new software. The client is required to send its current versions and the server will indicate if there is any newer version. The client can then decide if it will initiate an update or not. If such update is initiated it must be according to the procedure detailed in the Interworking document.

The command must provide the information described in Table 257, organized according to Table 258.

The response must provide the information described in Table 259, organized according to Table 260.

Table 257 – Data in the payload of a *Check Version* command.

Field Name	Field Description	Type	Value		M/O
Software version	The current software version used by the device.	Length	U8	0x00 - 0x20	M
		Character String	U8	0x20 - 0xFE (each U8)	
Hardware version	The current hardware version used by the device.	Length	U8	0x00 - 0x20	M
		Character String	U8	0x20 - 0xFE (each U8)	
URL	The URL that can be checked to know if there are new software versions.	Length	U8	0x00 - 0xFF	O
		Character String	U8	0x20 - 0xFE (each U8)	

Table 258 – Data ordering of the payload of a *Check Version* command.

8	7	6	5	4	3	2	1	Octet
Software version - Length								1
Software version - First character								2
...								:
Software version - Last character								3 + n
Hardware version - Length								4 + n
Hardware version - First character								5 + n
...								:
Hardware version - Last character								6 + n
URL - Length								8 + n
URL - First character								9 + n
...								:
URL - Last character								10 + n ³⁴

³⁴ The value n represents an accumulation of an arbitrary number of bytes.

Table 259 – Data in the payload of a Response to a *Check Version* command.

Field Name	Field Description	Type	Value		M/O
Response Code	Value that indicates the state of the command reception/processing.	U8	0x00 - New version available. 0x11 - No newer version available. 0x12 - Server or Network are down. 0xFF - Fail: Unknown Error		M
Software version	The new software version.	Length	U8	0x00 - 0x20	M
		Character String	U8	0x20 - 0xFE (each U8)	
Hardware version	The minimum hardware version supported by the new software version.	Length	U8	0x00 - 0x20	M
		Character String	U8	0x20 - 0xFE (each U8)	
URL	The URL from where the new software version might be obtained.	Length	U8	0x00 - 0xFF	O
		Character String	U8	0x20 - 0xFE (each U8)	

Table 260 – Data ordering of the payload of a Response to a *Check Version* command.

8	7	6	5	4	3	2	1	Octet
Response Code								1
Software version - Length								2
Software version - First character								3
...								:
Software version - Last character								4 + n
Hardware version - Length								5 + n
Hardware version - First character								6 + n
...								:
Hardware version - Last character								7 + n
URL - Length								8 + n
URL - First character								9 + n
...								:
URL - Last character								10 + n ³⁵

³⁵ The value n represents an accumulation of an arbitrary number of bytes.

6.16.4.2 Upgrade Complete

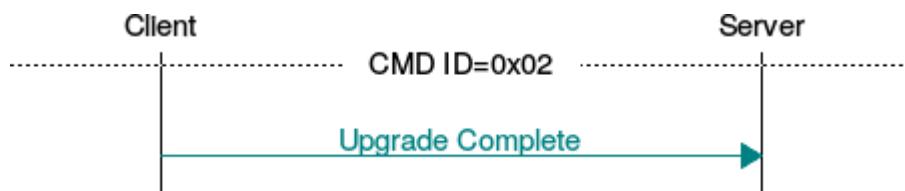


Figure 85 - SUOTA Interface, *Upgrade Complete* command.

This command must be sent to a server implementation of SUOTA interface, upon completion of an upgrade process to indicate the status of the operation.

The command must provide the information described in Table 261, organized according to Table 262. If the upgrade is successful then the *Software version* field must be present.

Table 261 – Data in the payload of an *Upgrade Complete* command.

Field Name	Field Description	Type	Value		M/O
Upgrade Status	Value that indicates the final state of the upgrade process.	U8	0x00 - Upgrade successful 0x11 - Upgrade aborted 0x12 - GMEP session error 0x13 - Invalid image 0x14 - Memory access error 0x15 - Battery too low 0xFF - Fail: Unknown error		M
Software version	The new software version, after the successful upgrade. This field is mandatory if upgrade is successful.	Length Character String	U8 U8	0x00 - 0x20 0x20 - 0xFE (each U8)	O

Table 262 – Data ordering of the payload of an *Upgrade Completed* command.

8	7	6	5	4	3	2	1	Octet
Upgrade Status								1
Software version - Length								2
Software version - First character								3
...								:
Software version - Last character								4 + n

Annex:

Terms and Conditions of Use

Version: 23 January 2014

The copyrights in this Document and the specifications contained herein are owned by the ULE Alliance and its Members (hereinafter, the ‘ULE Alliance’). Use of this Document and the specifications and any related intellectual property (collectively, the “Specification”), is governed by this Notice and the IPR Regulation of the ULE Alliance, where appropriate.

Use of the Specification by anyone who is not a member of the ULE Alliance or an Adopter is prohibited. However, such parties shall be permitted to view this Document. Requests for permission to reprint this Document, in whole or in part, or requests for a license to reproduce and/or distribute this Document, in any form, must be submitted via email to secretariat@ulealliance.org or in writing to: ULE Alliance Secretariat, Wabernstr. 40, 3007 Bern, Switzerland.

Elements of this Document and Specification may be subject to third party intellectual property rights, including without limitation, patent, copyright or trademark rights (such a third party may or may not be a Member of ULE Alliance). Nothing in this Document or Specification may be construed as a license to use such intellectual property, and such license must be separately sought with the right holder. The ULE Alliance is not responsible and shall not be held responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights.

THE DOCUMENT AND SPECIFICATION ARE PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, SATISFACTORY QUALITY, OR REASONABLE SKILL OR CARE, OR ANY WARRANTY ARISING OUT OF ANY COURSE OF DEALING, USAGE, TRADE PRACTICE, PROPOSAL, SPECIFICATION OR SAMPLE.

Each Member and Adopter hereby acknowledges that products equipped with the Specification (hereinafter, ‘Products’) may be subject to various regulatory controls under the laws and regulations of various governments worldwide. Each Member and Adopter is solely responsible for the compliance by Products with any such laws and regulations and for obtaining any and all required authorizations, permits, or licenses for their Products related to such regulations within the applicable jurisdictions. Each Member and Adopter acknowledges that nothing in the Specification provides any information or assistance in connection with securing such compliance, authorizations or licenses. NOTHING IN THE SPECIFICATION CREATES ANY WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING SUCH LAWS OR REGULATIONS.

IN NO EVENT WILL THE ULE ALLIANCE BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS, LOSS OF USE OF DATA, INTERRUPTION OF BUSINESS, OR FOR ANY OTHER DIRECT, INDIRECT, SPECIAL OR EXEMPLARY, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES OF ANY KIND, IN CONTRACT OR IN TORT, IN CONNECTION WITH THIS DOCUMENT OR THE INFORMATION CONTAINED HEREIN, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The ULE Alliance reserves the right to adopt any changes or alterations to the Document or Specification as it deems necessary or appropriate without notice.