MESSAGE ACCESS PROFILE:

The Message Access Profile (MAP) specification defines a set of

features and procedures to exchange messages between devices. It

is especially tailored for the automotive handsfree use case where

an onboard terminal device (typically a Car-Kit installed in the car)

takes advantage of the messaging capability of a communication

device (typically a mobile phone).

**USER REQUIREMENTS AND SCENARIOS**

The following are the main scenarios that are covered by this profile:

1. Notifying a remote device of the arrival of new messages on a messaging device:

In this scenario, the Message Access Profile is used to transfer message arrival

notifications, typically from a mobile phone (MSE) to an onboard unit (MCE).

2. Browsing messages in a messaging device:

In this scenario, the messages that are locally stored in an MSE device can be

browsed and, if needed, retrieved individually by a MCE. A typical configuration would

be that of a Bluetooth car-kit or a PC browsing the content of a mobile phone's

message repository.

3. Uploading messages onto a messaging device:

In this scenario, messages are created on the MCE device and uploaded to the MSE

device for storage.

4. Deleting messages onto a messaging device:

In this scenario, the MCE device deletes selected messages on the MSE, e.g. received

Spam mails .

5. Sending messages through a remote device:

In this scenario, the MCE uses the messaging capabilities of the MSE to send

messages to a network.

**MESSAGE TYPES**

The following message types are supported by this profile:

 EMAIL: emails on RFC2822 or MIME type basis

 SMS: short messages for GSM networks [12] and CDMA networks [14]

 MMS: 3GPP MMS messages [13]

Profile fundamentals

The MCE device shall use the services of the MSE device only after successfully

creating a secure connection. This includes exchanging of security initialization

messages, creation of link keys, and enabling encryption. Security mode 2 or 3 shall be

supported for devices implementing the Bluetooth 2.0+EDR. Either the MSE or MCE may initiate bonding. At a minimum, the MSE shall support

Inquiry and Paging and the MCE shall support Inquiry Scan and Page Scan in order to

initiate bonding. Either device can initiate the link establishment.

Only the MCE can start a MAP session on OBEX level.

**BLUETOOTH SECURITY**

The two devices shall create a secure connection using the GAP authentication

procedure as described in the Generic Access Profile [2]. This procedure shall include

alternatively entering a Bluetooth Passkey code for legacy devices or Secure Simple

Pairing and will include creation of link keys. For the first case a fixed passkey may also

be used during the GAP bonding procedure.

The Message Access Profile mandates the use of several Bluetooth security features:

Bonding: The MCE and MSE shall be bonded before setting up a Message

Access Profile connection. Security Mode 4 shall be used if supported both by

MSE and MCE. Each of the four association models of Secure Simple Pairing

may be used (see [21], chapter 5 and [22], chapter 7) where it is recommended

to use authenticated link keys, i.e. using the association models 'Numeric

Comparison', 'Out Of Band' or 'Passkey Entry'.

For legacy devices either security mode 2 or 3 shall be used for the Message

Access Profile connection.

Encryption: The link between MCE and MSE shall be encrypted using Bluetooth

encryption.

Bluetooth Passkey: When using security mode 2 or 3, the MCE and MSE shall

prohibit the use of a zero-length Bluetooth passkey.

Furthermore, the following issues are mandated for devices complying with the

Message Access Profile:

Link keys: Combination keys shall be used for Message Access Profile

connections.

Encryption key length: The length of the encryption key should be at least 64

bits. For increased security, use of the maximum length allowed given regional

regulation is encouraged.

**MESSAGE FORMAT (X-BT/MESSAGE)**

Exchanged messages shall use the bMessage format. The bMessage object

encapsulates the delivered message objects and provides additionally a suitable set of

properties with helpful information. The general encoding characteristics as defined for

vCards in chapter 2 of [5] shall be applied. The formal BNF definition of the bMessage

format is as follows:

# OBJECT PUSH PROFILE

OPP defines the roles of push server and push client. These roles are analogous to and must interoperate with the server and client device roles that GOEP defines.

## Usage Scenarios

An example scenario would be the exchange of a contact or appointment between two mobile phones, or a mobile phone and a PC.

### Example Products

Here are a few examples of the types of devices that you might find using the OPP:

* Mobile Phone
* PC
* Laptop

## Getting Technical

The OPP defines two roles, that of a Push Server and a Push Client:

* **Push Server**– This is the device that provides an object exchange server.
* **Push Client**– This is the device that pushes and pulls objects to and from the Push Server.

The Baseband, LMP and L2CAP are the OSI layer 1 and 2 Bluetoothprotocols. RFCOMM is theBluetooth adaptation of GSM TS 07.10. SDP is the Bluetooth Service Discovery Protocol. OBEX is the Bluetoothadaptation of IrOBEX.

# Object Exchange (OBEX)

OBEX is a transfer protocol that defines data objects and a communication protocol two devices can use to exchange those objects. OBEX is designed to enable devices supporting infrared communication to exchange a wide variety of data and commands in a resource-sensitive standardized fashion.

## Usage Scenario

The typical example could be an object push of business cards to someone else. A more complex example is synchronizing calendars on multiple devices using OBEX.

### Example Products

Here are a few examples of the types of devices that you might find using the OBEX:

* PC
* Notebook
* PDA
* Mobile Phone

## Getting Technical

### Overview

OBEX uses a client-server model and is independent of the transport mechanism and transport API. A Bluetooth enabled device wanting to set up an OBEX communication session with another device is considered to be the client device. The OBEX protocol also defines a folder-listing object, which is used to browse the contents of folders on remote device. RFCOMM is used as the main transport layer for OBEX.

OBEX enables applications to work over the Bluetooth technology protocol stack as well as the IrDA stack. For Bluetooth enabled devices, only connection-oriented OBEX is supported. Three application profiles have been developed using OBEX which include SYNC, FTP and OPP.

# BluetoothApplication Profiles Using OBEX

Synchronization

* Basically, the synchronization means comparing two object stores, determining their inequalities, and then unifying these two object stores.

File Transfer

* At the minimum, the File Transfer profile is intended for sending and retrieving generic files to and from the Bluetooth device.

Object Push

* The Object Push profile is the special case of the File Transfer Profile forbearing objects and optionally pulling the default objects.

File Transfer applications can be implemented using OBEX. For the Object Push and Synchronization applications, content formats can be the vCard, vCalendar, vMessage, and vNotes formats. The vCard, vCalendar, vMessage, and vNotes describe the formats for the electronic business card, the electronic calendaring and scheduling, the electronic message and mails, and the electronic notes, respectively.

# L2CAP LLC & Adaptation Protocol

Provides a link-layer protocol between entities with a number of services

Relies on lower layer for flow and error control

Makes use of ACL links, does not support SCO links

Provides two alternative services to upper-layer protocols

L2CAP permits higher level protocols and applications to transmit and receive upper layer data packets(L2CAP Service Data Units, SDU) up to 64 kilobytes in length. L2CAP also permits per-channel flow control and retransmission via the Flow Control and Retransmission Modes.

### Data Packet Format

L2CAP is packet-based but follows a communication model based on channels. A channel represents a data flow between L2CAP entities in remote devices. Channels may be connection-oriented or connectionless.

**L2CAP SIGNALING COMMANDS**

Command reject command

Sent to reject any command

Connection commands

Used to establish new connections

Configure commands

Used to establish a logical link transmission contract between two L2CAP entities