

Function Description

Ascom Open Access Protocol (OAP)

About this document

This document describes the main functions and components of the Open Access Protocol (OAP) solution that can be used together with Ascom professional messaging.

Cross-references in the document

Throughout this document you will find cross-references in the text which indicate further details that can be found in other sections of this document. The cross-references are coloured blue and linked to the relevant place in the document (example: see chapter [6. Document History](#) on page 41). Positioning your cursor over the cross-reference text and clicking the left mouse button will take you to the relevant section.

To return to the original page after viewing a cross-referred page in Adobe Acrobat or Adobe Reader, click on the "Previous View" arrow ( or ).

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	3
3 System Requirements.....	6
3.1 Licences	6
3.2 Configuration of the Interface Module.....	6
3.3 Programming the Handsets	6
4 Protocol Description	7
4.1 Introduction.....	7
4.2 OAP Protocol Version 3.x.....	8
4.3 OAP Version 4.x (OAPv4)	10
4.4 System Specific Requests.....	23
4.5 Services for OAP version 3.x.....	28
5. Related Documents	40
6. Document History	41

1. Introduction

The Open Access Protocol (OAP) is used as an interface between external applications or systems, and a cordless telephone system and/or System 900. The protocol enables exchange of data between the systems, for example handset alarms and messages.

Examples of applications are integration of cordless telephone system and System 900 services with a Nurse Call System. It can also be used to interface a building automation system or handle machine alarms in a production process etc.

1.1 Abbreviations

XML	Extensible Markup Language
LAN	Local Area Network
TCP	Transmission Control Protocol
IMS3	In this document, IMS3 will furthermore be called <i>Unite module</i> .
Unite CM	In this document, Unite CM will furthermore be called <i>Unite module</i> .

1.2 Definitions

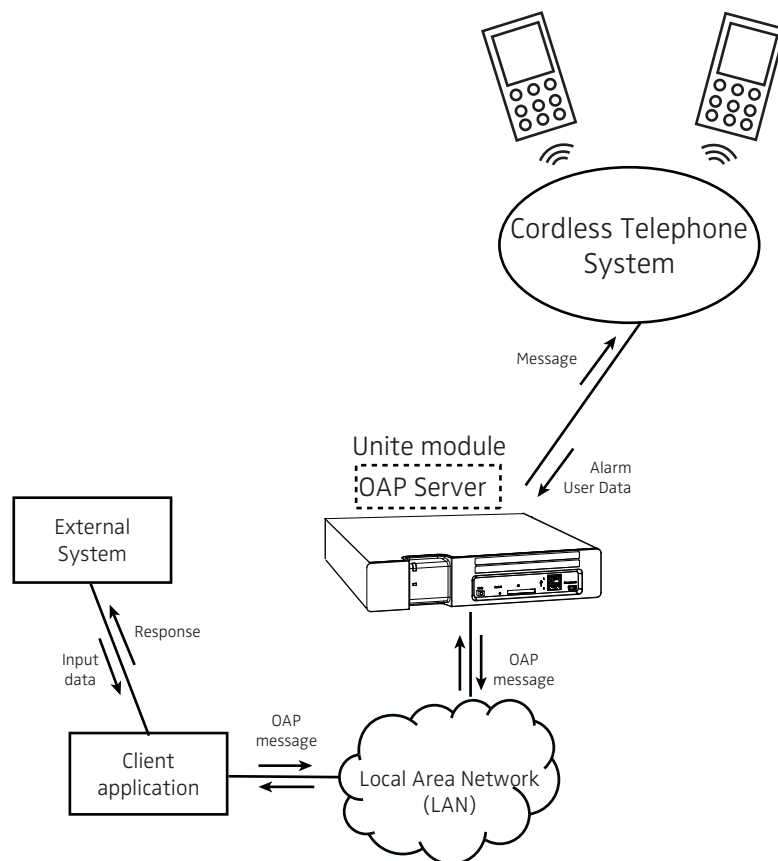
Request block	XML document sent between the OAP Server and client application
Response block	XML document sent as response to a previously sent request block.

2 Technical Solution

The Open Access Protocol can be used by a client application for communication with a cordless telephone system and System 900. 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

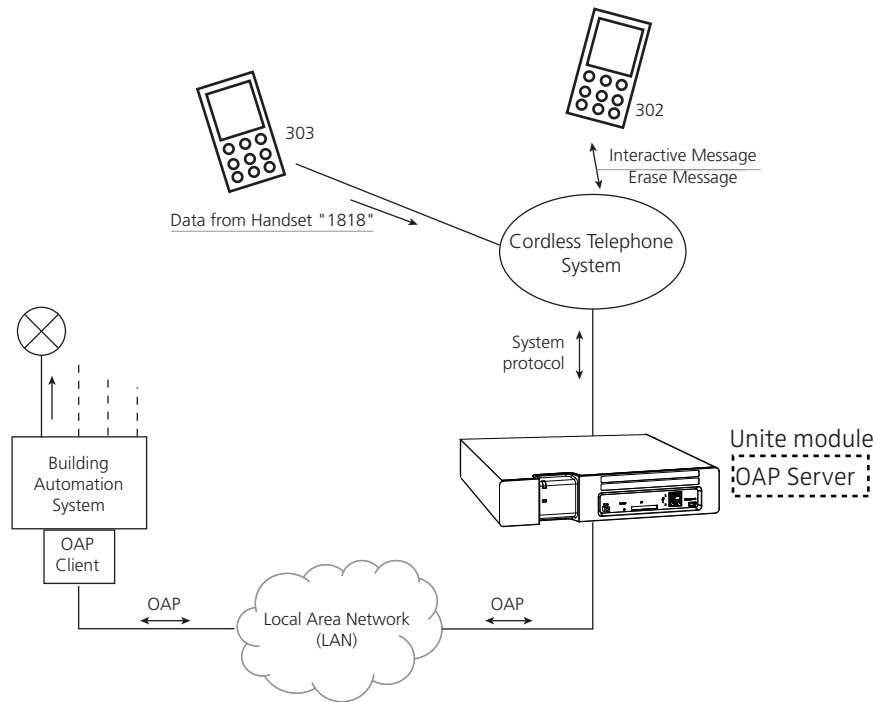
Figure 1. Receiving alarms/data from handsets and transmitting messages to handsets.



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.

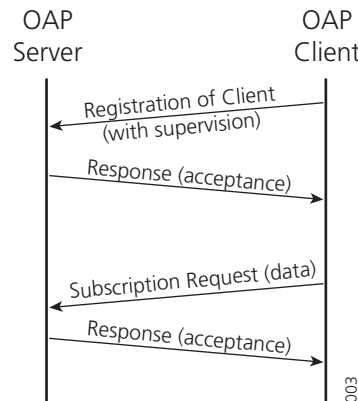


In this example, the Building Automation System has implemented the OAP protocol to be able to communicate with the handsets in the cordless telephone system. This implementation will hereafter be called OAP Client. The unit that includes the OAP Server transforms the OAP message to the protocol used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

Protocol Communication for Light Control via User Data

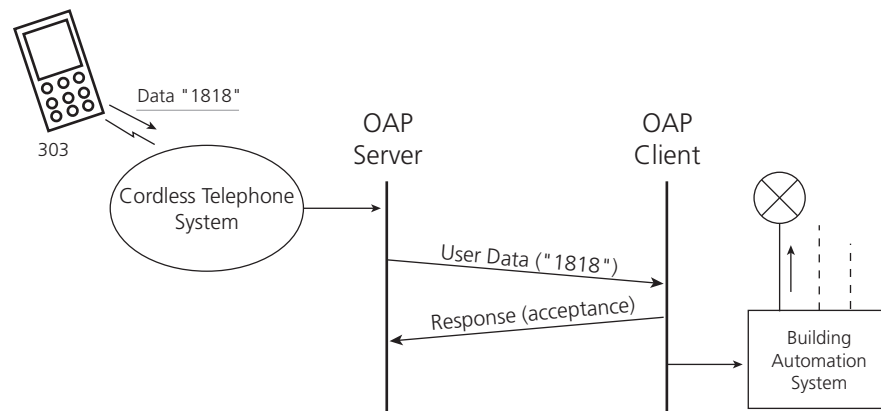
Before the OAP Client can receive User Data from the handset, it has to register in the OAP Server and subscribe for the information.

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.

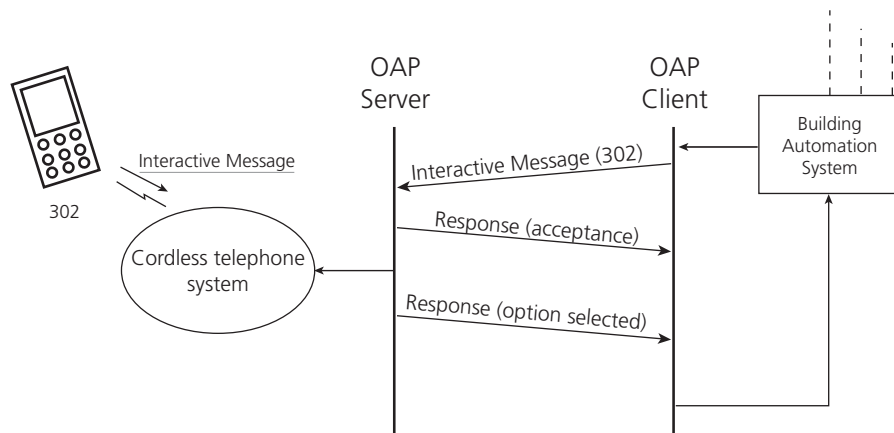


When the handset transmits the data, it is sent to the OAP Server. The OAP Server then transmits the information to the OAP Client. When the OAP Client receives the information, it sends a response to the OAP Server that the OAP message has been received. It interprets the information in the OAP message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

3.1 Licences

There are different OAP licences depending on which functionality that is required.

The following options are available for OAP version 3.x:

- Basic messaging with acknowledge
- Basic messaging, interactive messaging, and user data
- Basic messaging and alarm
- Basic messaging, interactive messaging, user data, and alarm

NOTE: Available licence options for OAP version 4.x, are listed in the data sheet for your Unite module.

3.2 Configuration of the Interface Module

The OAP Server can either be included in the Interface Module that the System 900 Central Unit distributes alarms and user data to, or another module. In both cases the Interface Module has to be configured to send the alarm and user data to the OAP Server. See documentation for the Interface Modules for more information.

3.3 Programming the Handsets

The handsets have to be programmed to be able to send user data. See the User Manual for each handset for information about how this is done and also which alarm types and location types that are available.

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 27 and [4.5.6 Service Status Response](#) on page 36.

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 28.

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="xxx">
  <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
- Manual acknowledge and IM response are sent as new OAP request from server to client instead of being embedded in an OAP response block.

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 [4.3.9 Example of sending Message with Confirmation Request to Server and receiving Confirmation](#) on page 18.

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>
    <ToEx>
      <Number>The Call ID of destination</Number>
    </ToEx>
    <From> Call ID of original sender </From>
    <FromEx>
      <MAC>MAC address of original sender</MAC>
      <Addr>Address of the original sender</Addr>
    </FromEx>
    <LogId>ID for tracking related correspondence between server and client</LogId>
    <AvailabilityStatus/>
    <DeliveryStatus/>
  </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>
    <ToEx>
      <Addr>The address of destination</Addr>
    </ToEx>
    <FromEx>
      <Number>Call ID of the original sender</Number>
    </FromEx>
    <LogId>ID for tracking related correspondence between server and client</LogId>
    <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
ToEx (optional)	This group of elements contains extended information about the receiver. In this case, it contains the child element <Number>. The <Number> can for example be the number to a smart phone.
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 elements <MAC> and <Addr>.
Addr (optional)	This is the address of the device/application that originally sent the message.
MAC (optional)	This is the MAC address of the device/application that originally sent the message.
LogId (optional)	This is the identification number used to keep related activities together. For example, the same identification number can be used to track which response that is related to certain request.
AvailabilityStatus (optional)	Request for availability status for the handset that is addressed in the <To> element. The OAP server will request for availability status from client when it distributes messages that are addressed to devices managed by the client, that is smart phones.
DeliveryStatus (optional)	Request for delivery status for the handset that is addressed in the <To> element.

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.
ToEx (optional)	This group of elements contains extended information about the addressee. In this case, it contains the child element <Addr>
Addr (optional)	This is the destination address (yyy@zzz) on the server side. This must be used if sending back a paging confirmation or interactive message response from client for a previous received paging from server. The value of the element must be the same as in <FromEx><Addr> elements in the previous received paging from the server side.
FromEx (optional)	This group of elements contains extended information about the sender. In this case, it contains the child element <Number>. It shall contain the number of the handset (that is smart phone) when sending a paging confirmation from client for a previous received paging from server. It is usually the same number as was set in the <ToEx><Number> element in the received paging.

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. <i>Note 1:</i> In order to get Availability status from the server the client has to request for DeliveryStatus as well. <i>Note 2:</i> 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.
LogId (optional)	This is the identification number used to keep related activities together. For example, the same identification number can be used to track which response that is related to certain request.

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. The OA-XML is an XML based protocol that is defined in a separate document, that can be found on the Configuration page on your Unite module. 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 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="IMS3 or Unite CM">
  <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. The OAP Server will only request Availability and Delivery when

sending messages that are addressed to handsets managed by the OAP client (that is pagers).

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 27 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 401 : Do not disturb 402 : Manual absent 403 : In meeting 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
------	---

4.3.6 OAP response handling when multiple clients receive paging services

Note: This is only applicable for OAP version 4.0.

If multiple clients are used in the system, all clients must obey the following rules when sending back requested OAP response for a received paging from server:

Acceptance response, client policy

Clients that do not have the handset requested by the server must respond with the status code 407 (Client not logged in). Only the client with the handset registered should respond as usual in a one-client system, that is, code 200 (OK) or context specific error code.

Acceptance response, server policy

If the server receives one of the following codes, the server will accept the response from the client and also forward it to the Unite system:

- First receipt of code 2xx from any client.
- All clients have sent a 4xx code or any non-2xx code.

Availability response, client policy

Client that does not have the handset requested by the server, must not send a availability response. Only the client with the handset registered shall respond.

Availability response, server policy

The server will consider having received response from client and forward it to the Unite system when the server received the first valid response from the any client.

Delivery response, client policy

Clients that do not have the handset requested by the server must respond with the status code 408 (addressee unknown). Only the client with the handset registered should respond as usual in a one-client system, that is, code 200 (OK) or context specific error code.

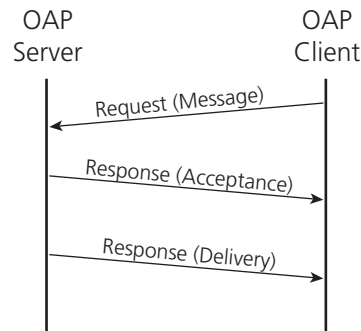
Delivery response, server policy

If the server receives one of the following codes, the server will accept the response from the client and also forward it to the Unite system:

- First receipt of code 2xx from any client.
- All clients have sent a 4xx code or any non-2xx code.

4.3.7 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="type of the Unite module">
  <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.

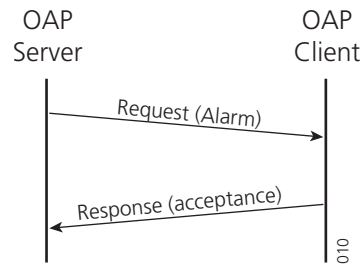
```

<?xml version="1.0" encoding="UTF-8"?>
<OAP-response version="4.0" id="5" provider="IMS3 or Unite CM">
  <Status>
    <Type> 3 </Type>
    <Code> 200 </Code>
  </Status>
</OAP-response>

```

4.3.8 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>
        <Location_Data>
          <Location>
            <Code>00103</Code>
            <Type>system900_IRLF</Type>
            <Time>10</Time>
          </Location>
        </Location_Data>
      </Alarm>
    </service>
  </Body>
</OAP-request>
  
```

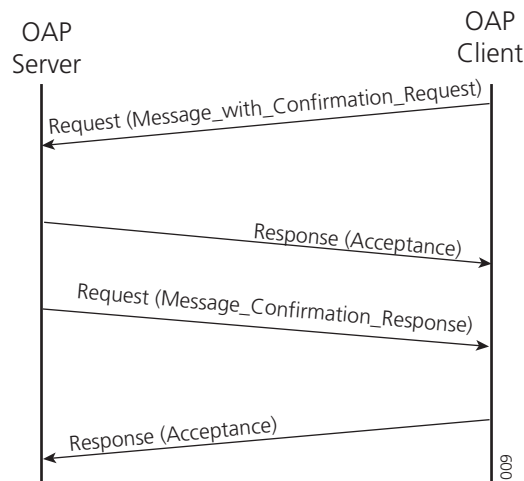
Client sends OAP response acceptance as immediate response:

```

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

4.3.9 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>
        <Subject>Subject Text</Subject>
        <Body>Body Text</Body>
        <Indication>
          <Beep>2</Beep>
        </Indication>
        <TTL>120</TTL>
        <Priority>7</Priority>
      </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="IMS3 or Unite CM">
  <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>

```

4.3.10 Example of Sending Interactive Message from OAP Server to Client

OAP Server sends an Interactive Message request:

```

<?xml version="1.0" encoding="UTF-8"?>
<OAP-request version="4.0" id="6">
  <Header>
    <Port> 1322 </Port>
    <ToEx>
      <Number>1234</Number>
    </ToEx>
    <FromEx>
      <Addr> xxx@yyy</Addr>
    </FromEx>
    <LogId>123456789</LogId>
  </Header>
  <Body service="OA-XML">
    <service version="1.0">
    <Id>2</Id>
    <Interactive_Message>
      <Message>
        <Subject> Subject text </Subject>
        ...
        ...
      </Message>
    </Interactive_Message>
    </Service>
  </Body>
</OAP-request>

```

Elements

Port	The server port. Always 1322
ToEx (optional)	This group of elements contains extended information about the receiver. In this case, it contains the child element <Number>. The <Number> can for example be the number to a smart phone.
FromEx (optional)	This group of elements contains extended information about the sender. In this case, it contains the child element <Addr>.
Addr (optional)	This is the address of the device/application that originally sent the message.
LogId	Common identification number for all requests
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

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>
```

4.3.11 Example of Interactive Message Response from Client

```
<?xml version="1.0" encoding="UTF-8"?>
<OAP-request version="4.0" id="7">
  <Header>
    <Port> 1322 </Port>
    <ToEx>
      <Addr>xxx@yyy</Addr>
    </ToEx>
    <FromEx>
      <Number> 1234</Number>
    </FromEx>
    <LogId>123456789</LogId>
  </Header>
  <Body service="OA-XML">
    <service version="1.0">
      <Id>2</Id>
      <Interactive_Message_Response>
        <Data_response>string</Data_response>
        <User_response> string</User_response>
        <Priority>7</Priority>
      </Interactive_Message_Response>
    </service>
  </Body>
</OAP-request>
```

Server sends OAP response acceptance as immediate response:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAP-response version="4.0" id="7">
  <Status>
    <Type> 1 </Type>
    <Code> 200 </Code>
  </Status>
</OAP-response>
```

4.3.12 Example of a Erase Message from OAP Server to Client

This service is used to erase a previously sent message.

The block looks as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAP-request version="4.0" id="5">
  <Header>
    <Port>1322</Port>
    <ToEx>
      <Number>1234</Number>
    </ToEx>
    <FromEx>
      <Addr>xxx@yyy</Addr>
    </FromEx>
  </Header>
  <Body service="OA-XML">
    <service version="1.0">
      <Id>2</Id>
      <Erase_Message>
        <Priority>7</Priority>
      </Erase_Message>
    </service>
  </Body>
</OAP-request>
```

The erase message request should have the same reference ID (for example, the value in the <Id> element) as the previously sent message 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>
```

4.3.13 Example of Interactive Message Update from OAP Server to Client

```
<?xml version="1.0" encoding="UTF-8"?>
<OAP-request version="4.0" id="5">
  <Header>
    <Port>1322</Port>
    <ToEx>
      <Number>1234</Number>
    </ToEx>
    <FromEx>
      <Addr>xxx@yyy</Addr>
    </FromEx>
  </Header>
```

```
<Body service="OA-XML">
  <service version="1.0">
    <Id>2</Id>
    <Interactive_Message>
      <Message>
        <Subject>Subject Text</Subject>
        <Body>Body Text</Body>
      </Message>
      <Update>
        <Method> onlyExisting</Method>
      </Update>
    </Interactive_Message>
  </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>
```

4.4 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 and user data 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	1.5 (2012-06-01)
Licensed OA-XML services:	Receive / Send
Alarm	Yes / –
Availability_Query	– / Yes
Availability_Response	Yes / –
Availability_Status	Yes / –
Call_Setup	– / Yes
Call_Setup_Response	Yes / –
Device_Property_Request	– / Yes
Device_Property_Response	Yes / –
Erase_Message	Yes / Yes
Interactive_Message	Yes / Yes
Interactive_Message_Response	Yes / Yes
Location_Data	Yes / –
Location_Request	– / Yes
Location_Response	Yes / –
Message	Yes / Yes
Message_Confirmation_Response	Yes / Yes
Message_with_Confirmation_Request	Yes / 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 25 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 9 for version 3.x and [4.3.5 Response Data](#) on page 13 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="xxx">
  <Header>
    <IP>Senders IP address</IP>
    <Port>Senders Port number</Port>
  </Header>

  <Body service="alarm">
    <From>handset address</From>
    <Type>Alarm type</Type>

    <LocationInfo>
      <Location>
        <Code>Location ID</Code>
        <Type>Location Type</Type>
        <Time>Location time</Time>
      </Location>
    </LocationInfo>
  </Body>
</OAP-request>
```

The block is identified as an alarm as the service is set to "alarm". The "LocationInfo" element can include several "Location" elements.

Elements

From	The address of the handset that transmitted the alarm. 1-100 characters.
Type (optional)	The type of alarm. 0-999 0 : Monitor receiver alarm 1 : Test alarm 2 : No new location alarm 3 : Push-button alarm 1 4 : Push-button alarm 2 5 : Man-down alarm 6 : Pull-cord alarm 7 : Auxiliary alarm 128 : Verification alarm

Location Elements

Code	The location ID. 1-50 characters. Receiver Interface location (Ascom paging system only): The location is 6 hexadecimal characters (ssBmml) ss = system address, B = bus (A/B), mm = module address, l = input number.
Type (optional)	The type of location. Range: 0-999 1 : Undefined location 2 : IR or LF location (Ascom paging system only) 3 : DECT Base Station location 4 : DECT Beacon location 5 : Manual location 6 : Ascom paging system Receiver Interface location 7 : Fixed 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: 0-99 Default value: 2 0-9 used for Ascom paging system backward compatibility. 10-99 reserved
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. <i>Note 1:</i> In order to get Availability status from the server the client has to request for DeliveryStatus as well. <i>Note 2:</i> 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 Interactive Messaging

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

See also considerations for IM found in the *Function Description Interactive Messaging (IM)*, TD 92168GB.

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="interactiveMessage">
    <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/>

    <TTL>Time to live</TTL>
    <Update>Update information for an existing message</Update>
    <Reminder>
      <Attention>Time</Attention>
      <Session>Time</Session>
    </Reminder>

    <Option id="1..N">
      <Text>Option text</Text>
      <OnSelection>
        <CallNo type="xxx">Digits to call</CallNo>
        <CallDisconnData>Response data</CallDisconnData>
        <CallDisconnect/>
        <DataResp>Response data</DataResp>
        <UserResponse>Text before input prompt</UserResponse>
        <NewTTL>New Time to live</NewTTL>
        <EraseMessage/>
        <EraseOption>Option to erase</EraseOption>
      </OnSelection>
      <FuncKeyId>Function key ID</FuncKeyId>
    </Option>
  </Body>
</OAP-request>
```

The request block is identified as an interactive message as the service is set to "interactiveMessage". The interactive message request has the same elements as an ordinary message request except for the acknowledge elements. In addition, there are elements that specifies the answering options.

Elements and Attributes

See [4.5.3 Message](#) on page 30 for information about elements that are common to the message and interactive message requests.

TTL (optional)	The time that the message is valid after it has arrived to its destination. 1-999999 seconds.
Update (optional)	Indicates that the message is an update of an existing interactive message. The following alternatives is known: OnlyIfNoExists : update only if message is unknown. OnlyExisting : update only existing message.

Reminder Elements

The reminder block is optional and includes elements to request user attention.

Attention (optional)	Defines the interval between attentions for the message. 1-255 seconds.
Session (optional)	Defines how long time after an option in the message has been selected until the message is deleted. 1-255 seconds.

Option Elements

An option block defines an option in the interactive message. The interactive message request will include one option block for each option in the message.

Text (optional)	The text that indicates the option in the handset. Up to 100 characters.
FuncKeyId (optional)	Provides the possibility to create a hot key association with an option, i.e. 2 can be pressed to select the second option in the list. A, B, C : soft keys A2, B2, C2 : additional soft key selection that appear when the first has been selected. 0-9 : numeric keys.

Option Attributes

The option element has an optional attribute that identifies the option. The ID is used to update an option in an existing interactive message.

ID (optional)	Unique identifier for the option. Decimal value between 1 and 100.
------------------	---

OnSelection Elements

CallNo (optional)	Connect a call. Defines the number to be dialled. Up 32 characters. For DECT GAP systems the following characters have certain meaning. P (hex 50) : Pause T (hex54) : DTMF on - (hex 2D) : DTMF off R (hex 52) : Register-recall Note that the operation is dependent of PBX support.
----------------------	---

CallDisconnData (optional)	Send data when the call is disconnected. Defines the data to be sent. Up to 255 characters.
CallDisconnect (optional)	Disconnect call that has been started from this message.
DataResp (optional)	Send a predefined response. Defines the response that should be sent. Up to 255 characters.
UserResponse (optional)	Send a response defined by the user. The text prompt can be preceded by a predefined text. Up to 32 characters.
New TTL (optional)	Update time to live. 1-999999 seconds.
EraseMessage (optional)	Erase the message on selection.
EraseOption (optional)	Erase an option on selection. Specify the option to be deleted. If no option is specified, the selected option will be deleted. Decimal number between 1-99.

CallNo Attribute

Type (optional)	Defines how the call number should be interpreted. Possible values: first : initiate a call to the number. If a call is already connected, there will be no action. once : initiate a call to the number and disconnect directly. If a call is already connected, there will be no action. more : the digits will be sent as DTMF tones. Only valid when a call is connected. last : the digits will be sent as DTMF tones and then the call will be disconnected. Only valid when a call is connected.
--------------------	--

4.5.5 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 "interactiveMessage".

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.

4.5.6 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 9.

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 27 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 27 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 27 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
	401 : Do not disturb
	402 : Manual absent
	403 : In meeting
	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.
------	---

Interactive Message

When an interactive 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 interaction response status can be received if they have been requested.

Acceptance

See [4.5.6 Service Status Response](#) on page 36 for more information about the response block with acceptance status.

Availability

See [Message](#) on page 36 for more information about the response block with availability status.

Delivery

See [Message](#) on page 36 for more information about the response block with delivery status.

Interaction response

This response is sent when the user has selected an option in the interactive message. If the original message was sent to a group number, more than one response can be sent depending on how many that has selected an option. This response will also be sent when an update of an existing interactive message is unsuccessful.

Code	Code that describes the status type Interaction response. 200–299 : Option has been selected 200 : Option selected 201 : Option selected but truncated data 400–499 : Message is not accepted 400 : No existing interactive message to update
------	--

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

```
...  
<TypeSpecific>  
  <From>Address of the unit where an option was selected</From>  
  <DataResp>Predefined data option</DataResp>  
  <UserResponse>User defined data option</UserResponse>  
</TypeSpecific>  
</Status>  
</OAP-request>
```

From	The address of the handset where the option was selected. 1-100 characters.
DataResp (optional)	Predefined response that was sent when the option was selected. Up to 255 characters.

UserResponse (optional)	Response defined by the user. Up to 32 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 27 for more information about the response block with acceptance status.

Delivery

See [Message](#) on page 36 for more information about the response block with delivery status.

5. Related Documents

System Descriptions for connected systems

Data Sheets for handsets included in the system

User Manuals for handsets included in the system

Function Description, Applications based on
Sending Data from Handsets

TD 92095GB

Function Description, Alarm from Handset

TD 92099GB

Function Description, Interactive Messaging (IM)

TD 92168GB

6. Document History

For details in the latest version, see change bars in the document.

Version	Date	Description
E	01 September 2009	Added information about OAP 4.x (OAPv4)
F	28 March 2010	Removed spaces in chapter 5.4.4
G	06 December 2010	Updates in Request Block from server in chapter 5.3.1 and in chapter 5.3.2 Header Data in request from server on page 17.
H	04 January 2012	Message option with priority 0 removed. See change bars for additional changes.
I	20 November 2013	Updated: 4.3.5 Response Data , 4.3.7 Example of sending Message with requested Delivery Status to Server , 4.3.9 Example of sending Message with Confirmation Request to Server and receiving Confirmation : "Provider" attribute added in OAP-responses from OAP Server.