

Open Access Protocol (OAP)

FUNCTION DESCRIPTION



NOTICE

The information contained in this document is believed to be accurate in all respects but is not warranted by Mitel Networks™ Corporation (MITEL®). Mitel makes no warranty of any kind with regards to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The information is subject to change without notice and should not be construed in any way as a commitment by Mitel or any of its affiliates or subsidiaries. Mitel and its affiliates and subsidiaries assume no responsibility for any errors or omissions in this document. Revisions of this document or new editions of it may be issued to incorporate such changes.

No part of this document can be reproduced or transmitted in any form or by any means - electronic or mechanical - for any purpose without written permission from Mitel Networks Corporation.

TRADEMARKS

The trademarks, service marks, logos and graphics (collectively "Trademarks") appearing on Mitel's Internet sites or in its publications are registered and unregistered trademarks of Mitel Networks Corporation (MNC) or its subsidiaries (collectively "Mitel") or others. Use of the Trademarks is prohibited without the express consent from Mitel. Please contact our legal department at legal@mitel.com for additional information. For a list of the worldwide Mitel Networks Corporation registered trademarks, please refer to the website: <http://www.mitel.com/trademarks>.

© Copyright 2017, Mitel Networks Corporation

All rights reserved

Contents

1. Introduction	1
1.1 Abbreviations	1
1.2 Definitions	1
2. Technical Solution.....	2
2.1 System Outline.....	2
2.2 Application Examples.....	2
3. System Requirements	5
3.1 Licences	5
4. Protocol Description	6
4.1 Introduction.....	6
4.1.1 Protocol Version	6
4.1.2 Transport	6
4.1.3 Character Encoding	6
4.2 OAP Protocol Version 3.x.....	7
4.2.1 Structure	7
4.2.2 Header Data.....	8
4.2.3 Body Data.....	8
4.2.4 Response Data	8
4.3 OAP Version 4.x (OAPv4).....	9
4.3.1 Structure	9
4.3.2 Header Data in request from server	10
4.3.3 Header Data in request from client	10
4.3.4 Body Data.....	11
4.3.5 Response Data	11
4.3.6 Example of sending Message with requested Delivery Status to Server	14
4.3.7 Example receiving Alarm from Server	15
4.3.8 Example of sending Message with Confirmation Request to Server and receiving Confirmation	16
4.4 System Specific Requests	17
4.4.1 Registration of Client	18
4.4.2 Supervision Request.....	19
4.4.3 Disconnection Request.....	19
4.4.4 Subscription Request.....	20
4.4.5 End Subscription Request	21
4.4.6 Response from Server	22
4.5 Services for OAP version 3.x	22
4.5.1 Alarm.....	22
4.5.2 User Data.....	23

4.5.3	Message.....	24
4.5.4	Erase Message.....	26
4.5.5	Service Status Response.....	26
5.	Related Documents	30

1. Introduction

The Open Access Protocol (OAP) is used as an interface between external applications or systems. The protocol enables exchange of data between the systems, for example handset alarms and messages.

Examples of applications are integration of cordless telephone system. It can also be used to interface a building automation system or handle machine alarms in a production process, and so on.

Mitel CPDM3 and Mitel WSM3 versions of the Unite server are serving the same purpose, but the first one is used with MX-ONE and the later one is used for MiVoice Business and MiVoice 250. For simplifications the code examples in this document are using the CPDM3, but that can of course be replaced by WSM3.

1.1 ABBREVIATIONS

XML	Extensible Markup Language.
LAN	Local Area Network.
TCP	Transmission Control Protocol.
CPDM3	In this document, CPDM3 (for use with MX-ONE) will furthermore be called Unite module.
WSM3	In this document, WSM3 (for use with MiVoice Business and MiVoice 250) is called Unite module.

1.2 DEFINITIONS

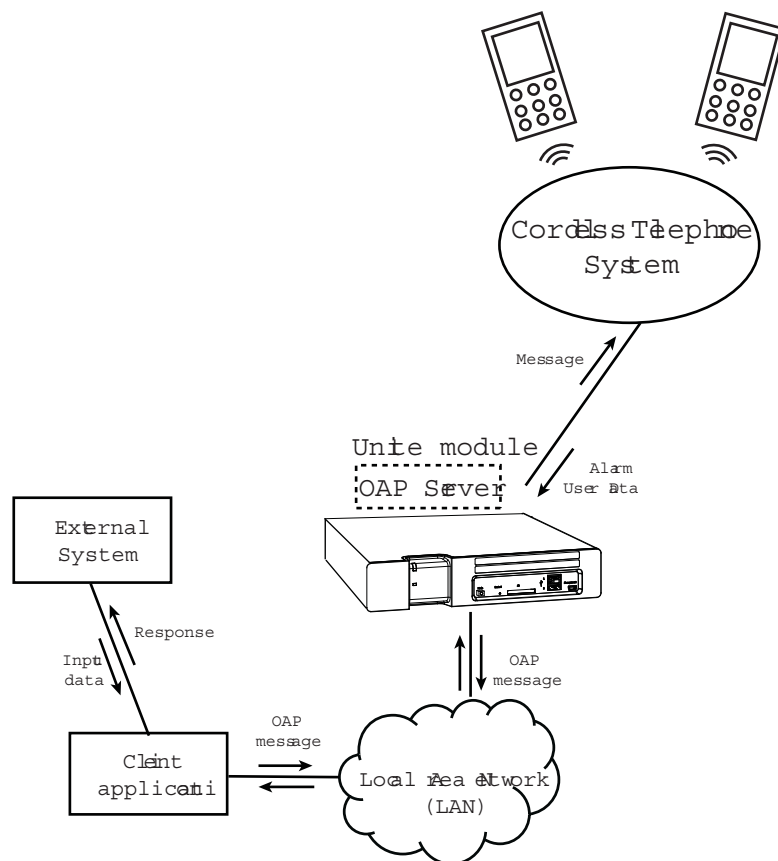
Request block	XML document sent between the OAP Server and client application.
Response block	XML document sent as response to a previously sent request block.

2. TECHNICAL SOLUTION

The Open Access Protocol can be used by a client application for communication with a cordless telephone system. The client communicates with an OAP Server. The protocol is XML based and is sent to the OAP Server over the LAN using a TCP connection.

2.1 SYSTEM OUTLINE

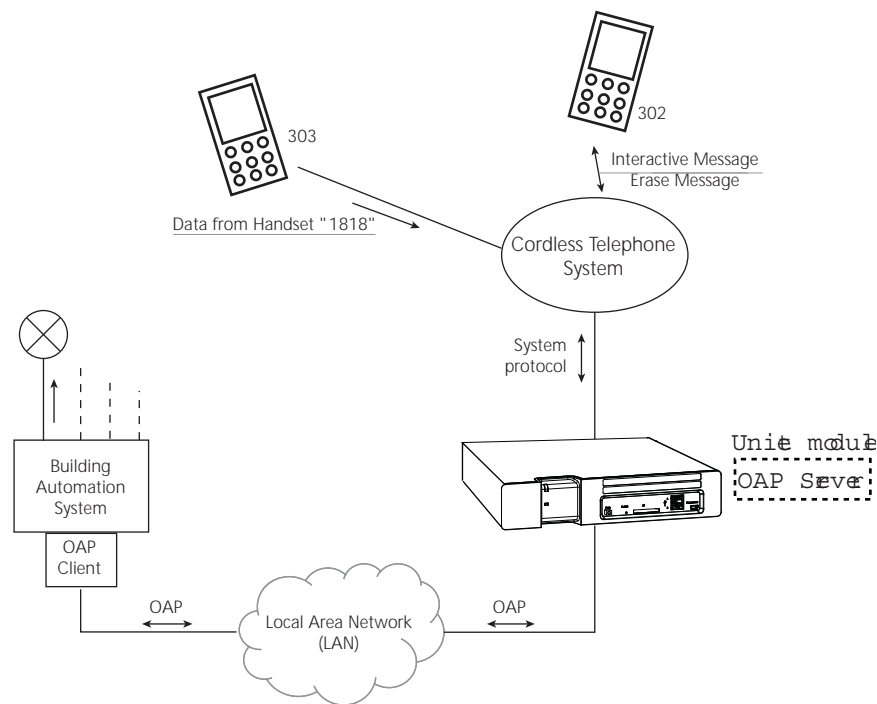
Figure 1. Receiving alarms/data from handsets and transmitting messages to handsets.



2.2 APPLICATION EXAMPLES

The handsets in the system are configured to turn on and off the light. The light is turned on/off by sending data (in this case "1818") from the handset. In addition, the Building Automation System can send Interactive Messages to the handsets, and take actions according to the response that is sent from the handset.

Figure 2. Controlling a Building Automation System from a handset via OAP.

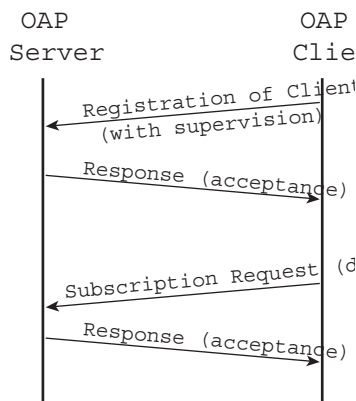


In this example, the Building Automation System has implemented the OAP protocol to be able to communicate with the handsets in the cordless telephone system. This implementation will hereafter be called OAP Client. The unit that includes the OAP Server transforms the OAP message to the protocol used by the receiving system.

Protocol Communication for Light Control via User Data

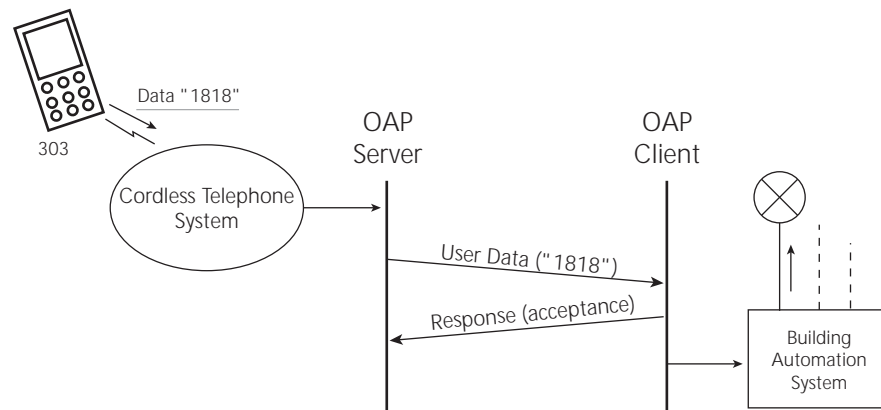
Before the OAP Client can receive User Data from the handset, it has to register in the OAP Server and subscribe for the information.

Figure 3. Registration of the Building Automation System and subscription for User Data from handsets.



Registration and subscription for user data and alarms only has to be done once when communication with the server is established.

Figure 4. Using data from handset to control a Building Automation System.

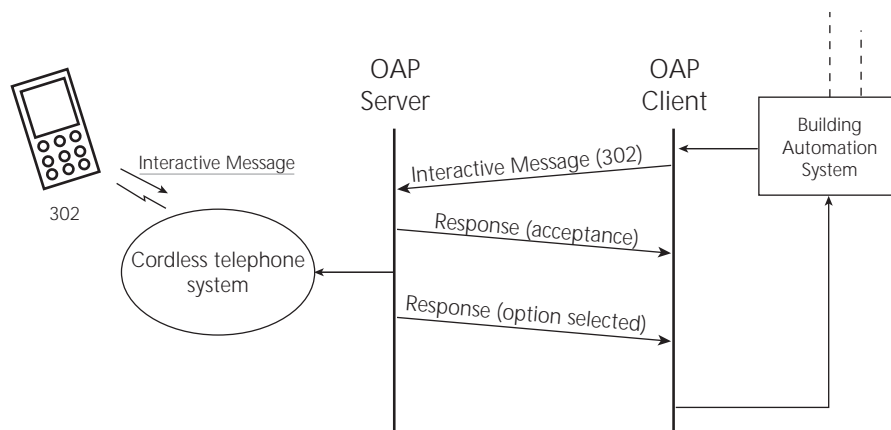


When the handset transmits the data, it is sent to the OAP Server. The OAP Server then transmits the information to the OAP Client. When the OAP Client receives the information, it sends a response to the OAP Server that the OAP message has been received. It interprets the information in the OAP message and turns on the light.

Protocol Communication for Error Indication to Handsets via Interactive Messaging

When an error occurs in the Building Automation System, an Interactive Message is sent to a handset in the cordless telephone system. In the message there is a set of answer options, for example, send detailed report, halt system, and call service personnel.

Figure 5. Error indication and control via Interactive Messaging.



When an error occurs the OAP Client sends an Interactive Message to the OAP Server in the cordless telephone system. The OAP Server sends a response to the OAP Client that the message has been received. It interprets the message and sends it to the addressed handset.

When the handset sends a response that should be handled by the Building Automation System, the response is sent from the OAP Server to the OAP Client. The OAP Client then interprets the response and the Building Automation System takes appropriate actions.

3. System Requirements

Up to 50 clients can connect to the same OAP server.

Handset

The handset must have support for the OAP services (that is alarm, user data and paging with confirmation) that are to be used. See corresponding Data Sheet for the handset for more information.

System 900

Transmitting and receiving equipment. See System Descriptions for Ascom Paging System for more information.

Cordless Telephone System

Equipment for message control. See documentation for the cordless telephone system for more information.

4. Protocol Description

This chapter describes the general structure of the Open Access Protocol, and also the transportation between the OAP Server and a client.

4.1 INTRODUCTION

The Open Access Protocol is a data transport protocol that uses XML documents as data carriers. The XML document will be called “request block” in this description. A response to a request block will be called “response block”.

4.1.1 PROTOCOL VERSION

The current protocol versions supported by the OAP Server are 3.x and 4.x. Version 4.x is also known as OAPv4. Available services in version 4.x depends on the OA-XML version. Both protocol version 3.x and 4.x are described in this document.

The OAP version is included in all blocks that are sent between the OAP Server and the client. The major number in the OAP version is a compatibility number that indicates whether the OAP Server can handle the block from the client or not.

Example:

A block from the client with OAP version 3.10 will be handled if the OAP Server supports version 3.x (x is any value).

4.1.2 TRANSPORT

The LAN is used for the communication between the server and the client and the messages are sent with TCP to the OAP Server. The OAP Server listens to port 1321 for messages sent with protocol version 3.x, and to port 1322 for messages sent with protocol version 4.x. A client application using protocol version 3.x must submit the port number and IP address that the OAP Server should use for responses to the client, while a client application using protocol version 4.x must only submit the port number.

If there is an open TCP connection to the client with a subscription for the information that the OAP Server sends, that connection is used, else the OAP Server will use a temporary connection.

If the OAP Server starts the TCP session and use a temporary connection for distribution of the service, it will disconnect the TCP session after a received response or after approximately 15 seconds if no response is received.

The recommendation for the client is to listen to the port given in the subscription and not keep the connection where the subscription was started.

4.1.3 CHARACTER ENCODING

XML element tag names

The element names used within the service protocol uses the characters 0x20 – 0x7F in US-ASCII (ISO 646) only.

XML data content

All characters within a block must use ISO 8859-1 encoding for protocol version 3.x and UTF-8 encoding for protocol version 4.x.

The data content between a start and end tag must be escaped according to XML standard.

Example:

- The character '&' must be written '&'.

Any character can also be written with its ASCII code in hex, for example '' for carriage return (ASCII character 013). This is normally not needed, but can be good to know to solve problems.

4.2 OAP PROTOCOL VERSION 3.X

4.2.1 STRUCTURE

The request block consists of two parts, header and body. The header contains address information and the body contains the data. The response block contains status information for a previously sent request block.

Message ID

All blocks includes an ID. The ID is hexadecimal number between 1 and FFFFFFFF. The ID is used as a reference when sending a response to a previously sent request block.

Recommendation:

The 16 most significant bits should be a random number that is regenerated at every reboot. The 16 least significant bits is then a sequence number that is increased by one for each request block.

Request Block

The request block will look as follows:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<OAP-request version="3.0" id="xxx">
  <Header>
    Addressing information
  </Header>
  <Body service="service_name">
    Service data
  </Body>
</OAP-request>
```

Text that is written in *italic* is the data.

Response Block

The response block will look as follows:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<OAP-response version="3.0" id="xxx">
  <Status service="service_name">
    Response status
  </Status>
</OAP-response>
```

Text that is written in *italic* is the data. The first response block will have the status 'Acceptance' and is sent as soon as the receiving part has evaluated the content and checked that the request can be handled. The client should always wait for the first response block before terminating the connection to the OAP Server. If the client has not received a response within 15 seconds the connection can be terminated.

The response block is described further in [4.4.6 Response from Server](#) on page 22 and [4.5.5 Service Status Response](#) on page 27.

4.2.2 HEADER DATA

The header data contains addressing information that the receiver of a request block uses when sending a response block.

The header in the request block looks as follows:

```
<Header>
  <IP>Senders IP address</IP>
  <Port>Senders Port number</Port>
</Header>
```

Elements

IP (optional)	The senders IP address. 7-15 characters. The IP address in decimal representation (for example 100.10.10.5)
Port (optional)	The senders port number. 1-5 decimal characters

If any of these elements are missing or empty, 'Acceptance' status only will be sent as response.

4.2.3 BODY DATA

The body data is the information that is sent to an application or unit. There can only be one body in a request block. The body tag has an attribute that contains the service in the request, for example alarm or message. For more information about the look of the body and the elements, see [4.5 Services for OAP version 3.x](#) on page 22.

4.2.4 RESPONSE DATA

The response data includes information about the process of the request block in the system.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<OAP-response version="3.0" id="xxxx">
  <Status service="service_name">
    <Type>Status Type</Type>
    <Code>Description code</Code>
    <Text>Descriptive text</Text>
    <TypeSpecific>Elements that are specific for the status type.</TypeSpecific>
  </Status>
</OAP-response>
```

Elements

Type	The type of the response. Range: 1-99 1 : Acceptance 2 : Availability 3 : Delivery 4 : Manual acknowledge 5 : Interaction response
Code	Code that describes the status type. 3 decimal characters. The codes are described in connection with the service descriptions.
Text	Text that describes the status that has been received. 1-160 characters.
TypeSpecific	Elements that are dependent on the status type. The type specific elements are described in connection with the services.

4.3 OAP VERSION 4.X (OAPV4)

4.3.1 STRUCTURE

An OAP block version 4 consists of header and body, just as version 3. The header contains address information and the body contains the data. There is also a response block that contains status information for a previously sent request block. A significant difference between version 3.x and 4.x is that the response block for version 4.x only contains delivery status for a sent request while the response for version 3.x also carries other information about Manual acknowledge and interaction response. Manual acknowledge and interaction response are instead sent as new request blocks from server to client in version 4.x.

Version 4 differs from version 3 in several ways:

- The XML encoding is UTF-8 instead of ISO-latin-1
- The body can contain services defined by the OA-XML protocol
- IP address is not included in header

Message ID

All blocks include an ID that is set as attribute in the <OAP-request> and <OAP-response> element. The ID is hexadecimal number between 1 and FFFFFFFF. The ID is used as a reference when sending an OAP-response containing status for a previously sent request block.

Note that the OA-XML service within the <Body> element also has an ID that is optional. This is as reference between two OA-XML services, for instance when receiving an OAP-request from server containing the service "Message_Confirmation_Response" for a previously sent OAP-request from client containing the service Message_with_Confirmation_Request. See example for this scenario in [4.3.8 Example of sending Message with Confirmation Request to Server and receiving Confirmation](#) on page 16.

Recommendation:

The 16 most significant bits should be a random number that is regenerated at every

reboot. The 16 least significant bits is then a sequence number that is increased by one for each request block.

Request Block from server

The request block will look as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAP-request version="4.0" id="5">
  <Header>
    <Port> 1322 </Port>
    <From> Call ID of original sender </From>
    <FromEx>
      <Addr>Address of the original sender</Addr>
    </FromEx>
  </Header>
  <Body service="OA-XML">
    OA-XML service
  </Body>
</OAP-request>
```

Request Block from client

The request block will look as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAP-request version="4.0" id="6">
  <Header>
    <Port> Clients IP port </Port>
    <To> Call ID of destination </To>
    <AvailabilityStatus/>
    <DeliveryStatus/>
  </Header>
  <Body service="OA-XML">
    OA-XML protocol service
  </Body>
</OAP-request>
```

4.3.2 HEADER DATA IN REQUEST FROM SERVER

The header data in a request from server can consist of the following elements:

Port	The server port. Always 1322
From (optional)	This is the Call ID of the user that originally sent the message. It can for instance be the number of the handset when an alarm is distributed from the OAP server to the client.
FromEx (optional)	This group of elements contains extended information about the sender. The element can be used in combination with the child.
Addr (optional)	This is the address of the device/application that originally sent the message.

4.3.3 HEADER DATA IN REQUEST FROM CLIENT

The header data in a request from client can consist of the following elements:

Port	The senders port number
------	-------------------------

To (optional)	This is the Call ID of the user that shall receive the message. It can for instance be the number of the handset when sending a paging from client.
Availability (optional)	Request for availability status for the handset that is addressed in the "To" element. The OAP Server will send the information as soon as it has got it from the carrier system. NOTE 1: In order to get Availability status from the server the client has to request for DeliveryStatus as well. NOTE 2: Not all services and / or handset interfaces supports this status. If that case, no status will be sent back.
DeliveryStatus (optional)	Request for delivery status from the carrier system.

4.3.4 BODY DATA

The Body part of an OAP request will contain an OA-XML service. The only exception is when sending system specific requests to the OAP server, see chapter 4.4. OAP server will send back Acceptance with an error code if the body part of an OAP request from client contains illegal OA-XML or if the necessary license is missing.

The OAP server may be upgraded with new version of OA-XML to support new services. See Installation and Operation Manual for the Unite module that supports OAP.

4.3.5 RESPONSE DATA

Both OAP server and OAP client shall send status information back when they receive an OAP request.

The response block from OAP server will look as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAP-response version="4.0" id="5" provider="CPDM3">
  <Status>
    <Type> Status Type </Type>
    <Code> Status Code</Code>
  </Status>
</OAP-response>
```

The response block from OAP client will look as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAP-response version="4.0" id="5">
  <Status>
    <Type> Status Type </Type>
    <Code> Status Code</Code>
  </Status>
</OAP-response>
```

A status with type "Acceptance" must be sent back within 15 seconds. The <Code> element will define if the request was accepted or not. The OAP server and client can also send availability status and delivery status back to each other if the message includes the elements <Availability> and/or <Delivery> in the header of the request.

The availability status indicates if a handset is present or not. The delivery status is sent when the message has reached the handset.

Type	The type of the response. Range: 1-99 1 : Acceptance 2 : Availability 3 : Delivery
Code	Code that describes the status type. 3 decimal characters. See below for list of status codes for each type.

Response Codes

Acceptance

A response with the status type 'Acceptance' is sent as soon as the receiving part has evaluated the content and checked that the request can be handled. The client should always wait for the first response block before terminating the connection to the OAP server. If the client has not received a response within 15 seconds the connection can be terminated. See [4.4.6 Response from Server](#) on page 22 for status codes.

Availability

The availability status must be sent when the OAP server/client has received the information from the handset. The codes that can be received are listed below.

Code	Code that describes the status type Availability. 200-299 : User is present 200 : Present 400-499 : User is absent 400 : General absent 402 : Manual absent 404 : In storage rack 405 : Deprecated, no longer used
------	---

Delivery

The delivery status will be sent when the message has been delivered to the address in the <To> element, i.e. it does not guarantee that the user has read the message.

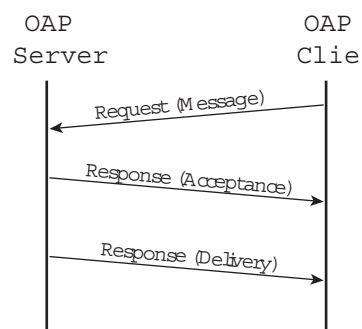
Code	Code that describes the status type Delivery. 100-199 : Message has not been delivered 200-299 : Message has been delivered 200 : Delivered 300-399 : Try again 300 : Unknown error 320 : Congestion 400-999 : Message has not been delivered 400 : Unknown error 401 : Communication error 402 : Licence error 403 : Message not understood 404 : Address not understood 405 : Deleted, too old 406 : Addressee is not reachable 407 : Not supported 408 : Addressee unknown
------	---

A device toggles from online to offline, which means that the OAP server sends an Availability_Status response for that device to the OAP client.

```
<?xml version="1.0" encoding="UTF-8"?>
<OAP-request id="22c40004" version="4.0">
  <Header>
    <LogId>vknfSfO0AGMveEyAAAA</LogId>
    <Port>1322</Port>
    <From>100</From>
    <User>
      <FirstName>Donald</FirstName>
      <LastName>Duck</LastName>
    </User>
    <FromEx>
      <MAC>FF0000000064</MAC>
      <Addr>100@172.20.13.106/WLAN?FF0000000064</Addr>
    </FromEx>
  </Header>
  <Body service="OA-XML">
    <service version="1.0">
      <Availability_Status>
        <Status>
          <Address>100@172.20.13.106/WLAN</Address>
          <Code>offline</Code>
        </Status>
      </Availability_Status>
    </service>
  </Body>
</OAP-request>
```

4.3.6 EXAMPLE OF SENDING MESSAGE WITH REQUESTED DELIVERY STATUS TO SERVER

Figure 6. Message with requested delivery status.



OAP Client sends OAP request:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAP-request version="4.0" id="5">
  <Header>
    <Port> 34553 </Port>
    <To>9420</To>
    <DeliveryStatus/>
  </Header>
```

```
<Body service="OA-XML">
  <service version="1.0">
    <Id>2</Id>
    <Message>
      <Subject>Subject Text</Subject>
      <Body>Body Text</Body>
      <Indication>
        <Beep>2</Beep>
      </Indication>
      <TTL>120</TTL>
      <Priority>7</Priority>
    </Message>
  </service>
</Body>
</OAP-request>
```

OAP Server sends OAP response acceptance as immediate response:

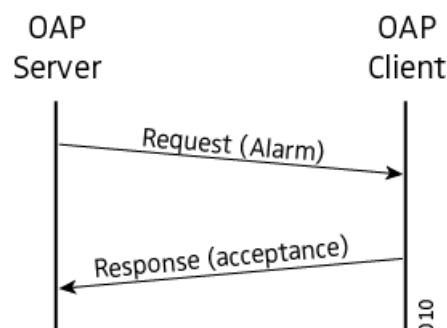
```
<?xml version="1.0" encoding="UTF-8"?>
<OAP-response version="4.0" id="5" provider="CPDM3">
  <Status>
    <Type> 1 </Type>
    <Code> 200 </Code>
  </Status>
</OAP-response>
```

OAP Server sends OAP response delivery when carrier system reports that message has been sent to portable.

```
<?xml version="1.0" encoding="UTF-8"?>
<OAP-response version="4.0" id="5" provider="CPDM3">
  <Status>
    <Type> 3 </Type>
    <Code> 200 </Code>
  </Status>
</OAP-response>
```

4.3.7 EXAMPLE RECEIVING ALARM FROM SERVER

Figure 7. Alarm from server



OAP Server sends OAP request:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAP-request version="4.0" id="5">
  <Header>
    <Port> 1322 </Port>
    <From>9430</From>
```

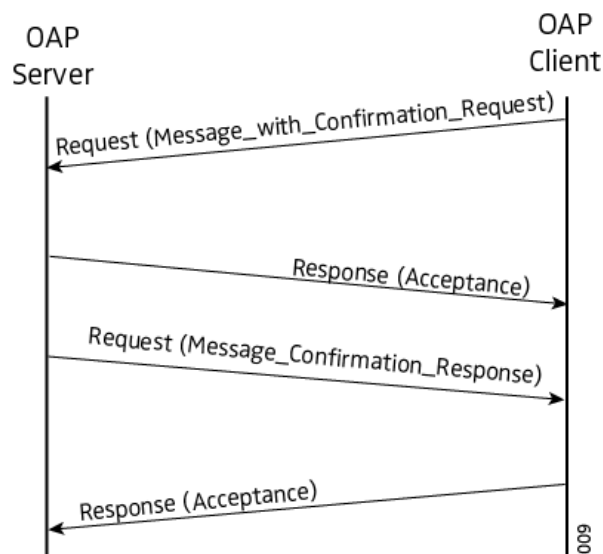
```
</Header>
<Body service="OA-XML">
  <service version="1.0">
    <Id>2</Id>
    <Alarm>
      <Type>monitorReceiver</Type>
      <Time>09:05:08:10:30:00</Time>
    </Alarm>
  </service>
</Body>
</OAP-request>
```

Client sends OAP response acceptance as immediate response:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAP-response version="4.0" id="5">
  <Status>
    <Type> 1 </Type>
    <Code> 200 </Code>
  </Status>
</OAP-response>
```

4.3.8 EXAMPLE OF SENDING MESSAGE WITH CONFIRMATION REQUEST TO SERVER AND RECEIVING CONFIRMATION

Figure 8. Message with confirmation request.



OAP Client sends OAP request:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAP-request version="4.0" id="5">
  <Header>
    <Port> 34553 </Port>
    <To>9420</To>
  </Header>
  <Body service="OA-XML">
    <service version="1.0">
      <Id>2</Id>
      <Message_with_Confirmation_Request>
        <Message>
```

```
        <Subject>Subject Text</Subject>
        <Body>Body Text</Body>
        <Indication>
            <Beep>2</Beep>
        </Indication>
        <TTL>120</TTL>
        <Priority>7</Priority>
    </Message>
    </Message_with_Confirmation_Request>
</service>
</Body>
</OAP-request>
```

OAP Server sends OAP response acceptance as immediate response:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAP-response version="4.0" id="5" provider="CPDM3">
    <Status>
        <Type> 1 </Type>
        <Code> 200 </Code>
    </Status>
</OAP-response>
```

OAP Server sends Message_Confirmation_Response as request:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAP-request version="4.0" id="6">
    <Header>
        <Port> 1322 </Port>
        <From>9420</From>
    </Header>
    <Body service="OA-XML">
        <service version="1.0">
            <Id>2</Id>
            <Message_Confirmation_Response>
                <Status>accepted</Status>
                <Priority>7</Priority>
            </Message_Confirmation_Response>
        </service>
    </Body>
</OAP-request>
```

Client sends OAP response acceptance as immediate response:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAP-response version="4.0" id="6">
    <Status>
        <Type> 1 </Type>
        <Code> 200 </Code>
    </Status>
</OAP-response>
```

4.4 SYSTEM SPECIFIC REQUESTS

The system specific requests can be used by the client to check the communication link to the OAP Server and also to subscribe for information that the OAP Server can send to the client, that is, alarm from handsets.

NOTE: Note that requests from clients that use protocol version 3.x port shall be sent to port 1321 and messages sent with protocol version 4.x shall be sent to port 1322.

4.4.1 REGISTRATION OF CLIENT

There are two reasons for a client to use registration:

- Supervision of the communication between client applications and the OAP Server. In case of a lost client, the system can take appropriate actions.
- Subscribe for information from the OAP Server to the client application.

It is not necessary to register the client if it only wants to send messages to the OAP server and supervision of lost clients is not required.

The format of the registration request when using version 3.x:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<OAP-request version="3.0" id="xxx">
  <Header>
    <IP>Senders IP address</IP>
    <Port>Senders Port number</Port>
  </Header>

  <Body service="addClient">
    <UpdTime>time</UpdTime>
  </Body>
</OAP-request>
```

The format of the registration request when using version 4.x:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAP-request version="4.0" id="xxx">
  <Header>
    <Port>Senders Port number</Port>
  </Header>
  <Body service="addClient">
    <UpdTime>time</UpdTime>
  </Body>
</OAP-request>
```

The client registers to the OAP Server by sending a request block with the service set to "addClient". When the client sends the registration request, all existing subscriptions that the client has will be deleted. This means that the client has to subscribe to the services that are to be used every time it is started.

The "UpdTime" sets the time limit for supervision requests. If the client has not sent a supervision request within the specified time limit, the OAP Server will consider the client disconnected and delete all existing subscriptions.

If the client does not get a response from the OAP Server within 15 seconds it should send a new registration request.

Elements

UpdTime	This element tells the OAP Server that the client will send a supervision request within the specified time. 10-900 seconds.
---------	---

4.4.2 SUPERVISION REQUEST

The supervision request is sent from the client to the OAP Server to check that the communication link is working and that the OAP Server is running. It is also the block that the client uses to tell the OAP Server that it is alive.

The format of the supervision request when using version 3.x:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<OAP-request version="3.0" id="xxx">
  <Header>
    <IP>Senders IP address</IP>
    <Port>Senders Port number</Port>
  </Header>

  <Body service="supervision">
  </Body>
</OAP-request>
```

The format of the supervision request when using version 4.x:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAP-request version="4.0" id="xxx">
  <Header>
    <Port>Senders Port number</Port>
  </Header>
  <Body service="supervision">
  </Body>
</OAP-request>
```

The client sends a supervision request to the OAP Server by sending a request block with the service set to "supervision". If the client has not received a response from the OAP Server within 15 seconds it should start over with a new registration.

4.4.3 DISCONNECTION REQUEST

This request should be sent when the client wants to stop the connection to the OAP Server. The OAP Server will then delete all existing subscriptions for the client. The format of the disconnection request when using version 3.x:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<OAP-request version="3.0" id="xxx">
  <Header>
    <IP>Senders IP address</IP>
    <Port>Senders Port number</Port>
  </Header>

  <Body service="removeClient">
  </Body>
</OAP-request>
```

The format of the disconnection request when using version 4.x:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<OAP-request version="4.0" id="xxx">
  <Header>
    <Port>Senders Port number</Port>
  </Header>
  <Body service="removeClient">
  </Body>
</OAP-request>
```

The client sends a disconnection request to the OAP Server by sending a request block with the service set to "removeClient". The client must not disconnect until it has got a response from the OAP Server. If a response has not been received within 15 seconds (time-out) the client can disconnect.

4.4.4 SUBSCRIPTION REQUEST

The client can subscribe for information from the OAP Server, provided that the client has registered with supervision. The subscription request will be denied if the client is not supervised by the OAP Server (that is, the UpdTime has to be specified in the registration request).

The requested services will be sent to the IP address and port that is specified in the header of the subscription request. The client may subscribe for several services by sending multiple requests.

The format of the subscription request when using version 3.x

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<OAP-request version="3.0" id="xxx">
  <Header>
    <IP>Senders IP address</IP>
    <Port>Senders Port number</Port>
  </Header>

  <Body service="subscriptionStart">
    <Service>Type of service</Service>
  </Body>
</OAP-request>
```

The format of the subscription request when using version 4.x:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAP-request version="4.0" id="xxx">
  <Header>
    <Port>Senders Port number</Port>
  </Header>
  <Body service="subscriptionStart">
    <Service>Type of service</Service>
  </Body>
</OAP-request>
```

The client subscribes for services by sending a request block with the service set to "subscriptionStart" to the OAP Server.

The services that are available for subscription are represented with a decimal number between 1 and 99 when using version 3.x. See below for available services. This number is submitted in the Service element in the subscription request.

Elements

Service	Decimal number that represents the requested service. Range: 1-99 1 : User Data 2 : Alarm
---------	--

Version 4.x uses the name of the OA-XML service instead of a number, for example "User_data". This name is submitted in Service element. The services available for subscription can be found in the Unite module as follows:

- 1 Select "Configuration".
- 2 Select "Advanced Configuration" under Other Settings.
- 3 Select Troubleshoot > System Information.

OAP Interface	
OA-XML version	1.6 (2013-06-13)
Licensed OA-XML services:	Receive / Send
Alarm	Yes / --
Availability_Query	-- / No
Availability_Response	No / --
Availability_Status	No / --
Call_Setup	-- / No
Call_Setup_Response	No / --
Device_Property_Request	-- / No
Device_Property_Response	No / --
Erase_Message	-- / Yes
Interactive_Message	-- / No
Interactive_Message_Response	No / --
Location_Data	No / --
Location_Request	-- / No
Location_Response	No / --
Message	-- / Yes
Message_Confirmation_Response	Yes / --
Message_with_Confirmation_Request	-- / Yes
User_data	Yes / --

4.4.5 END SUBSCRIPTION REQUEST

The client can stop the subscription for a service by sending an end subscription request. The format of the end subscription request when using version 3.x:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<OAP-request version="3.0" id="xxx">
  <Header>
    <IP>Senders IP address</IP>
    <Port>Senders Port number</Port>
  </Header>

  <Body service="subscriptionStop">
    <Service>Type of service</Service>
  </Body>
</OAP-request>
```

The format of the subscription request when using version 4.x:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAP-request version="4.0" id="xxx">
  <Header>
    <Port>Senders Port number</Port>
  </Header>
  <Body service="subscriptionStop">
    <Service>Type of service</Service>
```

</Body>
</OAP-request>

The client stops the subscription for a service by sending a request block with the service set to “subscriptionStop” to the OAP Server.

Elements

See [4.4.4 Subscription Request](#) on page 20 for more information about the elements.

4.4.6 RESPONSE FROM SERVER

For each of the requests from the client, the OAP Server will send a response with status 'Acceptance'. For more information about the response block, see [4.2.4 Response Data](#) on page 8 for version 3.x and [4.3.5 Response Data](#) on page 11 for version 4.x.

Acceptance

Code	Code that describes the status type Acceptance. 000–199 : Message not accepted 200–299 : Message accepted 200 : OK 300–399 : Try again 300 : Unknown error 320 : Congestion 400–999 : Message not accepted 400 : Unknown error 401 : Communication error (deprecated) 402 : Licence error 403 : Message not understood 404 : Address not understood 405 : Addressee unknown (deprecated) 406 : Not supported 407 : Client not registered 408 : OAP version mismatch
------	---

4.5 SERVICES FOR OAP VERSION 3.X

This is the services that are implemented in the Open Access Protocol version 3.

4.5.1 ALARM

This service includes information about an alarm from a handset within the system.

The block looks as follows:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<OAP-request version="3.0" id="xxxx">
  <Header>
    <IP>Senders IP address</IP>
    <Port>Senders Port number</Port>
  </Header>

  <Body service="alarm">
    <From>handset address</From>
```

```

    <Type>Alarm type</Type>
  </Body>
</OAP-request>

```

The block is identified as an alarm as the service is set to “alarm”.

Elements

From	The address of the handset that transmitted the alarm. 1-100 characters.
Type (optional)	The type of alarm. 0-999 1 : Test alarm 3 : Push-button alarm 1 4 : Push-button alarm 2

Location Elements

Code	The location ID. 1-50 characters.
Type (optional)	The type of location. Range: 0-999 1 : Undefined location 3 : DECT Base Station location 4 : DECT Beacon location 5 : Manual location
Time (optional)	Time in seconds since the location code was received. 6 decimal characters.

4.5.2 USER DATA

This service includes information about data that has been sent from a handset within the system.

The block looks as follows:

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<OAP-request version="3.0" id="xxx">
  <Header>
    <IP>Senders IP address</IP>
    <Port>Senders Port number</Port>
  </Header>

  <Body service="userData">
    <From>Handset address</From>
    <Data>Input data</Data>
  </Body>
</OAP-request>

```

The block is identified as user data as the service is set to “userData”.

Elements

From	The address of the handset that transmitted the alarm. 1-100 characters.
Data	A data string received from the handset. 1-255 characters.

4.5.3 MESSAGE

This service is used to send a message to a handset within the system.

The block looks as follows:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<OAP-request version="3.0" id="xxx">
  <Header>
    <IP>Senders IP address</IP>
    <Port>Senders Port number</Port>
  </Header>

  <Body service="message">
    <To>Handset address</To>
    <SubjectText>Subject for the message</SubjectText>
    <BodyText>Body for the message</BodyText>
    <Priority>Message priority</Priority>
    <BeepCode>Beep Code that is used for indication</BeepCode>

    <AvailabilityStatus/>
    <DeliveryStatus/>

    <AllowErase/>

    <ManualAckStatus>
      <Mode>Type of acknowledge</Mode>
    </ManualAckStatus>
  </Body>
</OAP-request>
```

The request block is identified as a message as the service is set to "message". It is possible to request status information by including the elements "AvailabilityStatus" and "DeliveryStatus". The availability status indicates if a handset is placed in a Charging Rack. The delivery status is sent when the message has reached the handset.

If it should be possible to erase the message later on the "AllowErase" element has to be included in the request block. The receiver of the request will then do any necessary preparations to support a remote erase of the message.

A manual acknowledge of the message can be requested by including the "ManualAckStatus" block. The acknowledge is sent as a response with status type 'Acknowledge'. The "Mode" element indicates if the user has the option to reject the message or if accept is the only possible option.

Elements

To	The address of the receiver of the message. For example a handset address. 1-100 characters.
SubjectText (optional)	The subject of the message, normally used for general information about the message. 0-150 characters. The text in the subject and body elements will be concatenated before transmission to handsets within System 900. If the text is too long, it will be truncated before transmission.
BodyText (optional)	Body of the message, normally used for the main content of the message. 0-500 characters. The text in the subject and body elements will be concatenated before transmission to handsets within System 900. If the text is too long, it will be truncated before transmission.
Priority (optional)	Priority of the message. Range: 1-9 Default value: 6 1 : alarm priority (should be used with special care) 2-4 : high priority 5-7 : normal priority 8-9 : low priority
BeepCode (optional)	A value that determines the indication in the handset. Range: 0-7 Default value: 2 See the manual of your handset for more information about how the handset interpreters the beep codes.
AvailabilityStatus (optional)	Request for availability status for the handset that is addressed in the "To" element. The OAP Server will send the information as soon as it has got it from the carrier system. Note 1: In order to get Availability status from the server the client has to request for DeliveryStatus as well. Note 2: Not all services and / or handset interfaces supports this status. In that case, no status will be sent back.
DeliveryStatus (optional)	Request for the status message 'Delivery'.
AllowErase (optional)	If this element is included, the message can be deleted by sending an "eraseMessage" block later on.

Acknowledge Elements

The acknowledge block is optional and is included when a response is requested.

Mode (optional)	Sets the types of acknowledge. Possible values: OnlyAccept - the user can only accept the message. AcceptAndReject - the user can accept or reject the message.
--------------------	--

4.5.4 ERASE MESSAGE

This service is used to erase a previously sent message.

The block looks as follows:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<OAP-request version="3.0" id="xxx">
  <Header>
    <IP>Senders IP address</IP>
    <Port>Senders Port number</Port>
  </Header>

  <Body service="eraseMessage">
    <To>Handset address</To>
    <OrigService>original service</OrigService>
    <Priority>Message priority</Priority>

    <DeliveryStatus/>
  </Body>
</OAP-request>
```

The request block is identified as an erase message request as the service is set to "eraseMessage". A message can only be deleted if the "AllowErase" element is included in the original message request. The erase message request should have the same reference ID as the previously sent message request. It should also be defined which service that was used for the original message, for example .

Elements

To	The address of the receiver of the message. For example a handset address. 1-100 characters.
OrigService	The service of the original message 1-100 characters.
Priority (optional)	Priority of the message. Range: 1-9 Default value: 6 1 : alarm priority (should be used with special care) 2-4 : high priority 5-7 : normal priority 8-9 : low priority
DeliveryStatus (optional)	Request for the status message 'Delivery'. The delivery status will be sent back when the message has reached the destination defined in the "To" element.

4.5.5 SERVICE STATUS RESPONSE

For all requests that are sent between the client application and the OAP Server a response block has to be sent. The reference ID and service name are identical to the ones in the request. Which types of statuses that can be sent for different services are described below. The response block is described in [4.2.4 Response Data](#) on page 8.

Alarm

When the client application has received an alarm block, it should check the block and then send a response block with status 'Acceptance' to the OAP Server.

Acceptance

See [4.4.6 Response from Server](#) on page 22 for more information about the codes for the acceptance status.

User Data

When the client application has received an user data block, it should check the block and then send a response block with status 'Acceptance' to the OAP Server.

Acceptance

See [4.4.6 Response from Server](#) on page 22 for more information about the codes for the acceptance status.

Message

When a message request has been sent the OAP server will send a response with 'Acceptance'. Normally this is done within 15 seconds. Additionally response blocks with availability, delivery, and manual acknowledge status can be received if they have been requested.

Acceptance

See [4.4.6 Response from Server](#) on page 22 for more information about the elements in a response block with acceptance status.

Availability

The availability status will be sent when the receiving application has received the information. The codes that can be received is listed below.

Code	Code that describes the status type Availability.
	200–299 : User is present 200 : Present 400–499 : User is absent 400 : General absent 402 : Manual absent 404 : In storage rack 405 : Deprecated, no longer used

Delivery

The delivery status will be sent when the message has been delivered to the address in the “To” element, i.e. it does not guarantee that the user has read the message.

Code	Code that describes the status type Delivery.
	100-199 : Message has not been delivered 200–299 : Message has been delivered 200 : Delivered 300–399 : Try again 300 : Unknown error 320 : Congestion 400–999 : Message has not been delivered 400 : Unknown error 401 : Communication error 402 : Licence error 403 : Message not understood 404 : Address not understood 405 : Deleted, too old 406 : Addressee is not reachable 407 : Not supported 408 : Addressee unknown

Manual Acknowledge

This message is sent when the user has accepted or rejected a message with request for acknowledge. If the original message was sent to a group number, more than one response can be sent depending on how many that has accepted or rejected.

Code	Code that describes the status type Manual Acknowledge.
	200–299 : Message is accepted 200 : Accepted 400–499 : Message is not accepted 400 : Rejected

The address of the handset that the response was sent from is included in the type specific element.

```
...
<TypeSpecific>
  <From>Address of the sender of the acknowledge</From>
</TypeSpecific>
</Status>
</OAP-request>
```


From	The address of the handset that transmitted the acknowledge. 1-100 characters.
------	---

Erase Message

When an erase message request has been sent, the OAP server will send a response with 'Acceptance'. Normally this is done within 15 seconds. Additionally a response block with delivery response status can be received if it has been requested.

Acceptance

See [4.4.6 Response from Server](#) on page 22 for more information about the response block with acceptance status.

Delivery

See [Message](#) on page 27 for more information about the response block with delivery status.

5. Related Documents

User Manuals for handsets included in the system