ETSI TS 127 103 V3.0.0 (2000-01)

Technical Specification

Universal Mobile Telecommunications System (UMTS); Wide Area Network Synchronisation Standard (3G TS 27.103 version 3.0.0 Release 1999)

Reference

DTS/TSGT-0227103U

Keywords

UMTS

ETSI

Postal address

F-06921 Sophia Antipolis Cedex - FRANCE

Office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16
Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Internet

secretariat@etsi.fr
Individual copies of this ETSI deliverable
can be downloaded from
http://www.etsi.org
If you find errors in the present document, send your
comment to: editor@etsi.fr

Important notice

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

Copyright Notification

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://www.etsi.org/ipr).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by the ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables. The mapping of document identities is as follows:

For 3GPP documents:

3G TS | TR nn.nnn "<title>" (with or without the prefix 3G)

is equivalent to

ETSI TS | TR 1nn nnn "[Digital cellular telecommunications system (Phase 2+) (GSM);] Universal Mobile Telecommunications System; <title>

For GSM document identities of type "GSM xx.yy", e.g. GSM 01.04, the corresponding ETSI document identity may be found in the Cross Reference List on www.etsi.org/key

Contents

Foreword

This Technical Specification has been produced by the 3GPP.

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of this TS, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 Indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the specification;

1 Scope

This specification provides a definition of a Wide Area Synchronisation protocol. The synchronization protocol is based upon IrMC level 4.

The present document covers Wide Area Network Synchronisation between current and future mobile communication end-user devices, desktop applications and server-based information servers. This is a living document and, as such, it will evaluate new technologies (e.g. XML) for inclusion as they become readily available.

2 References

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

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.
- A non-specific reference to an ETS shall also be taken to refer to later versions published as an EN with the same number.

[1] Bluetooth: Bluetooth SIG, Bluetooth Specifications, version 1.0, July 1999. (http://www.bluetooth.com/)

[2] IrMC, Infrared Data Association, "Specifications for Ir Mobile Communications (IrMC)", version 1.1, 01 March 1999, plus all applicable errata. (http://www.irda.org/)

[3] IrOBEX, Infrared Data Association, "Ir Object Exchange Protocol IrOBEX", version 1.2, April 1999, plus all applicable errata. (http://www.irda.org/)

[4] vCalendar, the Internet Mail Consortium, "vCalendar - The Electronic Calendaring and Scheduling Exchange Format - Version 1.0", 18 September 1996. (http://www.imc.org/pdi/vcal-10.doc)

[5] vCard, the Internet Mail Consortium, "vCard - The Electronic Business Card - Version 2.1", 18 September 1996.(http://www.imc.org/pdi/vcard-21.doc)

[6] WAP, WAP Forum, "WAP Technical Specifications Suite", version 1.1, June 1999. (http://www.wapforum.com/)

[7] XML, W3C, “Extensible Markup Language (XML) 1.0”, v1.0, REC-xml-19980210, Feb 1998

3 Definitions and abbreviations

3.1 Definitions

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

Bluetooth: a technology specification for short range radio links between mobile PCs, mobile phones and other portable devices. (http: //www.bluetooth.com/)

GET: the operation of requesting that the server returns an object from to the client as defined in the IrDA IrOBEX specification

GSM: Global System for Mobile communications

HTTP: HyperText Transfer Protocol

IrDA: an industry consortium set up to define a set of short range Ir communications standards. (http: //www.irda.org/)

Level 1: minimum level support defined in the IrDA IrMC set of specifications

Level 2: access level support defined in the IrDA IrMC set of specifications

Level 3: index level support defined in the IrDA IrMC set of specifications

Level 4: sync level support defined in the IrDA IrMC set of specifications

MIME: Multipurpose Internet Mail Extension

PUT: the operation of sending one object from the client to the server as defined in the IrDA IrOBEX specification

SSL: Secure Socket Layer

Synchronisation: the process of exchanging information between multiple physical or virtual locations for the purpose of ensuring that each location's copy of that information reflects the same information content

vCalendar: a format defined by the IMC for electronic calendaring and scheduling exchange with extensions as defined in the IrDA IrMC set of specifications

vCard: a format defined by the IMC for electronic business card exchange with extensions as defined in the IrDA IrMC set of specifications

WAP: an industry consortium set up to define a set of standards to empower mobile users with wireless devices to easily access and interact with information and services. (http: //www.wapforum.com/)

Wide Area Network: a geographically-large range wireless connection between two or more devices for the purpose of transferring information. Large geographical range is typically defined as one kilometer or more in distance

3.2 Abbreviations

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

Cookie:	a method of tracking http-based information
IETF	Internet Engineering Task Force
IMC	Internet Mail Consortium
Ir	Infrared
IrDA	Infrared Data Association
IrMC	Ir Mobile Communications
IrOBEX	Ir Object EXchange
OBEX	Object Exchange
PDA	Personal Digital Assistant
PIM	Personal Information Manager
URL:	Universal Resource Location
WAP	Wireless Application Protocol
WML:	Wireless Markup Language
XML:	eXtensible Markup Language

4 Background

4.1 IrMC

The IrMC standard was developed as an extension to the IrDA standard for the purpose of providing an open standard for data exchange between mobile devices or between mobile devices and desktops or PDAs. Among other things, IrMC defines four levels of support for information exchange. By definition, each higher level must support all of the preceding levels. The four levels are: Level 1 (Minimum Level), Level 2 (Access Level), Level 3 (Index Level), and Level 4 (Sync Level). Level 4 does not require Level 3. Level 2 and Level 4 are the most relevant for synchronisation. IrMC has been adopted by IrDA and Bluetooth initiatives and has wide industry support.

4.2 Bluetooth

Bluetooth has adopted the IrMC standard as the basis for their synchronisation specification.

4.3 WAP

WAP has not specified a synchronisation standard. Attempts to form a work group last year were abandoned.

5 IrMC

There are two approaches regarding syncing of a mobile device. Either the logic of the synchronization has to be controlled by the server or by the mobile device. It has to be decided whether the mobile device should be the client or the server in the synchronization process. As the mobile device has a limited amount of memory and limited processing capacity, it is desired to perform as much of the processing as possible outside of the mobile device. In this case the mobile device becomes the server in the synchronization process, only performing the operations the client tells it to perform. This introduces a problem, as the mobile device is an Internet client, and now has to act like a server. How this is solved is explained in chapter 6.2.

To be able to synchronize a mobile device calendar, a set of rules for how to read and write data from and to the mobile device has to be defined. It must also be decided how to keep track of changes done in the mobile device. An existing, and widely spread, standard for this is IrMC. IrMC provides a model for how to store and access data, such as calendar items, contacts and more. IrMC is usually put in the application layer on top of the OBEX layer in an IR stack. The purpose of this document is to describe how to apply IrMC and OBEX on the Internet, using 3GPP. This requires tunneling of OBEX in 3GPP and reversing the client/server roles.

6 Tunnelling of OBEX

There are two major problems with tunneling OBEX over a wide area network.

The first problem is that no logical connection is kept between the client and the server. In the same way that HTTP is stateless, 3GPP only knows a client at one Request/Response-pair at the time. This means that the state awareness of an application has to be implemented by the application.

The second problem is that the client and the server roles are strictly defined. The client always requests the server and never the other way around. To get around this, a protocol has to be defined that emulates the reversion of the roles.

6.1 Introduction of State

The problem with achieving state awareness on the Internet is usually solved by creating a session object on the server that identifies the client by a cookie. Cookies are not yet a standard of 3GPP and also introduce scalability problems on the server side. The option left is to pass a Session Id between the client and the server throughout the session. This solution is widely adopted on the Internet today.

Usually, when state awareness has to be achieved on the Internet, the client is a browser and the Session Id has to be passed back and forth in hidden fields of forms. As the synchronization of a calendar application in a mobile phone is performed by a program and does not involve a browser and no interactivity with the user, a Session Id only has to be passed to the client at initialization of the synchronization process. The client however has to pass the Session Id in every request to be identified by the server.

The Connection Id used in OBEX is a 4-byte number. The Session Id chosen for the synchronization is a 128-bit (16 bytes) number. Preferably this number should be generated as a GUID (Global Unique Identifier) as these numbers are guarantied to be unique.

6.2 Client/Server

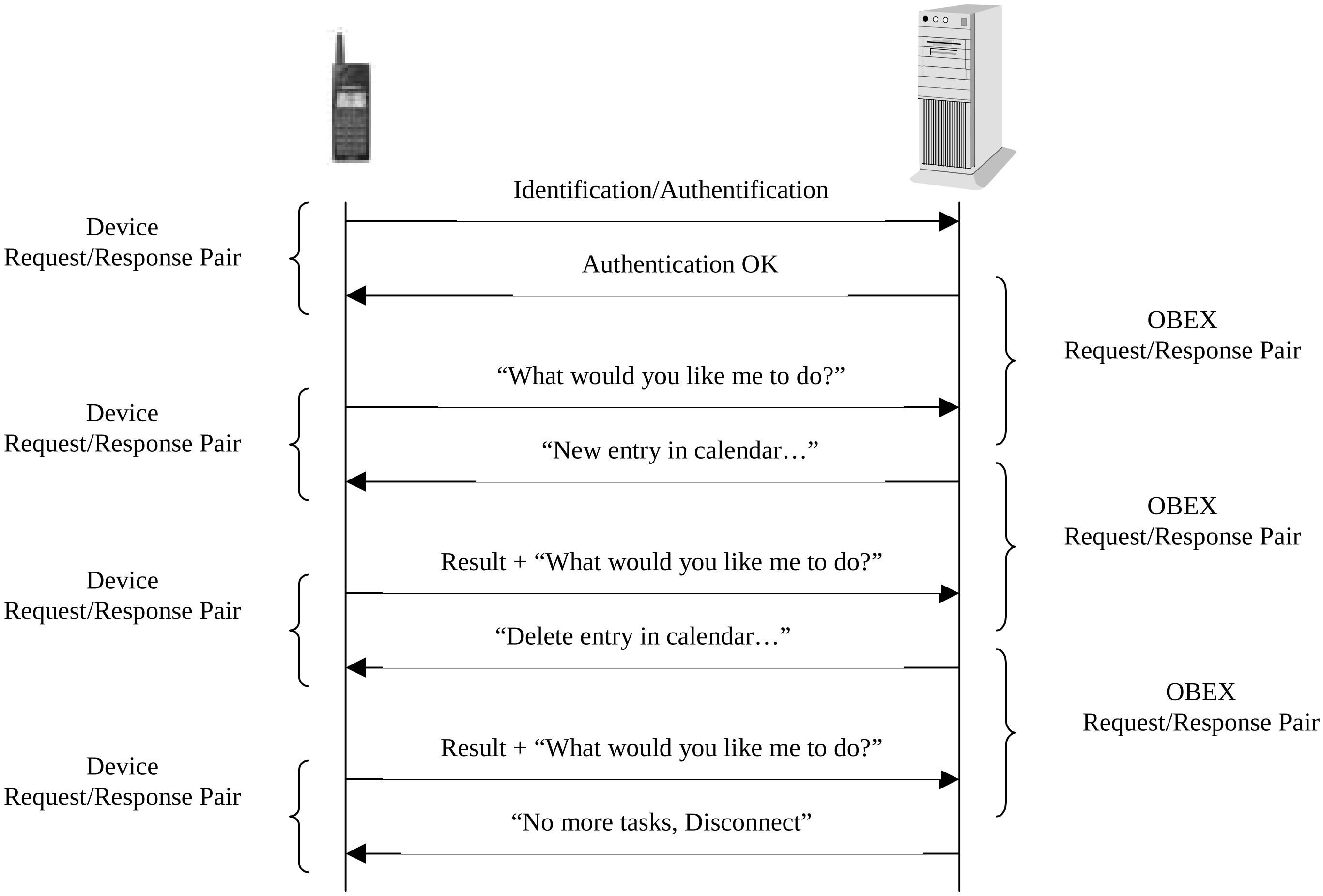
In the case of synchronizing a mobile device with a server’s data, it is preferable to put the synchronization logic on the server side, as the mobile device has limited resources of memory and processing capacity. The synchronization process should thus be controlled by the server. The connection however should be initiated by the client. As the Internet Request/Response model contradicts this, we have to define a way to get around this.

The approach is to let the client (the mobile device) consecutively query the server for what operation it wants to perform on the client. The client will then perform the action and query the server for a new task. This is repeated until the server has no more tasks to perform.

The client will always call the server with two parameters, except for the initial connect request, using the POST method. The reason for using post is that there is a size limit for sending data in the URL, using the GET method. Using the POST method also avoids problems with special characters. The two parameters should be named **sid** and **obex**. The connect request calls the server with one parameter, **userid**, which contains the user name. Every client request implies permission for the server to request a client task in its response.

Name	Size	Description
sid	16 bytes	This is the GUID assigned by the sever. The GUID should be coded as an array of 16 bytes, each byte representing a byte in the GUID. The first byte in the array is MSB.
obex	-	This parameter contains the obex headers sent from the client to the server. The format is pure binary.
userid	-	This parameter contains the user name. The format is plain text.

6.2.1 Overview


Identification/Authentification
Authentication OK
"What would you like me to do?"
"New entry in calendar…"
Result + "What would you like me to do?"
"Delete entry in calendar…"
Result + "What would you like me to do?"
"No more tasks, Disconnect"
Device Request/Response Pair
Device Request/Response Pair
Device Request/Response Pair
Device Request/Response Pair
OBEX Request/Response Pair
OBEX Request/Response Pair
OBEX Request/Response Pair

6.3 Authentication

For the server to authenticate the client (and vice versa) the OBEX Authenticate Challenge/Response procedure, with a few changes, is used.

The client and the server share a secret, i.e. a password. The password is used to generate the message digests, passed between the client and the server in the authentication process. This password is set during a registration process that has taken place before the first synchronisation, for instance on a sign-up web page. The password is then never sent over the internet again, only message digests generated with the password is sent. The hashing algorithm used for the digestion is the MD5 algorithm.

This means that the mobile device must have access to the user name and password, either in memory or from user interaction.

The authentication process is defined by the OBEX Connect procedure, including the OBEX Authenticate Challenge/Response procedure.

6.4 The secure connection

The authentication process only guaranties that the client and the server can rely on each others identity during the connection process. The connection that is established is not secure and could easily be tapped for information. It is therefore desired to encrypt all data that is sent between the client and the server. 3GPP currently does not guarantee strong enough encryption so we will ensure data is secure and untampered.

In the case of a synchronization of a mobile calendar over 3GPP, there are actually two different transports that has to be considered. First it is the transport from the mobile device to the 3GPP gateway. Then there is the transport from the gateway to the web server. The transport from the mobile device to the gateway is sent over GSM, which is fairly well encrypted. The transport from the gateway to the web server is not protected in any way though. To solve this problem we will use a third party product, e.g. “Wireless Jalda”, to establish a protected connection from the gateway to the web server. This should be transparent from the mobile device and set up the required SSL connection.

6.5 Connect

The connect sequence sets up the connection from the mobile device to the web server. The session id has to be assigned in the first response from the server, as more request/response pairs are needed to complete the authentication procedure. The Connect procedure is always invoked by the client.

	Data	Description
Request →	userid=<user name>	The mobile device calls the web server, using the POST method to send the user name.
Response ←	<session id> <obex connect with authenticate challenge>	The web server responds with a 16 byte session id and the obex headers for connect with authenticate challenge.
Request →	sid=<session id> obex=<obex unauthorised with authenticate challenge>	The mobile device responds to the connect request by sending an unauthorised response with authenticate challenge, forcing the web server to authenticate itself.
Response ←	<obex connect with authenticate challenge and authentication response>	The web server verifies the mobile device and authenticates itself.
Request →	sid=<session id> obex=<obex success with authenticate response>	The mobile device verifies the web server and sends an obex success.
Response ←	…	The web server now starts acting like the a client to the mobile device, sending PUT and GET operations to the mobile device.

6.6 Disconnect

Disconnection can either be invoked by the client or be invoked by the server as a last response. The client's session is then destroyed in the server. A third case is that the connection is lost for other reasons, e.g. power failure by the client. In this case, the session should be timed out automatically.

6.6.1 Client disconnection

The client normally should not invoke the disconnection. Should the client however need to disconnect, the following sequence should be used:

	Data	Description
Response ←	…	The web server asks the mobile device to perform some operation.

Request
→
	sid=<session id>
	obex=<obex disconnect>
		The mobile device send an obex disconnect to the web server.

Response ←	-	The web server destroys the session and responds with an empty response.

6.6.2 Server disconnection

When the server is done synchronizing its content, it should disconnect the client. The following sequence should be used:

	Data	Description
Response ←	<obex disconnect>	The web server send an obex disconnect to the mobile device and destroys the session.
	-	The mobile device disconnects and sends no more requests to the web server.

6.7 Put

The PUT operation sends a named vCalendar object from the server to the mobile device. The PUT operation can only be invoked by the web server.

	Data	Description
Response ←	<obex put>	The web server sends a put request to the mobile device.

Request
→
	sid=<session id>
	obex=<obex put response>
		The mobile device performs the put operation and responds with the resulting obex data.

6.8 Get

The GET operation retrieves a named vCalendar object from the mobile device. The GET operation can only be invoked by the web server.

	Data	Description
Response ←	<obex get>	The web server sends a get request to the mobile device.

Request
→
	sid=<session id>
	obex=<obex get response>
		The mobile device performs the get operation and responds with the resulting obex data.

6.9 Timeouts

The operation will wait for N seconds before retry. The timeout will be similar to one used on browsers and implementation dependent.

7 The server side

The server, which is a web server, has to act as described in chapter 6. The functionality should be implemented as a standard web application, with the difference that the delivered content will not be HTML or WML, but OBEX frames to be parsed by the mobile device. There will be no new MIME type for this contents as the information will not be displayed in a browser.

The server also has to maintain state for the application. As 3GPP, at present, does not support cookies, the usual session concept, using cookies, cannot be used. This is solved by letting the server generate and pass a GUID to the client during the authentication procedure. This GUID is stored by the client and passed from the client to the server in every request for the whole session. It is the servers responsibility to map the GUID to the correct client. This is the main discrepancy to the OBEX specification. See chapter 6.

History

Document history		
V3.0.0	January 2000	Publication