Mobile Location Protocol Specification

Abstract

The purpose of this specification is to define a simple and secure access method that allows Internet applications to query location information from a wireless network, irrespective of its underlying air interface technologies and positioning methods.

This specification covers the core of a Mobile Location Protocol that can be used by a location-based application to request MS location information from a location server (GMLC/MPC or other entity in the wireless network).

This specification has been prepared by LIF to provide a simple and secure API (Application Programmer's Interface) to the location server, but that also could be used for other kinds of location servers and entities in the wireless network.

The API is based on existing and well-known Internet technologies as HTTP, SSL/TLS and XML, in order to facilitate the development of location-based applications.

Contents

1 Revision History

1.0	23-Jan-2001	Sanjiv Bhatt, Motorola	Motorola, Nokia, Ericsson contribution to LIF
1.1	26-Jan-2001	Sanjiv Bhatt, Motorola	Updated after review in MLP adhoc committee in LIF #2 meeting
1.1.1	5-Nov-2001	Sanjiv Bhatt, Motorola	Updated after SIG#6 meeting
1.1.2	17-Nov-2001	Sanjiv Bhatt, Motorola	Updated after SIG#7 meeting
2.0.0	20-Nov-2001	Sanjiv Bhatt, Motorola	Final version (public release)

2 Introduction

The Mobile Location Protocol (MLP) is an application-level protocol for getting the position of mobile stations (mobile phones, wireless personal digital assistants, etc.) independent of underlying network technology. The MLP serves as the interface between a Location Server and a Location Services (LCS) Client. This specification defines the core set of operations that a Location Server should be able to perform.

2.1 Abbreviations

ANSI	American National Standards Institute
DTD	Document Type Definition
GMLC	Gateway Mobile Location Center
GMT	Greenwich Mean Time
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP Secure
LCS	Location Services
MLC	Mobile Location Center
MLP	Mobile Location Protocol
MPC	Mobile Positioning Center
MS	Mobile Station
MSID	Mobile Station Identifier
SSL	Secure Socket Layer
TLS	Transport Layer Security
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UTM	Universal Transverse Mercator
WGS	World Geodetic System
XML	Extensible Markup Language

2.2 Notational Conventions and Generic Grammar

The following rules are used throughout this specification to describe basic parsing constructs. ANSI X3.4-1986 defines the US-ASCII coded character set, see ref. [7]

CR	= <US-ASCII CR, carriage return (13)>
LF	= <US-ASCII LF, linefeed (10)>
SP	= <US-ASCII SP, space (32)>

A set of characters enclosed in brackets ([]) is a one-character expression that matches any of the characters in that set. E.g., "[lcs]" matches either an "l", "c", or "s". A range of characters is indicated with a dash. E.g., "[a-z]" matches any lower-case letter.

The one-character expression can be followed by an interval operator, for example [a-zA-Z]{min,max} in which case the one-character expression is repeated at least min and at most max times. E.g., "[a-zA-Z]{2,4}" matches for example the strings "at", "Good", and "biG".

DTD Syntax Notation

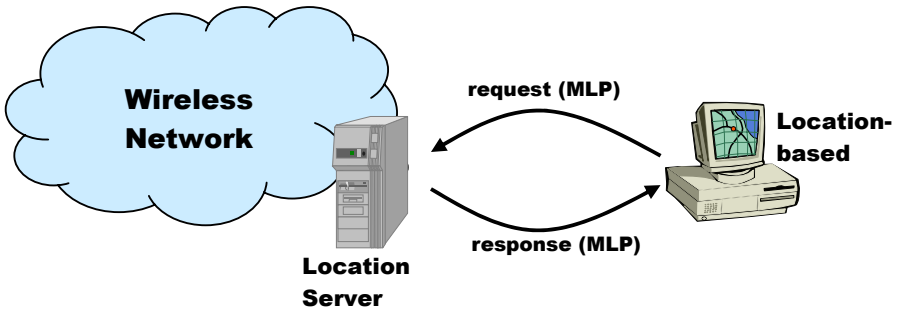
The table below describes the special characters and separators used in the DTDs defining the different services.

Character	Meaning
+	One or more occurrence
*	Zero or more occurrences
?	Optional
()	A group of expressions to be matched together
|	OR...as in, "this or that"
,	Strictly ordered. Like an AND

3 General

3.1 Overview

The Mobile Location Protocol (MLP) is an application-level protocol for querying the position of mobile stations independent of underlying network technology. The MLP serves as the interface between a Location Server and a location-based application.



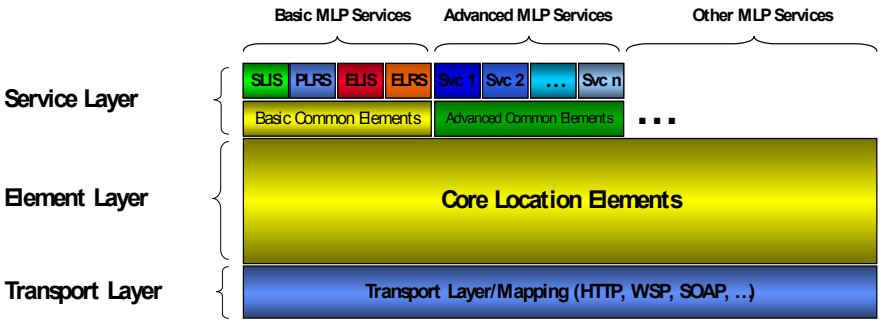
Possible realizations of a Location Server are the GMLC, which is the location server defined in GSM and UMTS, and the MPC, which is defined in ANSI standards. Since the location server should be seen as a logical entity, other implementations are possible.

In the most scenarios (except where explicitly mentioned) an LCS client initiates the dialogue by sending a query to the location server and the server responds to the query.

3.2 MLP structure

In our heterogeneous world, different devices may support different means of communication. A ubiquitous protocol for location services should support different transport mechanisms.

In MLP, the transport protocol is separated from the XML content. The following diagram shows a layered view of MLP.



On the lowest level, the transport protocol defines how XML content is transported. Possible MLP transport protocols include HTTP, WSP, SOAP and others. The HTTP protocol is the only currently defined MLP transport, but others will be defined in the future.

The Element Layer shown in the diagram defines all common elements used by the services in the service layer.

The Service Layer defines the actual services offered by the MLP framework. Basic MLP Services are based on location services defined by 3GPP, and are defined by this specification. The Advanced MLP Services and Other MLP Services are additional services LIF will define in other specifications.

Note: The boxes representing services in the Service Layer may contain more than one message. E.g. SLIS (Standard Location Immediate Service) consists of SLIR (Standard Location Immediate Request) and SLIA (Standard Location Immediate Answer) messages.

The Service Layer is divided into two sub-layers. The topmost defines the services mentioned in the previous paragraph. The lower sub-layer holds common elements used by that group of services.

3.3 Protocol bearer

MLP can be implemented using various transport mechanism as stated in above section. The following describes how to use MLP over the HTTP transport mechanism.

MLP is implemented on top of "HTTP/1.1". HTTP is a request/response protocol involving a server and a client. In the context of MLP, the client is referred to as the LCS Client and the server is the Location Server (GMLC/MPC). For more information about HTTP, refer to http://www.w3.org and ref [3].

The Location Server should provide two socket ports for operation, one for encryption with SSL/TLS and one without. The reason for having one insecure port is that encryption can consume resources, and if the client is in a secure domain there might not be a need for encryption. Applications residing in an insecure domain, i.e. on the Internet, may use the secure port to ensure the security and privacy of the location information.

For further information about SSL/TLS see ref [4].

Two port numbers should be selected and proposed as standard ports for location servers implementing MLP. The ports should be registered by IANA (Internet Assigned Numbers Authority, see ref [6]). Two port numbers are proposed below.

- 700 for secure SSL/TLS transfers

- 701 for insecure transfers

A Location Server can choose to introduce any other socket based or HTTP transparent technology for secure transfers. Any such technology should be provided over a different port than the two mentioned above.

3.4 Location Services

An LCS Client requests a Location Service by issuing an HTTP POST request towards the Location Server. For more information about HTTP POST, see ref. [3]. The request line syntax is shown below.

Request-line: POST *SP* host *SP* HTTP/1.1 *CRLF*

The request must include the entity-header Content-length field as part of the request. The message body of the request should include the XML formatted request and should have the length specified by the LCS Client in the Content-length field.

If the request is a deferred request (triggered or periodic) the result is delivered to the client through an HTTP POST operation issued by the Location Server. This implies that the client must be able to receive HTTP POST requests and be able to give a valid response.

All Location Services are invoked by sending a request using HTTP POST to a certain URI. An example of an URI is shown below.

http://host:port/LocationQueryService/

The response to the invocation of a Location Service is returned using an HTTP response.

If the LCS client requests triggered or periodic reporting of location, the Location Server will return the answer by performing an HTTP POST operation towards the client. The client must specify the URI that the answer should be posted to. This is done in the service request or by having it in the LCS client profile that can be stored in the Location Server.

The answer will be included in the message body and the Content-length entity will be set to the length of the answer.

There are a number of different possible types of location services. Each implementation of location server can select which services it wants/needs to support. The services are described in the table below.

Service	*Description*
Standard Location Immediate Service	This is a standard query service with support for a large set of parameters. This service is used when a single location response is required immediately (within a set time). This service consists of the following messages: • Standard Location Immediate Request • Standard Location Immediate Answer • Standard Location Immediate Report
Emergency Location Immediate Service	This is a service used especially for querying of the location of a mobile subscriber that has initiated an emergency call. The response to this service is required immediately (within a set time). This service consists of the following messages: • Emergency Location Immediate Request • Emergency Location Immediate Answer
Standard Location Reporting Service	This is a service that is used when a mobile subscriber wants an LCS Client to receive the MS location. The position is sent to the LCS Client from the location server. Which application and its address are specified by MS or defined in the location server. This service consists of the following message: • Standard Location Report
Emergency Location Reporting Service	This is a service that is used when the wireless network automatically initiates the positioning at an emergency call. The position and related data is then sent to the emergency application from the location server. Which application and its address are defined in the location server. This service consists of the following message: • Emergency Location Report
Triggered Location Reporting Service	This is a service used when the mobile subscriber's location should be reported at a specific time interval or on the occurrence of a specific event. This service consists of the following messages: • Triggered Location Reporting Request • Triggered Location Reporting Answer • Triggered Location Report • Triggered Location Reporting Stop • Triggered Location Reporting Stop Answer

3.5 Coordinate systems (Informative)

3.5.1 Cartesian coordinates

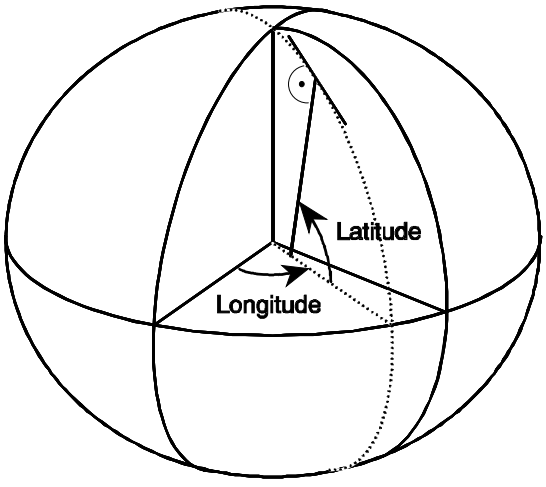
The simplest coordinate system is Cartesian coordinates, defined by values of (**x,y,z**). x is the distance from the x-axis, y is the distance from the y-axis, z the distance from the z-axis.

This coordinate system is primary used for conversions.

3.5.2 Ellipsoid coordinates

Most geographic calculations are based on the surface of the earth. So we need a second coordinate system that describes each position relative to other points and lines on the earth's surface. The irregular shape of the earth is typically approximated by an ellipsoid.

Each point with the Cartesian coordinates (x, y, z) can then be described as set of values (longitude, latitude, altitude) relative to the ellipsoid we choose to describe the earth. The longitude tells us how far east we have to move on the equator from the null-meridian, the latitude tells us how far north to move from the equator and the altitude tells us how far above the ellipsoid to go to finally reach the location. Negative values direct us to go in the opposite direction.



To minimize the difference between the real shape of the earth and the mathematical ellipsoid in different regions, different geographic authorities use

different ellipsoids. All such ellipsoids are defined by a set of 6 parameters which describe the size and the orientation of the ellipsoid in respect of the Cartesian coordinate system. This set is called "Datum". Typical examples are WGS-84 and Bessel-1841

3.5.3 Planar coordinates

Ellipsoid coordinates describe a 3D object (ellipsoid) and are not located on a plane. To draw 2D maps and easily calculate distances, angles or areas the ellipsoid coordinates have to be converted to planar coordinates (x,y).

There are several ways to convert ellipsoid coordinates to planar coordinates:

The simplest is a 1:1 transformation: x=longitude, y=latitude. If the longitude/latitude pairs of the borders of the countries are converted this way to (x,y) and drawn on a paper, we get a typical earth map with x in the range of -180 till +180 and y between -90 till +90.

If the transformation is x=1/180*longitude and y=1/90*latitude the resulting map will be smaller and have both for x and y a range of -1 till 1.

Using trigonometric transformations, we can create other earth maps including various well known maps.

Each transformation system has advantages and disadvantages. Some transformations have a special behaviors suited for particular regions or purposes. If we transform three different points from an ellipsoid to a planar map, the planar points may have the following desirable characteristics:

- Have nearly the same distance to each other as on the ellipsoid. A typical example is the UTM coordinate system
- Have nearly the same angle to each other as on the ellipsoid
- Span the same area as on the ellipsoid.

No planar system supports all these characteristics.

Furthermore, one planar transformation cannot be used for all points of the world. Depending on the transformation, some ellipsoid points cannot be transformed or several ellipsoid points can be mapped to the same planar destination. To solve this problem a rotation is added to the mathematical transformation. A candidate for a modification parameter is for example the longitude divided by a constant factor, for example 6 (UTM) or 15. This is the so called '**zone**'.

3.5.3.1 UTM coordinate system

The UTM coordinate system is a planar coordinate systems based on the Universal Transverse Mercator map projection.

It provides positional descriptions accurate to 1 meter in 2,500 across the entire earth's surface except the poles. At the poles, the Universal Polar Stereographic projection is used. The UTM system divides the earth's surface into a grid in which each cell, excluding overlap with its neighbors, is 6 degrees east to west and 8 degrees north to south (with the exception of the row from 72-84 degrees north latitude).

For any position in the UTM grid, coordinates can be determined in eastings and northings.

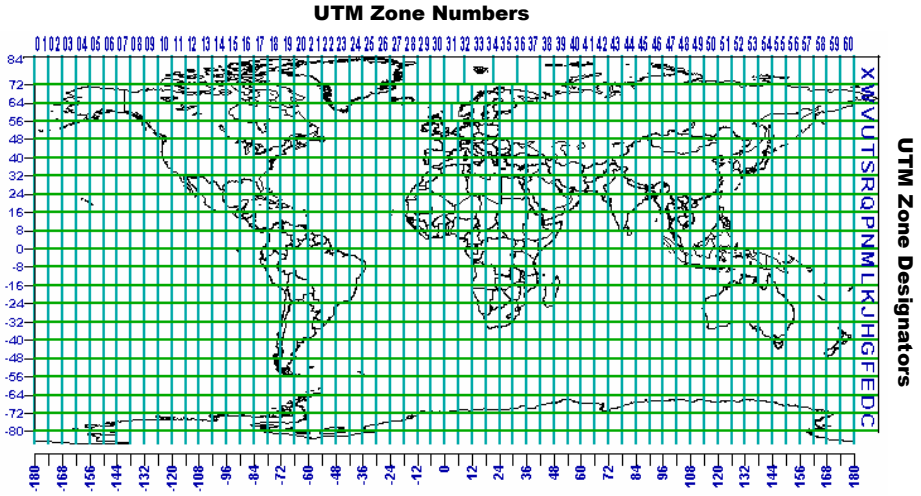
Eastings are in meters with respect to a central meridian drawn through the center of each grid zone (and given an arbitrary easting of 500,000 meters). In the Northern Hemisphere, northings are read in meters from the equator (0 meters). In the Southern Hemisphere, the equator is given the false northing of 10 million meters.

The UTM coordinate system defines two-dimensional, horizontal positions.

In the UTM coordinate system the world is divided into 6-degree longitudal strips extending from 80 degrees South latitude to 84 degrees North latitude. UTM zone characters designate 8-degree zones extending north and south from the equator.

There are special UTM zones between 0 degrees and 36 degrees longitude above 72 degrees latitude and a special zone 32 between 56 degrees and 64 degrees north latitude. Each zone has a central meridian. As an example, zone 14 has a central meridian of 99 degrees west longitude. The zone extends from 96 to 102 degrees west longitude.

Positions are measured in Easting from the central meridian and in Northing from the equator.



3.6 Supported coordinate systems and datum

All MLP implementations support at least the following coordinate systems:

- The ellipsoid coordinate system
- UTM

Although the ellipsoid coordinate system is sufficient for many calculations, MLP supports one additional planar coordinate system: UTM. Using UTM, clients can just ask for rectangular metric coordinates and don't have to worry about complicated coordinate transformations.

All implementations support at least the ellipsoid WGS-84.

Other locally used ellipsoid coordinate systems and datums (such as BESSEL-1841) may additionally be supported by particular implementations.

For each UTM position request, the UTM zone can be preselected. If the zone is not defined in the request the Location Server assigns a zone automatically.

When an application wants to compute the approximate distance between nearby points, it is easy if the planar coordinates are relative to the same zone. The zone with the point closest to the central meridian should be used as this would minimize distortion (unless the points are separated by more than ~6 degrees).

3.7 Conversions between coordinate systems and datum (Informative)

The supported coordinate systems and datums can be converted to other coordinate systems and datums. For example, to change coordinates based on datum WGS-84 to another datum perform the following two steps:

[1] Convert the values (longitude, latitude, altitude) on the WGS-84 ellipsoid to Cartesian coordinates (x,y,z). Well-known mathematical equations do this transformation and typically input the coordinate parameters (longitude, latitude, altitude) and the 6 datum parameters defined by the datum WGS-84.

[2] Transform the Cartesian coordinates from step [1] to the desired datum. The Cartesian coordinates (x,y,z) are independent of all ellipsoids. So these coordinates can be transformed with the inverse mathematical equation from step [1] to coordinates of the new ellipsoid. A typical equation takes as parameters (x,y,z) and the 6 datum parameters defined by the new requested datum. The result is a new set of values for (longitude, latitude, altitude).

To get from there to another coordinate system the ellipsoid points are simply mathematically converted into the new coordinate system, such as Gauss-Krüger. A typical input for these transformations is (longitude, latitude, altitude) and the output Cartesian values (x,y,h). The planar coordinate units are usually metric and may be used to calculate distances, angles or areas depending on the characteristics of the transformation which is used.

The units of Cartesian values depend on the transformation formulas used. If the same type of transformation (i.e. JP19) should be used with different units (ie. 1 meter or 0.1 meter) this can be handled by defining for each combination a transformation (ie. JP19_m and JP19_dm). In this example, if a client selects JP19_dm the client gets JP19 coordinates with the unit 0.1.

3.8 Shapes representing a geographical position

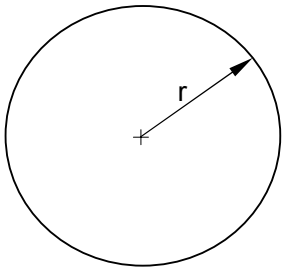
There are a number of shapes used to represent a geographic area that describes where a mobile subscriber is located.

Such shapes can also be used to define triggering criteria that might initiate a positioning when the mobile subscriber enters or leaves the described geographical area. (Such area-based triggering events are not yet defined in the MLP specification.)

3.8.1 Ellipsoid point with uncertainty circle

An ellipsoid point with uncertainty circle is characterized by the coordinates of an ellipsoid point (the origin) and a radius, “r”. It describes the set of points on the ellipsoid, which are at a distance from the point of origin less than or equal to “r”. This shape can be used to indicate points on the Earth surface, or near the Earth surface.

The typical use of this shape is to indicate a point when its position is known only with a limited accuracy.



3.8.2 Ellipsoid point with uncertainty ellipse

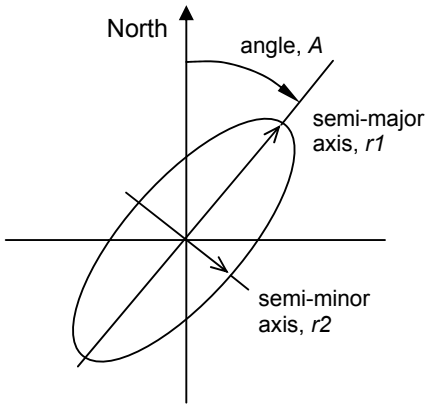
The shape of an "ellipsoid point with uncertainty ellipse" is characterized by the following:

- The coordinates of an ellipsoid point (the origin)
- The distances r1 and r2
- The angle of orientation A

It describes formally the set of points on the ellipsoid, which fall within or on the boundary of an ellipse. This ellipse has a semi-major axis of length r1 oriented at angle A (0 to 180°) measured clockwise from north and a semi-minor axis of length r2. The distances being the geodesic distance over the ellipsoid, i.e., the minimum length of a path staying on the ellipsoid and joining the two points, as shown in figure below.

As for the ellipsoid point, this can be used to indicate points on the Earth's surface, or near the Earth's surface, of same latitude and longitude.

The typical use of this shape is to indicate a point when its position is known only with a limited accuracy, but the geometrical contributions to uncertainty can be quantified.



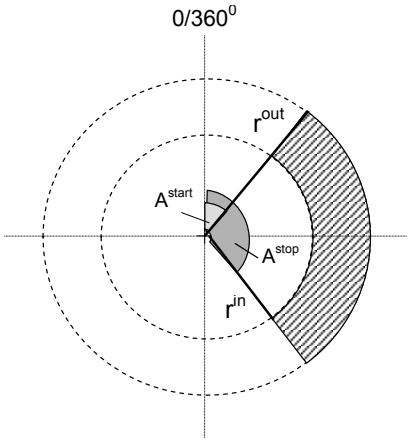
3.8.3 Ellipsoid point with uncertainty arc

The shape of an "ellipsoid point with uncertainty arc" is characterized by the following:

- The coordinates of an ellipsoid point (the origin)
- The inner and outer radius, r^{in} and r^{out}

- The start and stop angles, A^{start} and A^{stop}

An arc is defined by a point of origin with one start and one stop angle plus one inner radius and one outer radius. In this case the striped area describes the actual arc area. The smaller circle defines the inner radius and the outer circle defines the outer radius.



3.8.4 Polygon

A polygon is an arbitrary shape described by an ordered series of points. The minimum number of points allowed is 3, and the maximum number of points allowed is 15. The points shall be connected in the order that they are given. A connecting line is defined as the line over the ellipsoid joining the two points and of minimum distance (geodesic). The last point is connected to the first. The list of points must respect a number of conditions:

- A connecting line shall not cross another connecting line;
- Two successive points must not be diametrically opposed on the ellipsoid.

The described area is situated to the right of the lines with the downward direction being toward the Earth's center and the forward direction being from a point to the next.

NOTE: This definition does not permit connecting lines greater than roughly 20 000 km. If such a need arises, the polygon can be described by adding an intermediate point.

Computation of geodesic lines is not simple. Approximations leading to a maximum distance between the computed line and the geodesic line of less than 3 meters are acceptable.

3.9 MLP extension mechanism

The MLP specification has been designed with extensibility in mind. Examples of design principles employed to achieve this include:

- Separate DTDs for each message element allows new messages to be added.
- Separate DTDs for definitions that are common to all messages, e.g. client address and shapes, so they can be re-used.
- Parameter extension mechanism allowing the addition of additional parameters to existing messages. This mechanism works by specifying an entity parameter referring to an extension DTD. The extension DTD MUST contain another entity parameter '%extension.param' containing the definition of the extension as a string together with the actual parameters being added.

Each extension parameters should have a vendor specific prefix in order to guarantee their uniqueness.

In order to use the extension, the extension DTD has to be explicitly referenced in the XML document.

The Location Server may ignore any extension that is not recognized and process the message as if the extension is not available.

```
<!—truckco_MLP_extension -->

<!ENTITY     % extension param   "truko extension">

<!ELEMENT    Trucko_extension    (#PCDATA)>
```

Example

```
<?xml version = "1.0" ?>
<!DOCTYPE slir SYSTEM "MLP_SLIR_200.DTD" [
  <!ENTITY % extension SYSTEM
        "http://www.truckco.com/truckco_MLP_extension.dtd">
  %extension;
]>
<slir ver="2.0.0">
  ...
    <truckco_extension>
      ...
    </truckco_extension>
</slir>
```

4 Mobile Location Service Definitions

4.1 General

All the different Location Services in the MLP are defined using XML DTDs.

Since the presence of common structured element among the different services, the DTD that defines a single location service, is composed only by the definition of the root element and the inclusion of the necessary common DTD. The MLP is distributed over a set of common DTDs, each acting to define its core element:

MLP_ID.DTD : Identify Element Definitions

MLP_FUNC.DTD : Function Element Definitions

MLP_LOC.DTD : Location Element Definitions

MLP_SHAPE.DTD : Shape Element Definitions

MLP_QoP.DTD : Quality of Position Element Definitions

MLP_GSM_NET.DTD : GSM Network Parameters Element Definitions

MLP_CTXT: Context Element Definitions

The following DTDs describe the different Location Services. All elements and their attributes are described in chapter 5.

4.2 Common Element Definition

4.2.1 Identity Element Definitions

```
<!-- MLP ID -->

<!ELEMENT   msid                                          (#PCDATA)>
<!ATTLIST   msid
            type (MSISDN | IMSI | IMEI | MIN | MDN |      "MSISDN"
            EME_MSID | IPV4 | IPV6)
            enc (ASC | B64 | CRP)                         "ASC">
<!ELEMENT   msid_range                                    (start_msid, stop_msid)>
<!ATTLIST   msid_range
            type (MSISDN | IMSI | IMEI | MIN | MDN |      "MSISDN"
            EME MSID | IPV4 | IPV6)
            enc (ASC | B64 | CRP)                         "ASC">
<!ELEMENT   msids                                         ((msid, session? |
                                                          msid_range)+)>
<!ELEMENT   esrd                                          (#PCDATA)>
<!ATTLIST   esrd
            type (NA)                                     "NA">
<!ELEMENT   esrk                                          (#PCDATA)>
<!ATTLIST   esrk
            type (NA)                                     "NA">
<!ELEMENT   session                                       (#PCDATA)
            session
            Type (APN | dial)                             #REQUIRED>
```

```
<!ELEMENT   start_msid                                  (#PCDATA)>
<!ELEMENT   stop_msid                                   (#PCDATA)>
```

4.2.2 Function Element Definitions

```
<!-- MLP_FUNC -->
<!ELEMENT   add_info                        (#PCDATA)>
<!ELEMENT   eme_event                       (eme_pos+)>
<!ATTLIST   eme event
            eme trigger (EME ORG |          #REQUIRED>
            EME REL)
<!ELEMENT   tlrr_event                      (time | area_change | ms_action)>
<!ELEMENT   ms_action                       EMPTY>
<!ATTLIST   ms_action
            type (MS_AVAIL)                 #REQUIRED>
<!ELEMENT   interval                        (#PCDATA)>
<!ELEMENT   loc_type                        EMPTY>
<!ATTLIST   loc_type
            type (CURRENT | LAST |          "CURRENT">
            CURRENT OR LAST | INITIAL)
<!ELEMENT   poserr                          (result, time)>
<!ELEMENT   prio                            EMPTY>
<!ATTLIST   prio
            type (NORMAL | HIGH)            "NORMAL">
<!ELEMENT   pushaddr                        (url)>
<!ELEMENT   req_id                          (#PCDATA)>
<!ELEMENT   result                          (#PCDATA)>
<!ATTLIST   result
            resid CDATA                     #REQUIRED>
<!ELEMENT   start_time                      (#PCDATA)>
<!ATTLIST   start_time
            utc_off CDATA                   "0000">
<!ELEMENT   stop_time                       (#PCDATA)>
<!ATTLIST   stop_time
            utc_off CDATA                   "0000">
<!ELEMENT   time                            (#PCDATA)>
<!ATTLIST   time
            utc off CDATA                   "0000">
<!ELEMENT   url                             (#PCDATA)>
<!ELEMENT   Time remaining                  (#PCDATA)>
```

4.2.3 Location Element Definitions

```
<!-- MLP_LOC -->
<!ELEMENT   pos                             (msid, (pd | poserr), gsm_net_param?)>
<!ELEMENT   eme_pos                         (msid, esrd?, esrk?, (pd | poserr))>
<!ELEMENT   trl_pos                         (msid, (pd | poserr))>
<!ELEMENT   trl_pos
<!ATTLIST   trl_trigger (TIMER | PERIODIC
            | MS AVAIL)
<!ELEMENT   pd                              (time, shape, (alt, alt acc?)?, speed?,
                                            direction?, lev_conf?)>
<!ELEMENT   alt                             (#PCDATA)>
<!ELEMENT   alt_acc                         (#PCDATA)>
<!ELEMENT   direction                       (#PCDATA)>
<!ELEMENT   easting                         (#PCDATA)>
<!ELEMENT   geo_info                        (coord_sys, datum, format? Zone?)>
<!ELEMENT   coord_sys                       (#PCDATA)
<!ELEMENT   datum                           (#PCDATA)
<!ELEMENT   format                          (#PCDATA)>
<!ELEMENT   lat                             (#PCDATA)>
<!ELEMENT   ll_point                        (lat, long)>
<!ELEMENT   long                            (#PCDATA)>
<!ELEMENT   northing                        (#PCDATA)>
<!ELEMENT   utm_point                       (easting, northing, zone, zone_des)>
<!ELEMENT   speed                           (#PCDATA)>
<!ELEMENT   x                               (#PCDATA)>
<!ELEMENT   xy point                        (x, y)>
<!ELEMENT   y                               (#PCDATA)>
<!ELEMENT   zone                            (#PCDATA)>
<!ELEMENT   zone_des                        (#PCDATA)>
<!ELEMENT   lev_conf                        (#PCDATA)>
```

4.2.4 Shape Element Definitions

```
<!-- MLP_SHAPE -->
<!ELEMENT   shape                (point | circle | circ_arc | ellipse | polygon )>
<!ELEMENT   point                (ll point | utm point | xy point)>
<!ELEMENT   angle                (#PCDATA)>
<!ELEMENT   circ_arc             (point, in_rad, out_rad, start_angle, stop_angle)>
<!ELEMENT   circle               (point, rad)>
<!ELEMENT   ellipse              (point, angle, semi_major, semi_minor)>
<!ELEMENT   in_rad               (#PCDATA)>
<!ELEMENT   out_rad              (#PCDATA)>
<!ELEMENT   polygon              (point+)>
<!ELEMENT   rad                  (#PCDATA)>
<!ELEMENT   semi_major           (#PCDATA)>
<!ELEMENT   semi minor           (#PCDATA)>
<!ELEMENT   start angle          (#PCDATA)>
<!ELEMENT   stop_angle           (#PCDATA)>
```

4.2.5 Quality of Position Element Definitions

```
<!-- MLP_QoP -->
<!ELEMENT   eqop                           (resp_req?,  resp_timer?, (ll_acc | hor_acc)?,
                                           alt_acc?, max_loc_age?)>
<!ELEMENT   qop                            ((ll_acc | hor_acc)?, alt_acc?)>

<!ELEMENT   ll acc                         (#PCDATA)>
<!ELEMENT   hor_acc                        (#PCDATA)>
<!ELEMENT   resp req                       EMPTY>
<!ATTLIST   resp_req
            type (NO_DELAY | LOW_DELAY |   "DELAY_TOL">
            DELAY_TOL)
<!ELEMENT   resp_timer                     (#PCDATA)>
```

4.2.6 Network Parameters Element Definitions

```
<!-- MLP GSM NET -->
<!ELEMENT   gsm net param                  (cgi?, neid?, nmr?, ta?)>
<!ELEMENT   cgi                            (mcc, mnc, lac, cellid)>
<!ELEMENT   neid                           (vmscid | vlrid)>
<!ELEMENT   vmscid                         (cc, ndc, vmscno)>
<!ELEMENT   vlrid                          (cc, ndc, vlrno)>
<!ELEMENT   nmr                            (#PCDATA)>
<!ELEMENT   mcc                            (#PCDATA)>
<!ELEMENT   ndc                            (#PCDATA)>
<!ELEMENT   cc                             (#PCDATA)>
<!ELEMENT   vmscno                         (#PCDATA)>
<!ELEMENT   vlrno                          (#PCDATA)>
<!ELEMENT   lac                            (#PCDATA)>
<!ELEMENT   cellid                         (#PCDATA)>
<!ELEMENT   ta                             (#PCDATA)>
```

4.2.7 Context Element Definitions

```
<!-- MLP_CTXT -->
<!ELEMENT   client                         (id, pwd?, serviceid? servicetype?)>
<!ELEMENT   sessionid                      (#PCDATA)>
<!ELEMENT   id                             (#PCDATA)>
<!ELEMENT   originator                     (id, serviceid?)>
<!ELEMENT   pwd                            (#PCDATA)>
<!ELEMENT   serviceid                      (#PCDATA)>
<!ELEMENT   servicetype                    EMPTY>
<!ATTLIST   servicetype
```

```
            type (ACTIVE | PASSIVE)          "PASSIVE">
<!ELEMENT   subclient                        (id, pwd?, serviceid?)>
<!ATTLIST   subsclient
            last_client (YES | NO)           "NO">
```

4.3 Request Header Components

The service has two main parts, namely a context or header part and a body part. The body part is described in the sections 4.4 - 4.8. The context or header part consists of the information that identifies the client as defined in this section.

The SUBCLIENT elements, if present, identify the ASPs, resellers and portals in the chain of service providers between the network and the end-user. The distinction between CLIENT and SUBCLIENT elements is that the CLIENT element identifies the provider of the service that the network has the initial relationship with, whereas the SUBCLIENT elements identify the chain of other service providers up to the end-user. The final service provider in the chain is identified as such. On the other hand ORIGINATOR is indicating the initiator of the location request, so in this context besides an ASP it could also be an MS subscriber who is asking the position of another target MS. The identity of the ORIGINATOR may be an MSISDN or any other identifier identifying the initiator of the location request.

4.3.1 Context DTD

```
<!-- MLP_HDR -->

<!ELEMENT    hdr              ((client | sessionid), subclient*, originator?)>
<!ATTLIST    hdr
             ver CDATA        #FIXED "1.0">

<!ENTITY     % mlp_ctxt.dtd   SYSTEM "MLP_CTXT_200.DTD">

%mlp_ctxt.dtd;
```

4.3.1.1 Example (ASP as Originator)

```
<?xml version="1.0" ?>
<!DOCTYPE hdr SYSTEM "MLP_HDR_200.DTD">
<hdr ver="2.0.0">
  <client>
    <id>theasp</id>
    <pwd>thepwd</pwd>
    <serviceid>0005</serviceid>
    <servicetype type="PASSIVE"/>
  </client>
  <originator>
    <id>theoriginalasp</id>
    <serviceid>0003</serviceid>
  </originator>
  <subclient last_client="YES">
    <id>thelastasp</id>
    <serviceid>0007</serviceid>
  </subclient>
</hdr>
```

4.3.1.2 Example (MS as Originator)

```
<?xml version="1.0" ?>
<!DOCTYPE hdr SYSTEM "MLP_HDR_200.DTD">
<hdr ver="2.0.0">
  <client>
    <id>theasp</id>
    <pwd>thepwd</pwd>
    <serviceid>0005</serviceid>
    <servicetype type="ACTIVE"/>
  </client>
  <originator>
    <id>461018765710</id>
  </originator>
</hdr>
```

4.4 Standard Location Immediate Service

This is a standard service for requesting the location of one or more Mobile Subscribers. The service is used when a single location response is required immediately (within a set time). An immediate request should be used when the LCS Client wants to receive the answer to the request over a persistent connection.

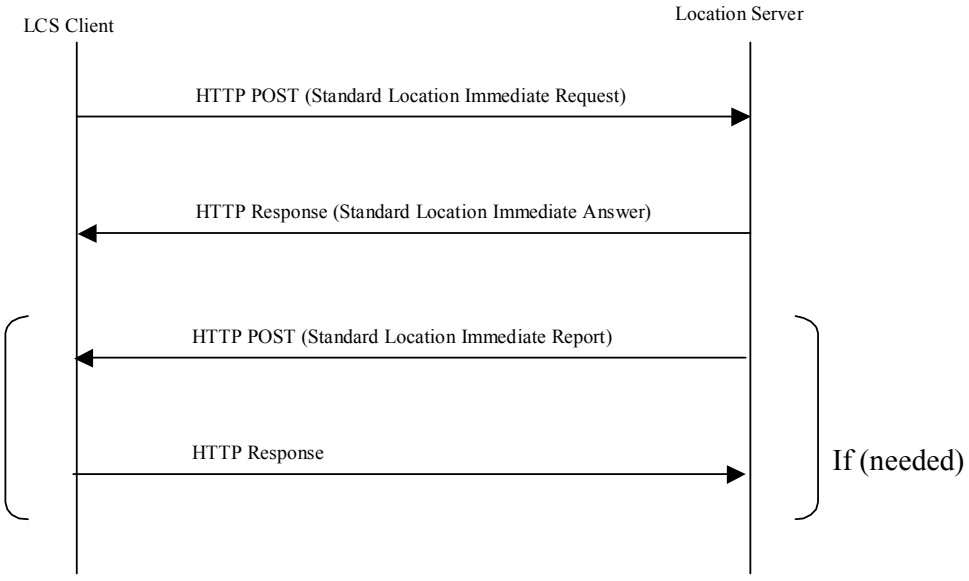
When a lot of positioning reports are requested, it may take an unacceptably long time to get the all responses from the network. If the Location Server supports it the LCS Client can how to receive the responses, either at a time using a persistent connection, or individually using one or more connections initiated by the Location Server.

The extended service supports a number of different formats for describing the location of the mobile subscriber. It has also support for requesting a certain Quality of Service, Type of location and priority.

The service consists of the following messages:

- Standard Location Immediate Request
- Standard Location Immediate Answer
- Standard Location Immediate Report

The following HTTP message flow encapsulates this service:



4.4.1 Standard Location Immediate Request DTD

```
<!--MLP_SLIR-->
<!ENTITY     % extension               "">
<!ENTITY     % extension.param         "">

<!ELEMENT    slir                      (client, (msids | (msid, gsm_net_param)+), eqop?,
                                       geo_info?, loc_type?, prio?, pushaddr?
                                       %extension.param;)>
<!ATTLIST    slir
             ver CDATA                 #FIXED "1.0"
             res_types                 "PERSISTENT">
             (PERSISTENT | PUSH)

<!ENTITY     % mlp loc.dtd             SYSTEM "MLP LOC 200.DTD">
<!ENTITY     % mlp_id.dtd              SYSTEM "MLP_ID_200.DTD">
<!ENTITY     % mlp_func.dtd            SYSTEM "MLP_FUNC_200.DTD">
<!ENTITY     % mlp_qop.dtd             SYSTEM "MLP_QoP_200.DTD">
<!ENTITY     % mlp_gsm_net_param.dtd   SYSTEM "MLP_GSM_NET_PARAM_200.DTD">

%mlp_loc.dtd;
%mlp id.dtd;
%mlp func.dtd;
%mlp qop.dtd;
%mlp gsm net_param.dtd
```

4.4.1.1 Example

```
<?xml version="1.0" ?>
<!DOCTYPE slir SYSTEM "MLP_SLIR_200.DTD"[
  <!ENTITY % extension SYSTEM "http://truckco.com/some_MLP_extension.dtd">
  %extension;
]>
<slir ver="2.0.0" res_type="PERSISTENT">
  <msids>
    <msid type="IPV4">93.10.0.250</msid>
    <msid_range>
      <start_msid>461018765710</start_msid>
      <stop_msid>461018765712</stop_msid>
    </msid_range>
  </msids>
  <eqop>
    <resp_req type="LOW_DELAY" />
    <hor_acc>1000</hor_acc>
  </eqop>
  <geo_info>
     <format>IDMS3</format>
  </geo_info>
  <loc_type type="CURRENT_OR_LAST" />
  <prio type="HIGH" />
  <truckco_send_sms>
    You have been positioned by The Truck Company
  </truckco_send_sms>
</slir>
```

4.4.2 Standard Location Immediate Answer DTD

```
<!-- MLP SLIA -->
<!ENTITY     % extension          "">
<!ENTITY     % extension.param    "">

<!ELEMENT    slia                 ((pos+ | req_id), result, add_info? %extension.param;)>
<!ATTLIST    slia
             ver CDATA            #FIXED "1.0">
             res_type             "PERSISTENT">
             (PERSISTENT |
             PUSH)

<!ENTITY     % mlp_shape.dtd      SYSTEM "MLP_SHAPE_200.DTD">
<!ENTITY     % mlp_loc.dtd        SYSTEM "MLP_LOC_200.DTD">
<!ENTITY     % mlp_id.dtd         SYSTEM "MLP_ID_200.DTD">
<!ENTITY     % mlp_func.dtd       SYSTEM "MLP_FUNC_200.DTD">

%mlp_shape.dtd;
%mlp_loc.dtd;
%mlp_id.dtd;
%mlp_func.dtd;
```

4.4.2.1 Example

```
<?xml version="1.0" ?>
<!DOCTYPE slia SYSTEM "MLP_SLIA_200.DTD">
<slia ver="2.0.0" res_type= "PERSISTENT">
  <pos>
    <msid>461011334411</msid>
    <pd>
      <time utc_off="+0200">20000623134453</time>
      <shape>
        <circle>
          <point>
            <ll_point>
              <lat>301628.312</lat>
              <long>451533.431</long>
            </ll_point>
          </point>
          <rad>240</rad>
        </circle>
      </shape>
    </pd>
  </pos>
  <pos>
    <msid>461018765710</msid>
    <pd>
      <time utc_off="+0300">20000623134454</time>
      <shape>
        <circle>
          <point>
            <ll_point>
              <lat>301228.302</lat>
              <long>865633.863</long>
            </ll_point>
          </point>
          <rad>570</rad>
        </circle>
```

```
      </shape>
    </pd>
  </pos>
  <pos>
    <msid>461018765711</msid>
    <pd>
      <time utc_off="+0300">20000623110205</time>
      <shape>
        <circle>
          <point>
            <ll_point>
              <lat>781234.322</lat>
              <long>762162.823</long>
            </ll_point>
          </point>
          <rad>15</rad>
        </circle>
      </shape>
    </pd>
  </pos>
  <pos>
    <msid>461018765712</msid>
    <poserr>
      <result resid="10">QOP NOT ATTAINABLE</result>
      <time>20000623134454</time>
    </poserr>
  </pos>
  <result resid="0">OK</result>
</slia>
```

4.4.3 Standard Location Immediate Report DTD

```
<!-- MLP_SLIREP -->
<!ENTITY     % extension          "">
<!ENTITY     % extension.param    "">

<!ELEMENT    slirep               (req_id, pos+, utc_off, result, add_info?
                                  %extension.param;)>
<!ATTLIST    slirep
             ver CDATA            #FIXED "1.0">

<!ENTITY     % mlp_shape.dtd      SYSTEM "MLP_SHAPE_200.DTD">
<!ENTITY     % mlp_loc.dtd        SYSTEM "MLP_LOC_200.DTD">
<!ENTITY     % mlp_id.dtd         SYSTEM "MLP_ID_200.DTD">
<!ENTITY     % mlp_func.dtd       SYSTEM "MLP_FUNC_200.DTD">

%mlp_shape.dtd;
%mlp_loc.dtd;
%mlp_id.dtd;
%mlp_func.dtd;
```

4.4.3.1 Example

```
<?xml version="1.0" ?>
<!DOCTYPE slirep SYSTEM "MLP_SLIREP_200.DTD">
<slirep ver="2.0.0">
  <req_id>25267</req_id>
  <pos>
    <msid type="IPV6">10:A1:45::23:B7:89</msid>
    <pd>
      <time utc_off="+0300">20000813010423</time>
      <shape>
        <circle>
          <point>
```

```
            <ll_point>
              <lat>301628.312</lat>
              <long>451533.431</long>
            </ll_point>
          </point>
          <rad>15</rad>
        </circle>
      </shape>
    </pd>
  </pos>
  <result resid="0">OK</result>
</slirep>
```

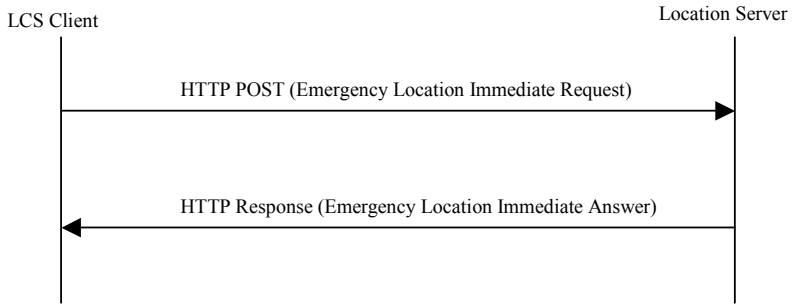
4.5 Emergency Location Immediate Service

The emergency location immediate service is used to retrieve the position of a mobile subscriber that is involved in an emergency call or have initiated an emergency service in some other way.

The service consists of the following messages:

- Emergency Location Immediate Request
- Emergency Location Immediate Answer

The following HTTP message flow encapsulates this service:



4.5.1 Emergency Location Immediate Request DTD

```
<!-- MLP_EME_LIR -->
<!ENTITY     % extension               "">
<!ENTITY     % extension.param         "">

<!ELEMENT    eme_lir                   ((msids | (msid, gsm_net_param)+),, qop?,
                                       geo_info?, loc_type? %extension.param;)>
<!ATTLIST    eme_lir
             ver CDATA                 #FIXED "1.0">

<!ENTITY     % mlp_id.dtd              SYSTEM "MLP_ID_200.DTD">
<!ENTITY     % mlp_loc.dtd             SYSTEM "MLP_LOC_200.DTD">
<!ENTITY     % mlp_func.dtd            SYSTEM "MLP_FUNC_200.DTD">
<!ENTITY     % mlp_qop.dtd             SYSTEM "MLP_QoP_200.DTD">
<!ENTITY     % mlp_gsm_net_param.dtd   SYSTEM "MLP_GSM_NET_PARAM_200.DTD">

%mlp_id.dtd;
%mlp_loc.dtd;
%mlp_func.dtd;
%mlp_qop.dtd;
%mlp_gsm_net_param.dtd;
```

4.5.1.1 Example

```
<?xml version="1.0" ?>
<!DOCTYPE eme_lir SYSTEM "MLP_EME_LIR_200.DTD">
<eme_lir ver="2.0.0">
  <msids>
    <msid type="EME_MSID">520002-51-431172-6-06</msid>
    <esrk>7839298236</esrk>
  </msids>
  <geo_info>
     <format>IDMS3</format>
  </geo_info>
  <loc_type type="CURRENT_OR_LAST" />
</eme_lir>
```

4.5.2 Emergency Location Immediate Answer DTD

```
<!-- MLP_EME_LIA -->
<!ENTITY     % extension          "">
<!ENTITY     % extension.param    "">

<!ELEMENT    eme_lia              (eme_pos+, result, add_info? %extension.param;)>
<!ATTLIST    eme lia
             ver CDATA            #FIXED "1.0">

<!ENTITY     % mlp shape.dtd      SYSTEM "MLP SHAPE 200.DTD">
<!ENTITY     % mlp_id.dtd         SYSTEM "MLP_ID_200.DTD">
<!ENTITY     % mlp_loc.dtd        SYSTEM "MLP_LOC_200.DTD">
<!ENTITY     % mlp_func.dtd       SYSTEM "MLP_FUNC_200.DTD">

%mlp_shape.dtd;
%mlp_id.dtd;
%mlp_loc.dtd;
%mlp func.dtd;
```

4.5.2.1 Example

```
<?xml version="1.0" ?>
<!DOCTYPE eme_lia SYSTEM "MLP_EME_LIA_200.DTD">
<eme_lia ver="2.0.0">
  <eme_pos>
    <msid type="EME_MSID">520002-51-431172-6-06</msid>
    <esrk>7839298236</esrk>
    <pd>
      <time utc_off="+0300">20000623134453</time>
      <shape>
        <circle>
          <point>
            <ll_point>
              <lat>N301628.312</lat>
              <long>W451533.431</long>
            </ll_point>
          </point>
          <rad>20</rad>
        </circle>
      </shape>
    </pd>
  </eme_pos>
  <result resid="0">OK</result>
</eme_lia>
```

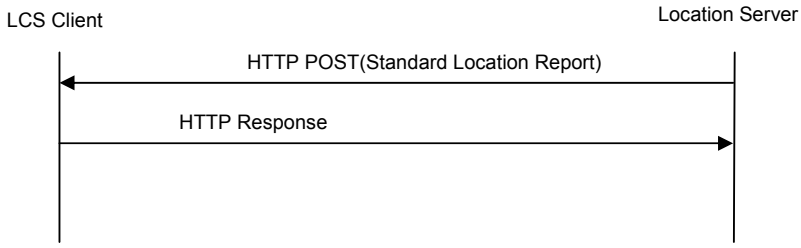
4.6 Standard Location Reporting Service

When a mobile subscriber wants an LCS client to receive the MS location a standard location report is generated. The LCS Client that the location report should be sent to is specified by MS or defined within the Location Server.

The service consists of the following message:

- Standard Location Report

The following HTTP message flow encapsulates this service:



4.6.1 Standard Location Report DTD

```
<!-- MLP_SLREP -->
<!ENTITY     % extension          "">
<!ENTITY     % extension.param    "">

<!ELEMENT    slrep                (result, pos+, add_info? %extension.param;)>
<!ATTLIST    slrep
             ver CDATA            #FIXED "1.0">

<!ENTITY     % mlp shape.dtd      SYSTEM "MLP SHAPE 200.DTD">
<!ENTITY     % mlp_loc.dtd        SYSTEM "MLP_LOC_200.DTD">
<!ENTITY     % mlp_id.dtd         SYSTEM "MLP_ID_200.DTD">
<!ENTITY     % mlp_func.dtd       SYSTEM "MLP_FUNC_200.DTD">

%mlp_shape.dtd;
%mlp_loc.dtd;
%mlp_id.dtd;
%mlp func.dtd;
```

4.6.1.1 Example

```
<?xml version="1.0" ?>
<!DOCTYPE slrep SYSTEM "MLP_SLREP_200.DTD">
<slrep ver="2.0.0">
  <result resid="0">OK</result>
  <pos>
    <msid>461011678298</msid>
    <pd>
      <time>20000813010423</time>
      <shape>
        <circle>
          <point>
            <ll_point>
              <lat>301628.312</lat>
              <long>451533.431</long>
            </ll_point>
          </point>
```

```
            <rad>15</rad>
          </circle>
        </shape>
      </pd>
    </pos>
</slrep>
```

4.7 Emergency Location Reporting Service

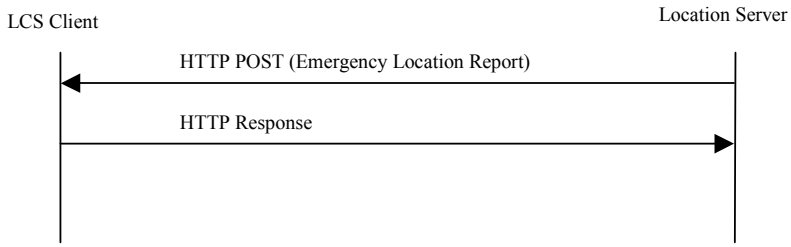
If the wireless network spontaneously initiates a positioning when a user initiates or releases an emergency call, an emergency location report is generated. The application(s) that the emergency location report should be sent to is defined within the location server. Data as required geographical format and address to application is also defined within the location server.

This is the only case where location server initiate the dialogue instead of LCS client.

The service consists of the following message:

- Emergency Location Report

The following HTTP message flow encapsulates this service:



4.7.1 Emergency Location Report DTD

```
<!-- MLP EMEREP -->
<!ENTITY          % extension          "">
<!ENTITY          % extension.param    "">

<!ELEMENT         emerep               (result, eme_event, add_info? %extension.param;)>
<!ATTLIST         emerep
                  ver CDATA            #FIXED "1.0">

<!ENTITY          % mlp_shape.dtd      SYSTEM "MLP_SHAPE_200.DTD">
<!ENTITY          % mlp_loc.dtd        SYSTEM "MLP_LOC_200.DTD">
<!ENTITY          % mlp id.dtd         SYSTEM "MLP ID 200.DTD">
<!ENTITY          % mlp func.dtd       SYSTEM "MLP FUNC 200.DTD">

%mlp_shape.dtd;
%mlp_loc.dtd;
%mlp_id.dtd;
%mlp_func.dtd;
```

4.7.1.1 Example

```
<?xml version="1.0" ?>
<!DOCTYPE emerep SYSTEM "MLP_EMEREP_200.DTD">
<emerep ver="2.0.0">
  <id>LocApplID</id>
  <pwd>LocApplPW</pwd>
  <result resid="0">OK</result>
  <eme_event eme_trigger="EME_ORG">
    <eme_pos>
      <msid>461011678298</msid>
      <pd>
        <time utc_off="+0300">20000623010003</time>
        <shape>
          <circle>
            <point>
              <ll_point>
                <lat>301628.312</lat>
                <long>451533.431</long>
              </ll_point>
            </point>
            <rad>15</rad>
          </circle>
        </shape>
      </pd>
    </eme_pos>
  </eme_event>
</emerep>
```

4.8 Triggered Location Reporting Service

The triggered location reporting service is used when an application wants the position of the a list of MS to be tracked. The triggers could be:

- The periodicity time defined by an interval
- An MS action, defined as the event "UE available" in 3GPP TS 23.271 rel. 4 [ref. 11].

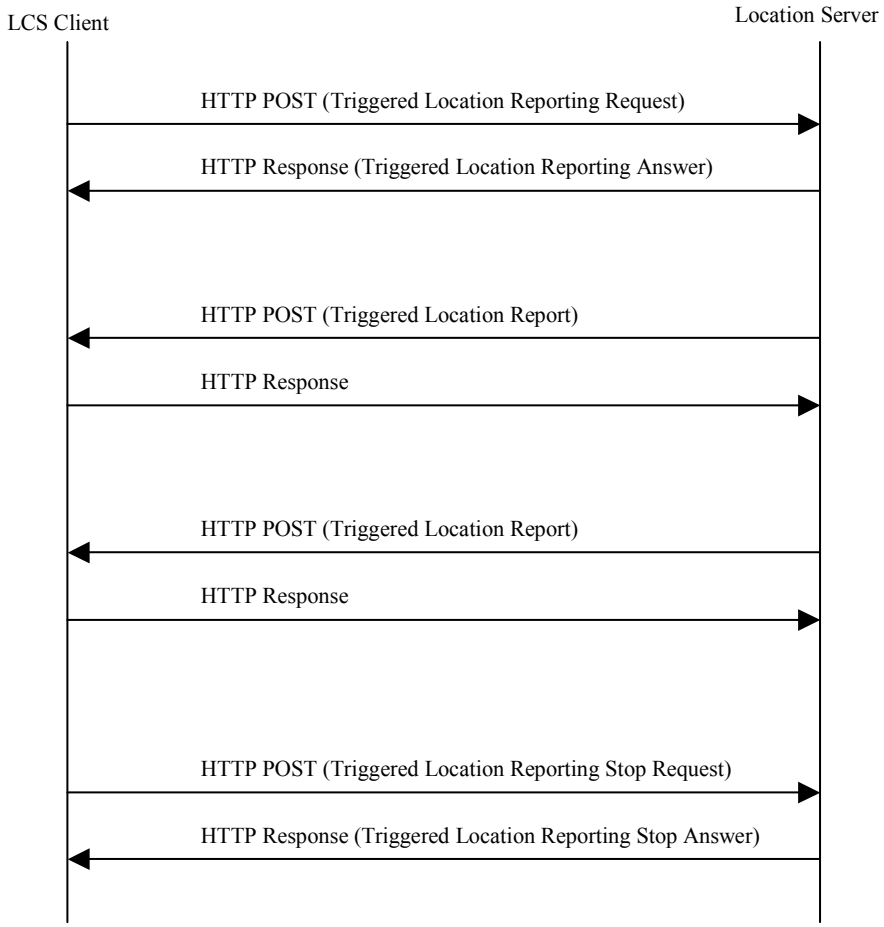
The report will be triggered when one of the pre-defined MS's action is occurred or the time interval elapses.

The service consists of the following messages:

- Triggered Location Reporting Request
- Triggered Location Reporting Answer
- Triggered Location Report
- Triggered Location Reporting Stop
- Triggered Location Reporting Stop Answer

Note:It is the intention that Triggered services will support entering or leaving an area in future releases. An area may be defined as a specified geographical area, a city or locale, a country or a network. Other triggers that may be supported are specific events not yet defined, such a subscriber being in proximity to a friend in a FriendFinder application. Other events are FFS within 3GPP and are targeted for rel.5.

The following HTTP message flow encapsulates this service:



4.8.1 Triggered Location Reporting Request DTD

```
<!-- MLP_TLRR -->
<!ENTITY          % extension          "">
<!ENTITY          % extension.param    "">

<!ELEMENT         tlrr                 (msids, interval, start_time?, stop_time?,
                                        tlrr_event?, qop?, geo_info?, pushaddr?,
                                        loc_type?, prio? %extension.param;)>
<!ATTLIST         tlrr
                  ver CDATA            #FIXED "1.0">

<!ENTITY          % mlp_id.dtd         SYSTEM "MLP_ID_200.DTD">
<!ENTITY          % mlp_loc.dtd        SYSTEM "MLP_LOC_200.DTD">
<!ENTITY          % mlp_func.dtd       SYSTEM "MLP_FUNC_200.DTD">
<!ENTITY          % mlp_qop.dtd        SYSTEM "MLP_QoP_200.DTD">

%mlp_id.dtd;
%mlp_loc.dtd;
%mlp_func.dtd;
%mlp_qop.dtd;
```

The following rules apply to the use of 'start_time' and stop_time':

- If no START_TIME is specified reporting starts immediately.
- If no STOP_TIME is specified the reporting will occur until explicitly canceled with 'Triggered Location Stop Request' or a time out occurs (depending on system configuration).
- If START_TIME is 'older' than current time the Location Server MUST reject the request with an error indication '110' to the client.
- If STOP_TIME is 'older' than current time the Location Server MUST reject the request with an error indication '110' to the client.
- If STOP_TIME is earlier than START_TIME the implementation MUST reject the request with an error indication '110' to the client.

4.8.1.1 Example

```
<?xml version="1.0" ?>
<!DOCTYPE tlrr SYSTEM "MLP_TLRR_200.DTD">
<tlrr ver="2.0.0">
  <msids>
    <msid>461011678298</msid>
  </msids>
  <interval>00003000</interval>
  <start_time offset="+0300">20011003112700</start_time>
  <stop_time offset="+0300">20011003152700</stop_time>
  <qop>
    <hor_acc>100</hor_acc>
  </qop>
  <geo_infocoord_sys="LL" datum="WGS-84">
     <format>IDMS3</format>
  </geo_info>
  <pushaddr>
    <url>http://location.application.com</url>
    <id>LocApplID</id>
    <pwd>LocApplPW</pwd>
  </pushaddr>
  <loc_type type="CURRENT" />
  <prio type="HIGH" />
</tlrr>
```

4.8.2 Triggered Location Reporting Answer DTD

```
<!-- MLP_TLRA -->

<!ENTITY     % extension          "">
<!ENTITY     % extension.param    "">
<!ELEMENT    tlra                 (result, req_id?, add_info? %extension.param;)>
<!ATTLIST    tlra
             ver CDATA            #FIXED "1.0">

<!ENTITY     % mlp func.dtd       SYSTEM "MLP FUNC 200.DTD">

%mlp func.dtd;
```

4.8.2.1 Example

```
<?xml version="1.0" ?>
<!DOCTYPE plra SYSTEM "MLP_TLRA_200.DTD">
<tlra ver="2.0.0">
  <result resid="0">OK</result>
  <req_id>25293</req_id>
</tlra>
```

4.8.3 Triggered Location Report DTD

```
<!-- MLP_TLREP -->
<!ENTITY          % extension         "">
<!ENTITY          % extension.param   "">

<!ELEMENT         tlrep               (req_id, (pos + trl pos)+, time_remaining?, result,
                                      add_info? %extension.param;)>
<!ATTLIST         tlrep
                  ver CDATA           #FIXED "1.0">

<!ENTITY          % mlp shape.dtd     SYSTEM "MLP SHAPE 200.DTD">
<!ENTITY          % mlp loc.dtd       SYSTEM "MLP LOC 200.DTD">
<!ENTITY          % mlp_id.dtd        SYSTEM "MLP_ID_200.DTD">
<!ENTITY          % mlp_func.dtd      SYSTEM "MLP_FUNC_200.DTD">

%mlp_shape.dtd;
%mlp_loc.dtd;
%mlp_id.dtd;
%mlp_func.dtd;
```

4.8.3.1 Example

```
<?xml version="1.0" ?>
<!DOCTYPE tlrep SYSTEM "MLP_TLREP_200.DTD">
<plrep ver="2.0.0">
  <req_id>25267</req_id>
  <id>ServerID</id>
  <pwd>ServerPW</pwd>
  <trl pos trl_trigger="TIMER">
    <msid>461011678298</msid>
    <pd>
      <time utc_off="+0300">20000813010423</time>
      <shape>
        <circle>
          <point>
            <ll_point>
              <lat>301628.312</lat>
              <long>451533.431</long>
            </ll_point>
          </point>
          <rad>15</rad>
        </circle>
      </shape>
    </pd>
  </pos>
  <time_remaining>00010000/time_remaining>
```

```
  <result resid="0">OK</result>
</tlrep>
```

4.8.4 Triggered Location Reporting Stop Request DTD

```
<!-- MLP_TLRSR -->

<!ENTITY          % extension           "">
<!ENTITY          % extension.param     "">
<!ELEMENT         tlrsr                 (req id %extension.param;)>
<!ATTLIST         tlrsr
                  ver CDATA             #FIXED "1.0">

<!ENTITY          % mlp_func.dtd        SYSTEM "MLP_FUNC_200.DTD">

%mlp_func.dtd;
```

4.8.4.1 Example

```
<?xml version="1.0" ?>
<!DOCTYPE plrsr SYSTEM "MLP_TLRSR_200.DTD">
<tlrsr ver="2.0.0">
  <req_id>25293</req_id>
</tlrsr>
```

4.8.5 Triggered Location Reporting Stop Answer DTD

```
<!-- MLP_TLRSA -->
<!ENTITY          % extension         "">
<!ENTITY          % extension.param   "">

<!ELEMENT         tlrsa               (req_id, result, add_info? %extension, param;)>
<!ATTLIST         tlrsa
                  ver CDATA           #FIXED "1.0">

<!ENTITY          % mlp_func.dtd      SYSTEM "MLP_FUNC_200.DTD">

%mlp_func.dtd;
```

4.8.5.1 Example

```
<?xml version="1.0" ?>
<!DOCTYPE plsra SYSTEM "MLP_TLRSA_200.DTD">
<tlrsa ver="2.0.0">
  <req_id>25293</req_id>
  <result resid="0">OK</result>
</tlrsa>
```

5 Elements and attributes in DTD

5.1 add_info

Description:	
A text string containing additional information about a certain result.	
Type:	Element
Format:	Char string
Defined values:	-
Default value:	-
Example:	<add_info>EVENT</add_info>
Note: -	

5.2 alt

Description:	
The altitude of the MS in meters in respect of the ellipsoid which is used to be define the coordinates	
Type:	Element
Format:	Char String
Defined values:	[+|-] [0-9]+
Default value:	-
Example:	<alt>1200</alt>
Note: This element is present if altitude is possible to attain by the used positioning method.	

5.3 alt_acc

Description:	
Accuracy of altitude in meters	
Type:	Element
Format:	Char String
Defined values:	[0-9]+
Default value:	-
Example:	<alt_acc>200</alt_acc>
Note: -	

5.4 angle

Description:		
Specifies the angle of rotation of an ellipse measured clockwise from north.		
Type:	Element	
Format:	Char String	
Defined values:	0-360	
Default value:	-	
Example:	`<angle>60</angle>`	
Note: -		

5.5 cc

Description:	
Specifies the country code.	
Type:	Element
Format:	Char String
Defined values:	2 digits e.g. 44 for UK
Default value:	-
Example:	`<cc>44</cc>`
Note: This element is present if direction is possible to attain by the used positioning method.	

5.6 cellid

Description:	
Identifies the Cell Identity	
Type:	Element
Format:	Char String
Defined values:	0-65535
Default value:	-
Example:	`<cellid>546</cellid>`
Note: This element is present if direction is possible to attain by the used positioning method.	

5.7 coord_sys

Description:		
Specifies which coordinate system that should be used in the position answer		
Type:	Attribute	
Format:	Char string	
Defined values:	`LL`	Longitude Latitude
	`UTM`	Universal Transverse Mercator
Default value:		
Example:	`<coord_sys>"UTM"</coord_sys>`	
Note: -		

5.8 direction

Description:		
Specifies the direction, in degrees, that a positioned MS is moving in.		
Type:	Element	
Format:	Char String	
Defined values:	0-360	
Default value:	-	
Example:	<direction>120</direction>	
Note: This element is present if direction is possible to attain by the used positioning method.		

5.9 datum

Description:		
Specifies a geodetic datum		
Type:	Attribute	
Format:	Char string	
Defined values:	WGS-84	World Geodetic System 1984
Default value:	-	
Example:	<datum>"WGS-84"</datum>	
Note: WGS-84 is the only mandatory datum supported in all implementations , however regional and implementation specific defined elements (such as BESSEL-1841) may additionally be supported.		

5.10 easting

Description:	
Used in the UTM coordinate system. Eastings are measured from central meridian. The number of decimals provided in the response is defined by UTM_FORMAT that is provided in the request.	
Type:	Element
Format:	Char string
Defined values:	-
Default value:	-
Example:	<easting>621160.98</easting> *(if UTM_FORMAT=2)*
Note: -	

5.11 eme_event

Description:	
Specifies the events that initiated the positioning of the MS at an emergency call.	
Type:	Element
Format:	-
Defined values:	-
Default value:	-
Example:	<eme_event eme_trigger="EME_ORG">
Note: -	

5.11.1 eme_trigger

Description:		
Specifies the trigger that initiated the positioning of the MS at an emergency call.		
Type:	Attribute	
Format:	Char string	
Defined values:	EME_ORG	An emergency service user originated an emergency call
	EME_REL	An emergency service user released an emergency call
Default value:	-	
Example:	<eme_event eme_trigger="EME_ORG">	
Note: -		

5.12 esrd

Description:	
This element specifies Emergency Services Routing Digits (ESRD).	
Type:	Element
Format:	Char string
Defined values:	-
Default value:	-
Example:	<esrd>761287612582</esrd>
Note: -	

5.12.1 type

Description:		
Defines the origin of the ESRD		
Type:	Attribute	
Format:	Char string	
Defined values:	NA	Indicates that the ERSD is defined as the North American ESRD (NA-ERSD). NA-ESRD is a telephone number in the North American Numbering Plan that can be used to identify a North American emergency services provider and it's associated Location Services client. The NA-ESRD also identifies the base station, cell site or sector from which a North American emergency call originates
Default value:	NA	
Example:	<esrd type="NA">12345678</ersd>	
Note: Currently only NA is specified. It is expected that other origins will be specified in the future		

5.13 esrk

Description:		
This element specifies the Services Routing Key (ESRK).		
Type:	Element	
Format:	Char string	
Defined values:	-	
Default value:	-	
Example:	`<esrk>928273633343</esrk>`	
Note: -		

5.13.1 type

Description:		
Defines the origin of the ESRK		
Type:	Attribute	
Format:	Char string	
Defined values:	`NA`	Indicates that the ERSK is defined as the North American ESRK (NA-ERSK). NA-ESRK is a telephone number in the North American Numbering Plan that is assigned to an emergency services call for the duration of the call. The NA-ESRK is used to identify (e.g. route to) both the emergency services provider and the switch currently serving the emergency caller. During the lifetime of an emergency services call, the NA-ESRK also identifies the calling subscriber.
Default value:	`NA`	
Example:	`<esrk type="NA">12345678</ersk>`	
Note: Currently only NA is specified. It is expected that other origins will be specified in the future		

5.14 format

Description:			
With type="LL" This element specifies the output format of the geographical position when it is expressed in longitude and latitude.			
Type:	Element		
Format:	Char String The LL format is expressed as: [l]?[D | DM | DMS | M | MS | S] [0-9] [l]? Detailed description:		
	String	**Requested output**	**Example of output**
	`D`	Degrees only	45.403
	`DM`	Degrees and minutes	4515.557
	`DMS`	Degrees, minutes and seconds	451533.431
	`M`	Minutes only	16215.557
	`MS`	Minutes and seconds	1621533.431
	`S`	Seconds only	972933.431

	I	Output direction indicator The indicator has two valid positions, at the beginning or at the end of the string. If present a direction indicator (N | S | E | W) will be added to the output.	45.403W N16215.557
	The digit [0-9] defines the decimal precision of the output. Any trailing zero valued decimals will be deleted.		
Defined values:	-		
Default value:	DMS3		
Example:	<format type="LL">DM2I</format>		
Note: -			

Description:	
With type="UTM" Specifies the output format for UTM coordinates. The digit specifies the number of decimals that should be used in the output.	
Type:	Element
Format:	Char String
Defined values:	[0-9]
Default value:	3
Example:	<format type="UTM">5</format>
Note: -	

Description:	
With type="XY" Specifies the output format for coordinates expressed in X and Y. The digit specifies the number of decimals that should be used in the output.	
Type:	Element
Format:	Char String
Defined values:	[0-9]
Default value:	3
Example:	<format type="XY">2</format>
Note: -	

5.15 geo_info

Description:	
Type:	Element
Format:	-
Defined values:	-
Default value:	-
Example:	<geo_info coord_sys="UTM" datum="WGS-84">
Note: -	

5.16 hor_acc

Description:	
Requested horizontal accuracy in meters	
Type:	Element
Format:	Char String
Defined values:	[0-9]+
Default value:	-
Example:	`<hor_acc>200</hor_acc>`
Note: -	

5.17 id

Description:	
A string defining the name of a registered user performing a location request. In an answer the string represents the name of a location server.	
Type:	Element
Format:	Char string
Defined values:	-
Default value:	-
Example:	`<id>TheTruckCompany</id>`
Note: - This element is implementation specific. The second example illustrates an MSISDN number.	

5.18 in_rad

Description:	
The inner radius of an arc in meters	
Type:	Element
Format:	Char String
Defined values:	[0-9]+
Default value:	-
Example:	`<in_rad>100</in_rad>`
Note: If the inner radius is 0 (zero) the area described represents a circle sector.	

5.19 interval

Description:		
Specifies the interval between two responses in case of an LDR that indicates timer controlled, periodic responses.		
Type:	Element	
Format:	Char string The interval is expressed as `ddhhmmss` where:	
	String	**Description**
	`dd`	Number of days between responses
	`hh`	Number of hours between responses
	`mm`	Number of minutes between responses
	`ss`	Number of seconds between responses

Defined values:	-
Default value:	-
Example:	<interval>00010000</interval>
Note: -	

5.20 lac

Description:	
Identifies the Location Area Code	
Type:	Element
Format:	Char String
Defined values:	1-65535
Default value:	-
Example:	<lac>234</lac>
Note:	

5.21 lat

Description:	
Specifies the geodetic latitude of a point is the angle from the equatorial plane to the vertical direction of a line normal to the reference ellipsoid.	
Type:	Element
Format:	Char string The format is determined by the value of the LL_FORMAT element in the request.
Defined values:	-
Default value:	-
Example:	<lat>N301628.3</lat> *(if LL_FORMAT=IDMS1)*
Note: -	

5.22 lev_conf

Description:	
This parameter indicates the probability in percent that the MS is located in the position area that is returned.	
Type:	Element
Format:	Char String
Defined values:	0-100
Default value:	-
Example:	<lev_conf>80</lev_conf>
Note: -	

5.23 ll_acc

Description:	
Longitude and latitude accuracy in seconds.	
Type:	Element
Format:	Char String
Defined values:	-
Default value:	-
Example:	`<ll_acc>7.5</ll_acc>`
Note: -	

5.24 loc_type

Description:	
Defines the type of location requested.	
Type:	Element
Format:	-
Defined values:	-
Default value:	-
Example:	`<loc_type type="INITIAL" />`
Note: -	

5.24.1 type

Description:		
Defines the type of location requested		
Type:	Attribute	
Format:	Char string	
Defined values:	`CURRENT`	After a location attempt has successfully delivered a location estimate the location estimate is known as the current location at that point in time.
	`LAST`	The current location estimate is generally stored in the network until replaced by a later location estimate and is known as the last known location. The last known location may be distinct from the initial location., i.e. more recent.
	`INITIAL`	In an originating emergency call, this is the location estimate at the commencement of the call set-up and is known as the initial location.
Default value:	`CURRENT`	
Example:	`<loc_type type="INITIAL" />`	
Note: -		

5.25 long

Description:	
Specifies the longitude of a point is the angle between a reference plane and a plane passing through the point, both planes being perpendicular to the equatorial plane.	
Type:	Element
Format:	Char string The format is determined by the value of the LL_FORMAT element in the request.

Defined values:	-
Default value:	-
Example:	`<long>W974425.2</long>` *(if LL_FORMAT=IDMS1)*
Note: -	

5.26 max_loc_age

Description:	
This states the maximum allowable age in seconds of a location sent in a response to a location request.	
Type:	Element
Format:	Char string representing seconds
Defined values:	Maximum number of seconds (must be >= 0)
Default value:	Implementation specific.
Example:	`<max_loc_age>3600</max_loc_age>`
Note: -	

5.27 mcc

Description:	
Specifies the mobile country code (MCC).	
Type:	Element
Format:	Char String
Defined values:	`3 digits, e.g. 234 for the UK`
Default value:	-
Example:	`<mcc>234</mcc>`
Note:	

5.28 mnc

Description:	
Specifies the mobile network code.	
Type:	Element
Format:	Char string
Defined values:	Up to 3 digits e.g. 15 for Vodafone
Default value:	-
Example:	`<mnc>215</mnc>`
Note: -	

5.29 ms_action

Description:	
Specifies the trigger that initiated the positioning of the MS.	
Type:	Element
Format:	-
Defined values:	-
Default value:	-

Example:	`<ms_action type="MO" />`
Note: -	

5.29.1 type

Description:		
Specifies the trigger that initiated the positioning of the MS.		
Type:	Attribute	
Format:	Char string	
Defined values:	`MS_AVAIL`	The positioning is triggered by the MS originating a call The positioning is triggered by the MS available notification when the MS regains radio connection with the network if the connection was previously lost. For more information refer to 3GPP TS 23.271 rel. 4.
Default value:	-	
Example:	`<ms_action type="MS_AVAIL" />`	
Note: -		

5.30 msid

Description:	
This element represents an identifier of a mobile subscriber	
Type:	Element
Format:	Char string
Defined values:	-
Default value:	-
Example:	`<msid>460703057640</msid>`
Note: -	

5.30.1 type

Description:		
Type of identifier for the mobile subscriber		
Type:	Attribute	
Format:	Char string	
Defined values:	`MSISDN`	Mobile Station International ISDN Number
	`IMSI`	International Mobile Subscriber Identity
	`IMEI`	International Mobile station Equipment Identity
	`MIN`	Mobile Identification Number
	`MDN`	Mobile Directory Number
	`EME_MSID`	Emergency MSID
	`IPV4`	Mobile station IP address (Version 4)
	`IPV6`	Mobile station IP address (Version 6)
Default value:	`MSISDN`	
Example:	`<msid type="IMSI">`	
Note: -		

5.30.2 enc

Description:		
Type of encoding for MSID identifier for the mobile subscriber		
Type:	Attribute	
Format:	Char string	
Defined values:	ASC	Normal textual format
	B64	Base 64 encoding
	CRP	Encrypted format: In some countries the Network Operator (where is placed the Location Server) isn't allowed to send to a LCS the private information of an MS like MSISDN. The Network Operator can send out to LCS the Encrypted MSID, since only the Network Operator is the only entity able to decode this information, the LCS will be never able to break the privacy of the MS.
Default value:	ASC	
Example:	`<msid type="IMSI" enc="B64">`	
Note: -		

5.31 ndc

Description:	
Specifies the network destination code.	
Type:	Element
Format:	Char string
Defined values:	Up to 4 digits e.g. 7785 for Vodafone
Default value:	-
Example:	`<ndc>215</ndc>`
Note: -	

5.32 nmr

Description:	
Network specific measurement result for the target MS.	
Type:	Element
Format:	Char string
Defined values:	For examples see relevant standards documents. (GSM 04.08 – rel.98 section 10.5.2.20)
Default value:	-
Example:	
Note: This element remains to be defined.	

5.33 northing

Description:	
Used in the UTM coordinate system. Positions are measured in northings from the equator.. The number of decimals provided in the response is defined by UTM_FORMAT that is provided in the request.	
Type:	Element

Format:	Char string
Defined values:	-
Default value:	–
Example:	`<northing>3349893.53</northing>` *(if UTM_FORMAT=2)*
Note: -	

5.34 out_rad

Description:	
The outer radius of an arc in meters	
Type:	Element
Format:	Char String
Defined values:	[0-9]+
Default value:	-
Example:	`<out_rad>850</out_rad>`
Note: -	

5.35 prio

Description:	
Defines the priority of a location request	
Type:	Element
Format:	-
Defined values:	-
Default value:	-
Example:	`<prio />`
Note: -	

5.35.1 type

Description:		
Defines the priority of a location request		
Type:	Attribute	
Format:	Char string	
Defined values:	`NORMAL`	The request is handled with normal priority
	`HIGH`	The request is handled with high priority
Default value:	`NORMAL`	
Example:	`<prio type="HIGH" />`	
Note: -		

5.36 pwd

Description:	
The password for the registered user performing a location request. In an answer the string represents the password for a location server.	
Type:	Element
Format:	Char string
Defined values:	
Default value:	-
Example:	`<pwd>the5pwd</pwd>`
Note: -	

5.37 rad

Description:	
The radius of a circle in meters	
Type:	Element
Format:	Char String
Defined values:	[0-9]+
Default value:	-
Example:	`<rad>120</rad>`
Note: -	

5.38 req_id

Description:	
Unique identification of a request	
Type:	Element
Format:	Char string
Defined values:	-
Default value:	-
Example:	`<req_id>435.23.01</req_id>`
Note: -	

5.39 resp_req

Description:	
This attribute represents response time requirement.	
Type:	Element
Format:	Char string
Defined values:	-
Default value:	-
Example:	`<resp_req type="NO_DELAY" />`
Note: -	

5.39.1 type

Description:		
This attribute represents response time requirement		
Type:	Attribute	
Format:	Char String	
Defined values:	NO_DELAY	No delay: The server should immediately return any location estimate that it currently has.
	LOW_DELAY	Low delay: Fulfilment of the response time requirement takes precedence over fulfilment of the accuracy requirement.
	DELAY_TOL	Delay tolerant: Fulfillment of the accuracy requirement takes precedence over fulfillment of the response time requirement
Default value:	DELAY_TOL	
Example:	<resp_req />	
Note: -		

5.40 resp_timer

Description:		
Defines a timer for the response time within which the current location should be obtained and returned to the LCS Client.		
Type:	Element	
Format:	Char String The time is expressed as mmss where:	
	mm	Minutes
	ss	Seconds
Defined values:	-	
Default value:	The default value is defined in the location server	
Example:	<resp_timer>0010</resp_timer>	
Note: -		

5.41 result

Description:	
A text string indicating the result of the request or an individual positioning	
Type:	Element
Format:	Char string
Defined values:	See chapter 6.1
Default value:	-
Example:	<result resid=0>OK</result>
Note: -	

5.41.1 resid

Description:	
This attribute represents a numeric representation of a result message	
Type:	Attribute

Format:	Char String
Defined values:	[0-9]+ See chapter 6.1
Default value:	-
Example:	`<result resid=0>OK</result>`
Note: -	

5.42 semi_major

Description:	
Specifies the length of the semi-major axis of an ellipse in meters.	
Type:	Element
Format:	Char String
Defined values:	[0-9]+
Default value:	-
Example:	`<semi_major>560</semi_major>`
Note: -	

5.43 semi_minor

Description:	
Specifies the length of the semi-minor axis of an ellipse in meters.	
Type:	Element
Format:	Char String
Defined values:	[0-9]+
Default value:	-
Example:	`<semi_minor>560</semi_minor>`
Note: -	

5.44 serviceid

Description:	
Specifies an id that is used by an entity to identify the service or application that is accessing the network.	
Type:	Element
Format:	Char String
Defined values:	-
Default value:	-
Example:	`<serviceid>0005</serviceid>`
Note:	

5.45 servicetype

Description:		
Defines the type of the service that has been requested by the ASP.		
Type:	Element	
Format:	-	
Defined values:	-	
Default value:	-	
Example:	`<servicetype />`	
Note:		

5.45.1 type

Description:		
Defines the type of the service that has been requested by the ASP		
Type:	Attribute	
Format:	Char string	
Defined values:	`PASSIVE`	The service is one that is not directly initiated by the user.
	`ACTIVE`	The service is one that the user is initiating personally.
Default value:	`PASSIVE`	
Example:	`<servicetype type="ACTIVE" />`	
Note: the default value is set to PASSIVE, as this is likely to be the one that is most restrictively defined by the user.		

5.46 session

Description:		
This element identifies should be presented in location request when the LCS Client is making has an active session with the User Equipment, this will be either the number called by the UE or the APN on which the UE established the session.		
Type:	Element	
Format:	Char String	
Defined values:	-	
Default value:	-	
Example:	`<session>447073100177</session>`	
Note: This information may be required for privacy validation of the location request by the VMSC, SGSN or MSC server		

5.46.1 type

Description:		
Defines the type of the session that is established between the User Equipment and LCS Client		
Type:	Attribute	
Format:	Char string	
Defined values:	`APN`	Access Point Name.
	`dial`	The number dialed by the user to access the LCS client.
Default value:	-	
Example:	`<session type type="dial" />`	
Note:		

5.47 sessionid

Description:	
Specifies an id that can be used by an entity to replace the need to use an ID and PWD to use the location services. In an answer it indicates the sessionid that the entity can use on subsequent requests. The Session id is a generated alphanumeric string and can be time-limited.	
Type:	Element
Format:	Char String
Defined values:	-
Default value:	-
Example:	`<sessionid>34eg6.876.76h4</sessionid>`
Note:	

5.48 speed

Description:	
The speed of the MS in m/s.	
Type:	Element
Format:	Char String
Defined values:	[0-9]+
Default value:	-
Example:	`<speed>23</speed>`
Note: This element is present if speed is possible to attain by the used positioning method.	

5.49 start_angle

Description:	
Specifies a start angle in degrees.	
Type:	Element
Format:	Char string
Defined values:	0-360
Default value:	-
Example:	`<start_angle>60</start_angle>`
Note: -	

5.50 start_msid

Description:	
This element represents an identifier of a mobile subscriber, which is used as start of a range.	
Type:	Element
Format:	Char string
Defined values:	-
Default value:	-

Example:	`<start_msid>460703057640</start_msid>`
Note: -	

5.51 start_time

Description:	
This element defines the absolute start time in a range of times.	
Type:	Element
Format:	Char String The time is expressed as yyyyMM`ddhhmmss` where:

String	**Description**
`yyyy`	Year
`MM`	Month
`dd`	Day
`hh`	Hours
`mm`	Minutes
`ss`	Seconds

Defined values:	-
Default value:	-
Example:	`<start_time>20010630142810</start_time>`
Note: -	

5.51.1 utc_off

Description:	
Specifies the UTC offset in hours and minutes. Positive values indicate time zones east of Greenwich.	
Type:	Attribute
Format:	Char string
Defined values:	[+|-]0000-1400
Default value:	-
Example:	`<start_time utc_off="+0200">20000813010423</start_time>`
Note: utc_off is specifed as 'HHMM', where 'HH' can range between 0-14 and 'MM' between '0-59'. All other values shall result in error 105, 'Format error'.	

5.52 stop_angle

Description:	
Specifies a stop angle in degrees.	
Type:	Element
Format:	Char string
Defined values:	0-360
Default value:	-

Example:	`<stop_angle>180</stop_angle>`
Note: -	

5.53 stop_msid

Description:	
This element represents an identifier of a mobile subscriber, which is used as end of a range.	
Type:	Element
Format:	Char string
Defined values:	-
Default value:	-
Example:	`<stop_msid>460703057640</stop_msid>`
Note: -	

5.54 stop_time

Description:		
This element defines the absolute stop time in a range of times.		
Type:	Element	
Format:	Char String The time is expressed as yyyyMM`ddhhmmss` where:	
	String	**Description**
	`yyyy`	Year
	`MM`	Month
	`dd`	Day
	`hh`	Hours
	`mm`	Minutes
	`ss`	Seconds
Defined values:	-	
Default value:	-	
Example:	`<stop_time>20010630142810</stop_time>`	
Note: -		

5.54.1 utc_off

Description:	
Specifies the UTC offset in hours and minutes. Positive values indicate time zones east of Greenwich.	
Type:	Attribute
Format:	Char string
Defined values:	[+|-]0000-1400
Default value:	-
Example:	`<stop_time utc_off="+0200">20000813010423</stop_time>`
Note: utc_off is specifed as 'HHMM', where 'HH' can range between 0-14 and 'MM' between '0-59'. All other values shall result in error 105, 'Format error'.	

5.55 subclient

Description:		
Identifies the ASPs, resellers and portals in the chain of service providers between the network and the end-user		
Type:	Element	
Format:	-	
Defined values:	-	
Default value:	-	

Example:	`<subclient last_client="NO">`
	`<id>TheASP</id>`
	`<serviceid>0006</serviceid>`
	`</subclient>`	

Note: -		

5.55.1 last_client

Description:		
Identifies whether the SUBCLIENT is the last one in the chain or not		
Type:	Attribute	
Format:	Char String	
Defined values:	YES	This is the last client – the one that the end-user is actually communicating with
	NO	This is not the last client
Default value:	NO	
Example:	`<subclient last_client="YES">`	
Note: -		

5.56 ta

Description:	
This Radio Access Network element that can arguably be used to offer enhanced positioning.	
Type:	Element
Format:	Char string
Defined values:	0-62
Default value:	0
Example:	`<ta></ta>`
Note: Further Information regarding this element can be found in the relevant GSM Specifications	

5.57 time

Description:
This element defines the absolute time when to perform a positioning at a timer controlled LDR. In a location answer this element indicates the time when the positioning was performed.

Type:	Element
Format:	Char String The time is expressed as yyyyMMddhhmmss where:

String	Description
yyyy	Year
MM	Month
dd	Day
hh	Hours
mm	Minutes
ss	Seconds

Defined values:	-
Default value:	-
Example:	<time>20010630142810</time>
Note: -	

5.57.1 utc_off

Description:	
Specifies the UTC offset in hours and minutes. Positive values indicate time zones east of Greenwich.	
Type:	Attribute
Format:	Char string
Defined values:	[+|-]0000-1400
Default value:	-
Example:	<time utc_off="+0200">20000813010423</time>
Note: utc_off is specifed as 'HHMM', where 'HH' can range between 0-14 and 'MM' between '0-59'. All other values shall result in error 105, 'Format error'.	

5.58 time_remaining

Description:	
Defines the time remaining until the location server terminates the current triggered location service. The time for which the service is valid is either specified by the client using start time and stop time, or is a network operator specific default value where no s stop time is defined or where the stop time exceeds the allowed value by the location server involved.	
Type:	Element
Format:	Char String The time is expressed as ddhhmmss where:

String	Description
dd	Day
hh	Hours
mm	Minutes
ss	Seconds

Defined values:	-
Default value:	The default value is defined in the location server
Example:	<time_remaining>00010000</time_remaining>

Note: -

5.59 trl_trigger

Description:		
Specifies the trigger that initiated the positioning of the MS at a triggered location report.		
Format:	Char string	
Defined values:	TIMER	The positioning is triggered by the appointed time
	PERIODIC	The positioning is triggered when the periodical timer expired
	MS_AVAIL	The positioning is triggered by the MS presence notification
Default value:	-	
Example:	<MS_ACTION ms_action_type="MS_AVAIL" />	
Note: -		

5.60 url

Description:	
Specifies the location to which a response to a LDR should be sent to	
Type:	Element
Format:	Char string
Defined values:	-
Default value:	-
Example:	<url>http://location.client.com/Response/</url>
Note: -	

5.61 vlrno

Description:	
Uniquely specifies a VLR within a network.	
Type:	Element
Format:	Char String
Defined values:	In GSM this is the Global Title address. The Global Title is in the same format as an E.164 number.
Default value:	-
Example:	<vlrno>1541154871</vlrno>
Note:	

5.62 vmscno

Description:
Uniquely specifies a VMSC within a network.

Type:	Element
Format:	Char String
Defined values:	In GSM this is the Global Title address. The Global Title is in the same format as an E.164 number.
Default value:	-
Example:	<vmscno>1541154871</vmscno>
Note:	

5.63 x

Description:	
Used when a position is expressed in X and Y coordinates to represent a position on the X-axis of the coordinate system. The number of decimals provided in the response is defined by FORMAT that is provided in the request.	
Type:	Element
Format:	Char string
Defined values:	-
Default value:	-
Example:	<x>33498.23</x> *(if FORMAT=2)*
Note: -	

5.64 y

Description:	
Used when a position is expressed in X and Y coordinates to represent a position on the Y-axis of the coordinate system. The number of decimals provided in the response is defined by FORMAT that is provided in the request.	
Type:	Element
Format:	Char string
Defined values:	-
Default value:	-
Example:	<y>33498.23</y> *(if FORMAT=2)*
Note: -	

5.65 zone

Description:	
Identifies the zone when the UTM or other planar coordinate system is used.	
Type:	Element
Format:	Char String
Defined values:	[0-9]+
Default value:	-
Example:	<zone>14</zone>
Note: -	

5.66 zone_des

Description:		
Identifies the zone designator when the UTM coordinate system is used.		
Type:	Element	
Format:	Char string	
Defined values:	-	
Default value:	–	
Example:	`<zone_des>R</zone_des>`	
Note: -		

5.67 Service attributes

5.67.1 res_type

Description:		
Defines a response type at the Standard Location Immediate Service		
Type:	Attribute	
Format:	Char string	
Defined values:	`PERSISTENT`	An LCS Client can receive the responses at one time using persistent connection
	`PUSH`	An LCS Client can receive the responses one by one using some connections initiated by the LCS Server
Default value:	`PERSISTENT`	
Example:	`<slir ver="2.0.0" res_type="PERSISTENT">`	
Note: -		

5.67.2 ver

Description:	
Defines the version of the location protocol. This attribute is valid for ALL messages	
Type:	Element
Format:	Char string
Defined values:	[0-9].[0-9]
Default value:	–
Example:	`<slia ver="2.0.0">`
Note: -	

6 Result codes and error codes

6.1 Result codes

This table defines the general result codes that indicate the result of the whole request.

Result code	Slogan	Description
0	OK	No error occurred while processing the request.
1	SYSTEM FAILURE	The request can not be handled because of a general problem in the location server or the underlying network
2	UNAUTHORIZED NETWORK	The requesting network is not allowed to access the location server
3	UNAUTHORIZED APPLICATION	The requesting location-based application is not allowed to access the location server
4	UNKNOWN SUBSCRIBER	Unknown subscriber. The user is unknown, i.e. no such subscription exists.
5	ABSENT SUBSCRIBER	Absent subscriber. The user is currently not reachable.
6	POSITION METHOD FAILURE	Position method failure. The location service failed to obtain the user's position.
101	CONGESTION IN LOCATION SERVER	The request can not be handled due to congestion in the location server
102	CONGESTION IN MOBILE NETWORK	The request can not be handled due to congestion in the mobile network
103	INCORRECT PASSWORD	The location-based application is allowed to access the location server, but the supplied password is incorrect.
104	TOO MANY POSITION ITEMS	Too many position items have been specified in the request.
105	FORMAT ERROR	A parameter in the request has invalid format. The invalid parameter is indicated in ADD_INFO.
106	SYNTAX ERROR	The position request has invalid syntax. Details may be indicated in ADD_INFO.
107	PROTOCOL ELEMENT NOT SUPPORTED	An element specified in the position request is not supported by the Location Server. The element is indicated in ADD_INFO.
108	SERVICE NOT SUPPORTED	The requested service is not supported in the Location Server. The service may be indicated in ADD_INFO
109	ELEMENT ATTRIBUTE NOT SUPPORTED	An element attribute is not supported in the Location Server. The attribute is indicated in ADD_INFO
110	INVALID TIME RANGE	The time range specified is incorrect. The incorrect time is indicated in ADD_INFO
201	UNKNOWN SUBSCRIBER	The user is unknown, i.e. no such subscription exists
202	NOT IN PRIVACY EXCEPTION LIST	The requesting application not in the privacy exception list of the MS.
203	CALL TO USER NOT SETUP	The requesting application has not a call set up to an MS that only allows call related location requests.
204	DISALLOWED BY LOCAL REGULATIONS	The location request is disallowed by local regulatory requirements
207	MISCONFIGURATION OF LOCATION SERVER	The location server is not completely configured to be able to calculate a position.

7 References (Normative and Informative)

[3] Hypertext Transfer Protocol –HTTP/1.1
RFC 2068, June 1999
Available at http://www.w3c.org

[4] The TLS Protocol Version 1.0
RFC 2246, January 1999
Available at http://www.ietf.org

[5] Extensible Markup Language (XML) 1.0
W3C Recommendation: REC-xml-19980210
Available at http://www.w3c.org

[6] Internet Assigned Numbers Authority (IANA)
http://www.iana.org/

[7] US-ASCII. Coded Character Set - 7-Bit American Standard Code for Information Interchange. Standard ANSI X3.4-1986, ANSI, 1986.

[8] GSM 02.71: "Digital cellular telecommunications system (Phase 2+); Location Services (LCS); Service description; Stage 1".

[9] GSM 03.71: "Digital cellular telecommunications system (Phase 2+); Location Services (LCS); Functional description; Stage 2".

[10] GSM 09.02: "Digital cellular telecommunications system (Phase 2+); Mobile Application Part (MAP) specification".

[11] 3G TS 22.071: "Location Services (LCS); Service description, Stage 1".

[12] 3G TS 23.171: "Functional stage 2 description of location services in UMTS"

[13] Parlay Mobility Service Interface version 1.1 that is a part of version 2.1 of the Parlay API Specification.
Available on the Parlay web-site at http://www.parlay.org

8 Appendix A (informative)

The terminology mapping table with 3GPP LCS Specifications

The following is the list of the terms in MLP used differently from the ones defined for 3GPP.

Term		notes
MLP	3GPP	
Location Server	LCS Server	
MS (Mobile Station)	UE	
MSID (Mobile Station Identifer)	Identification of the target UE	
MPC (Mobile Positioning Center)	□	There is no term which is applicable to 3GPP.

9 Appendix B (informative)

The corresponding terms used for the location procedures in 3GPP LCS Definition

The following is the list of terms defined in MLP corresponding to the 3GPP LCS definition in TS23.271 for the location procedures.

Location procedures defined with 3GPP(23.271)		Services defined with MLP
Circuit Switched Mobile Terminating Location Request CS-MT-LR	LCS Service Request	Standard Location Immediate Request
	LCS Service Responce	Standard Location Immediate Answer
CS-MT-LR without HLR Query - applicable to North America Emergency Calls only	LCS Service Request	Emergency Location Immediate Request
	LCS Service Responce	Emergency Location Immediate Answer
Packet Switched Mobile Terminating Location Request PS-MT-LR	LCS Service Request	Standard Location Immediate Request
	LCS Service Responce	Standard Location Immediate Answer
Network Induced Location Request NI-LR	Location Information	Emergency Location Report
Packet Switched Network Induced Location Request PS-NI-LR	Location Information	Emergency Location Report
Mobile Terminating Deferred Location Request	LCS Service Request	Triggered Location Reporting Request
	LCS Service Responce (Provide Subscriber Location ack)	Triggerrd Location Reporting Answer
	LCS Service Responce (Subscriber Location Report)	Triggered Location Report
Cancellation of a Deferred Location Request	LCS Cancel Service Request	Triggered Location Reporting Stop
	LCS Cancel Service Response	Triggered Location Reporting Stop Answer
Mobile Originating Location Request, Circuit Switched CS-MO-LR	Location Information	Standard Location Report
Mobile Originating Location Request, Packet Switched PS-MO-LR	Location Information	Standard Location Report