

ETSI ES 201 915-6 V1.2.1 (2002-07)

ETSI Standard

Open Service Access (OSA); Application Programming Interface (API); Part 6: Mobility SCF



Reference

RES/SPAN-120076-6

Keywords

API, IDL, UML, OSA

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - 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

Important notice

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

The present document 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.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, send your comment to:

editor@etsi.fr

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2002.

© The Parlay Group 2002.

All rights reserved.

DECT™, **PLUGTESTS™** and **UMTS™** are Trade Marks of ETSI registered for the benefit of its Members.
TIPHON™ and the **TIPHON logo** are Trade Marks currently being registered by ETSI for the benefit of its Members.
3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

Contents

Intellectual Property Rights	6
Foreword.....	6
1 Scope	7
2 References	7
3 Definitions and abbreviations.....	7
3.1 Definitions	7
3.2 Abbreviations	7
4 Mobility SCF.....	8
5 Sequence Diagrams	8
5.1 User Location Sequence Diagrams.....	8
5.1.1 User Location Interrogation - Triggered Request	8
5.1.2 User Location Interrogation - Periodic Request.....	10
5.1.3 User Location Interrogation - Parameter Error	10
5.1.4 User Location Interrogation - Network Error	11
5.1.5 User Location Interrogation - Interactive Request.....	12
5.2 User Location Camel Sequence Diagrams	13
5.2.1 User Location Camel Interrogation - Triggered Request.....	13
5.2.2 User Location Camel Interrogation - Periodic Request	14
5.2.3 User Location Camel Interrogation - Parameter Error.....	15
5.2.4 User Location Camel Interrogation - Network Error	15
5.2.5 User Location Camel Interrogation - Interactive Request	16
5.3 User Location Emergency Sequence Diagrams	17
5.3.1 Subscription and Network Induced Location Reports	17
5.3.2 Network Induced Location Reports	18
5.3.3 Interactive Request Parameter Error	18
5.3.4 Interactive Request Network Error	19
5.3.5 Interactive Location Request	19
5.4 User Status Sequence Diagrams	20
5.4.1 Triggered Reporting.....	20
5.4.2 Interactive Request Parameter Error	21
5.4.3 Interactive Request Network Error	21
5.4.4 Interactive Request	22
6 Class Diagrams.....	23
6.1 User Location Class Diagrams	23
6.2 User Location Camel Class Diagrams.....	24
6.3 User Location Emergency Class Diagrams	25
6.4 User Status Class Diagrams.....	26
7 The Service Interface Specifications	26
7.1 Interface Specification Format	26
7.1.1 Interface Class	26
7.1.2 Method descriptions.....	27
7.1.3 Parameter descriptions.....	27
7.1.4 State Model.....	27
7.2 Base Interface	27
7.2.1 Interface Class IpInterface	27
7.3 Service Interfaces	27
7.3.1 Overview	27
7.4 Generic Service Interface	28
7.4.1 Interface Class IpService	28
8 Mobility Interface Classes.....	29
8.1 User Location Interface Classes	29

8.1.1	Interface Class IpUserLocation.....	29
8.1.2	Interface Class IpAppUserLocation.....	33
8.1.3	Interface Class IpTriggeredUserLocation.....	35
8.1.4	Interface Class IpAppTriggeredUserLocation.....	37
8.2	User Location Camel Interface Classes.....	38
8.2.1	Interface Class IpUserLocationCamel.....	38
8.2.2	Interface Class IpAppUserLocationCamel.....	42
8.3	User Location Emergency Interface Classes.....	44
8.3.1	Interface Class IpUserLocationEmergency.....	45
8.3.2	Interface Class IpAppUserLocationEmergency.....	47
8.4	User Status Interface Classes.....	48
8.4.1	Interface Class IpAppUserStatus.....	48
8.4.2	Interface Class IpUserStatus.....	50
9	State Transition Diagrams.....	52
9.1	User Location.....	52
9.2	User Location Camel.....	52
9.2.1	State Transition Diagrams for IpUserLocationCamel.....	52
9.2.1.1	Active State.....	52
9.3	User Location Emergency.....	52
9.4	User Status.....	53
9.4.1	State Transition Diagrams for IpUserStatus.....	53
9.4.1.1	Active State.....	53
10	Service Properties.....	53
10.1	Mobility Properties.....	53
10.1.1	Emergency Application Subtypes.....	53
10.1.2	Value Added Application Subtypes.....	54
10.1.3	PLMN Operator Application Subtypes.....	54
10.1.4	Lawful Intercept Application Subtypes.....	54
10.1.5	Altitude Obtainable.....	54
10.1.6	Location Methods.....	54
10.1.7	Priorities.....	55
10.1.8	Max Interactive Requests.....	55
10.1.9	Max Triggered Users.....	55
10.1.10	Max Periodic Users.....	55
10.1.11	Min Periodic Interval Duration.....	55
10.2	User Location Service Properties.....	55
10.3	User Location Camel Service Properties.....	56
10.4	User Location Emergency Service Properties.....	56
10.5	User Status Service Properties.....	57
11	Data Definitions.....	57
11.1	Common Mobility Data Definitions.....	57
11.1.1	TpGeographicalPosition.....	57
11.1.2	TpLocationPriority.....	58
11.1.3	TpLocationRequest.....	59
11.1.4	TpLocationResponseIndicator.....	59
11.1.5	TpLocationResponseTime.....	59
11.1.6	TpLocationType.....	59
11.1.7	TpLocationUncertaintyShape.....	60
11.1.8	TpMobilityDiagnostic.....	60
11.1.9	TpMobilityError.....	61
11.1.10	TpMobilityStopAssignmentData.....	61
11.1.11	TpMobilityStopScope.....	61
11.1.12	TpTerminalType.....	61
11.2	User Location Data Definitions.....	62
11.2.1	IpUserLocation.....	62
11.2.2	IpUserLocationRef.....	62
11.2.3	IpAppUserLocation.....	62
11.2.4	IpAppUserLocationRef.....	62
11.2.5	IpTriggeredUserLocation.....	62
11.2.6	IpTriggeredUserLocationRef.....	62

11.2.7	IpAppTriggeredUserLocation.....	62
11.2.8	IpAppTriggeredUserLocationRef.....	62
11.2.9	TpUIExtendedData.....	62
11.2.10	TpUIExtendedDataSet.....	62
11.2.11	TpUserLocationExtended.....	63
11.2.12	TpUserLocationExtendedSet.....	63
11.2.13	TpLocationTrigger.....	63
11.2.14	TpLocationTriggerSet.....	63
11.2.15	TpLocationTriggerCriteria.....	63
11.2.16	TpUserLocation.....	63
11.2.17	TpUserLocationSet.....	63
11.3	User Location Camel Data Definitions.....	64
11.3.1	IpUserLocationCamel.....	64
11.3.2	IpUserLocationCamelRef.....	64
11.3.3	IpAppUserLocationCamel.....	64
11.3.4	IpAppUserLocationCamelRef.....	64
11.3.5	TpLocationCellIDOrLAI.....	64
11.3.6	TpLocationTriggerCamel.....	64
11.3.7	TpUserLocationCamel.....	65
11.3.8	TpUserLocationCamelSet.....	65
11.4	User Location Emergency Data Definitions.....	65
11.4.1	IpUserLocationEmergency.....	65
11.4.2	IpUserLocationEmergencyRef.....	65
11.4.3	IpAppUserLocationEmergency.....	65
11.4.4	IpAppUserLocationEmergencyRef.....	65
11.4.5	TpIMEI.....	65
11.4.6	TpNaESRD.....	65
11.4.7	TpNaESRK.....	66
11.4.8	TpUserLocationEmergencyRequest.....	66
11.4.9	TpUserLocationEmergency.....	66
11.4.10	TpUserLocationEmergencyTrigger.....	67
11.5	User Status Data Definitions.....	67
11.5.1	IpUserStatus.....	67
11.5.2	IpUserStatusRef.....	67
11.5.3	IpAppUserStatus.....	67
11.5.4	IpAppUserStatusRef.....	67
11.5.5	TpUserStatus.....	67
11.5.6	TpUserStatusSet.....	67
11.5.7	TpUserStatusIndicator.....	67
11.6	Units and Validations of Parameters.....	68
12	Exception Classes.....	69
Annex A (normative):	OMG IDL Description of Mobility SCF.....	70
Annex B (informative):	Contents of 3GPP OSA R4 Mobility.....	71
Annex C (informative):	Summary of differences between V1.1.1 (Parlay 3.0) and V1.2.1 (Parlay 3.1).....	72
C.1	IpUserLocation.....	72
C.2	IpTriggeredUserLocation.....	72
C.3	IpUserLocationCamel.....	72
C.4	IpUserLocationEmergency.....	72
C.5	IpUserStatus.....	72
C.6	IpService.....	72
History	73

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 ETSI 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://webapp.etsi.org/IPR/home.asp>).

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 ETSI 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 ETSI Standard (ES) has been produced by ETSI Technical Committee Services and Protocols for Advanced Networks (SPAN).

The present document is part 6 of a multi-part deliverable covering Open Service Access (OSA); Application Programming Interface (API), as identified below. The API specification (ES 201 915) is structured in the following parts:

- Part 1: "Overview";
- Part 2: "Common Data Definitions";
- Part 3: "Framework";
- Part 4: "Call Control SCF";
- Part 5: "User Interaction SCF";
- Part 6: "Mobility SCF";**
- Part 7: "Terminal Capabilities SCF";
- Part 8: "Data Session Control SCF";
- Part 9: "Generic Messaging SCF";
- Part 10: "Connectivity Manager SCF";
- Part 11: "Account Management SCF";
- Part 12: "Charging SCF".

The present document has been defined jointly between ETSI, The Parlay Group [24] of ES 201 915-1 and the 3GPP, in co-operation with a number of JAIN™ Community [25] of ES 201 915-1 member companies.

The present document forms part of the Parlay 3.1 set of specifications.

1 Scope

The present document is part 6 of the Stage 3 specification for an Application Programming Interface (API) for Open Service Access (OSA).

The OSA specifications define an architecture that enables application developers to make use of network functionality through an open standardised interface, i.e. the OSA APIs.

The present document specifies the Mobility Service Capability Feature (SCF) aspects of the interface. All aspects of the Mobility SCF are defined here, these being:

- Sequence Diagrams
- Class Diagrams
- Interface specification plus detailed method descriptions
- State Transition diagrams
- Data Definitions
- IDL Description of the interfaces

The process by which this task is accomplished is through the use of object modelling techniques described by the Unified Modelling Language (UML).

2 References

The references listed in clause 2 of ES 201 915-1 contain provisions which, through reference in this text, constitute provisions of the present document.

ETSI ES 201 915-1: "Open Service Access (OSA); Application Programming Interface (API); Part 1: Overview".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ES 201 915-1 apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations defined in ES 201 915-1 apply.

4 Mobility SCF

The following clauses describe each aspect of the Mobility Service Capability Feature (SCF).

The order is as follows:

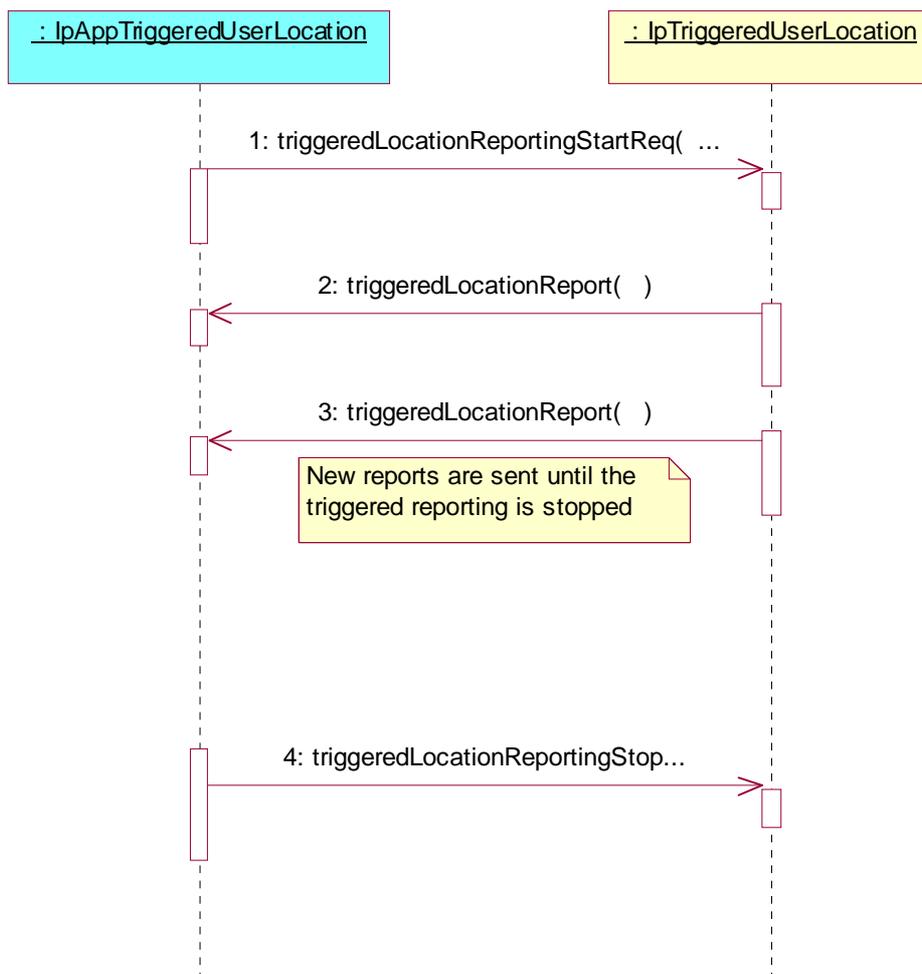
- The Sequence diagrams give the reader a practical idea of how each of the SCFs is implemented.
- The Class relationships clause show how each of the interfaces applicable to the SCF, relate to one another.
- The Interface specification clause describes in detail each of the interfaces shown within the Class diagram part.
- The State Transition Diagrams (STD) show the transition between states in the SCF. The states and transitions are well-defined; either methods specified in the Interface specification or events occurring in the underlying networks cause state transitions.
- The Data Definitions clause show a detailed expansion of each of the data types associated with the methods within the classes. Note that some data types are used in other methods and classes and are therefore defined within the Common Data types part of this specification.

5 Sequence Diagrams

5.1 User Location Sequence Diagrams

5.1.1 User Location Interrogation - Triggered Request

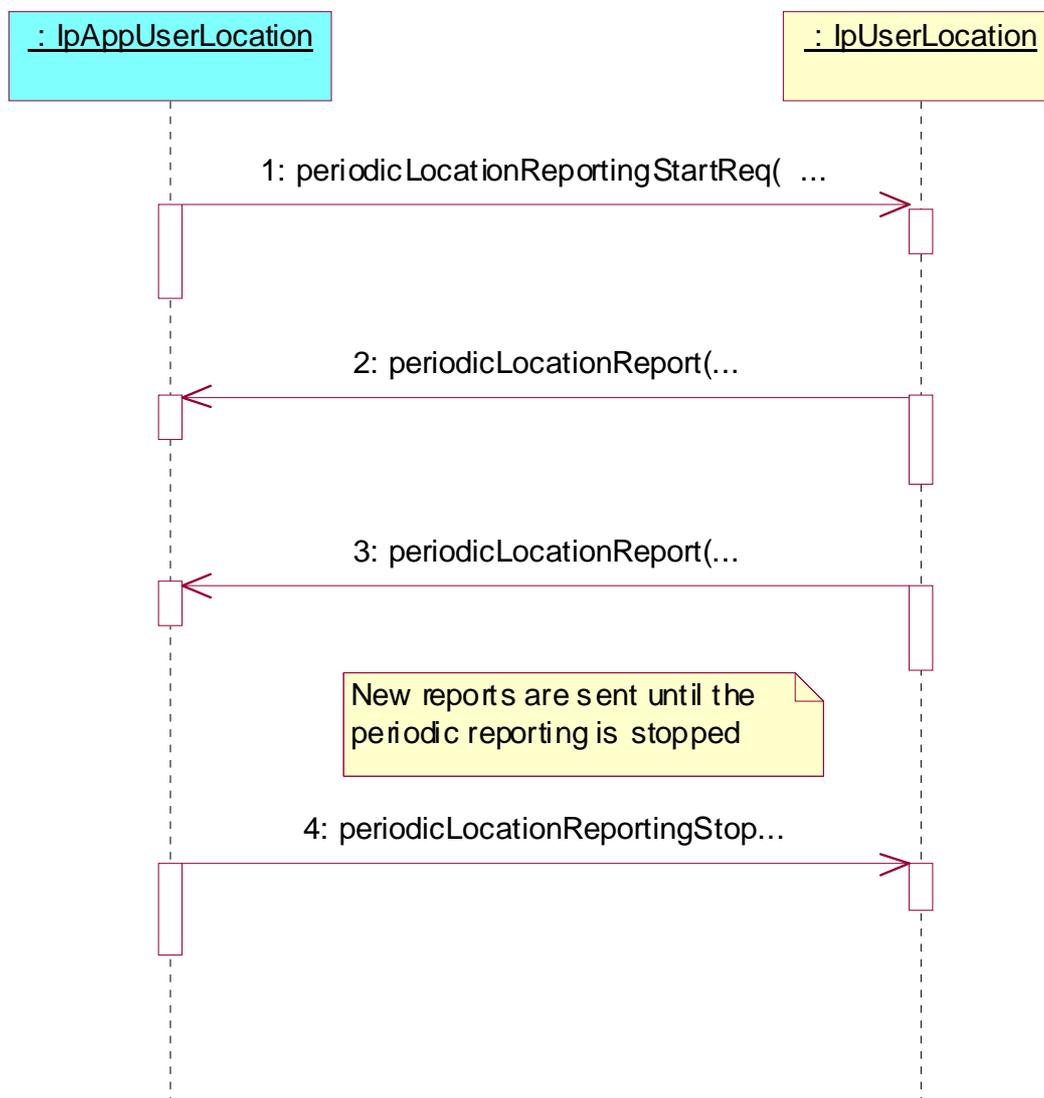
The following sequence diagram shows how an application requests triggered location reports from the User Location service. When users location changes, the service reports this to the application.



- 1: This message is used to start triggered location reporting for one or several users.
- 2: When the trigger condition is fulfilled then this message passes the location of the affected user to its callback object.
- 3: This is repeated until the application stops triggered location reporting (see next message).
- 4: This message is used to stop triggered location reporting.

5.1.2 User Location Interrogation - Periodic Request

The following sequence diagram shows how an application requests periodic location reports from the User Location service.



1: This message is used to start periodic location reporting for one or several users.

2: This message passes the location of one or several users to its callback object.

3: This message passes the location of one or several users to its callback object.

This is repeated at regular intervals until the application stops periodic location reporting (see next message).

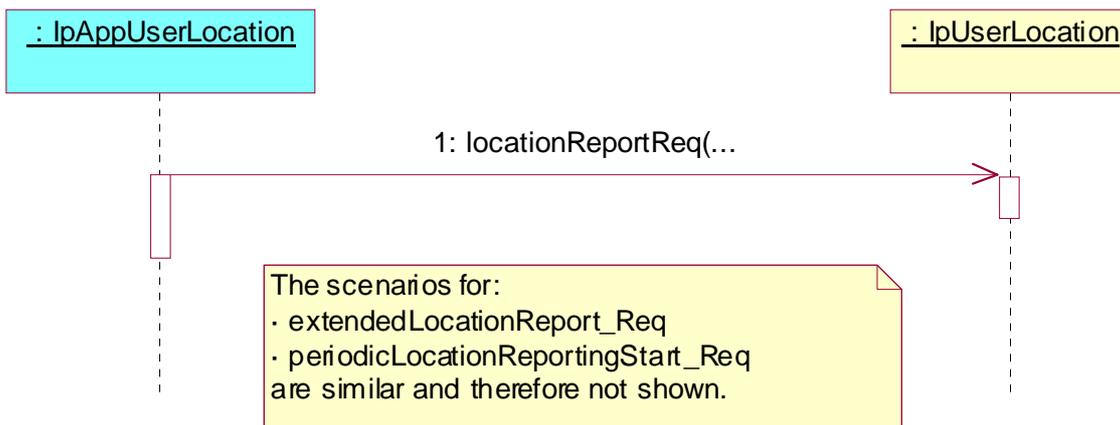
4: This message is used to stop periodic location reporting.

5.1.3 User Location Interrogation - Parameter Error

The following sequence diagram show a scenario where the application is requesting a location report from the User Location service but there is at least one error in the parameters that is detected by the service. The scenarios for:

- extendedLocationReportReq
- periodicLocationReportingStartReq

are similar and therefore not shown.



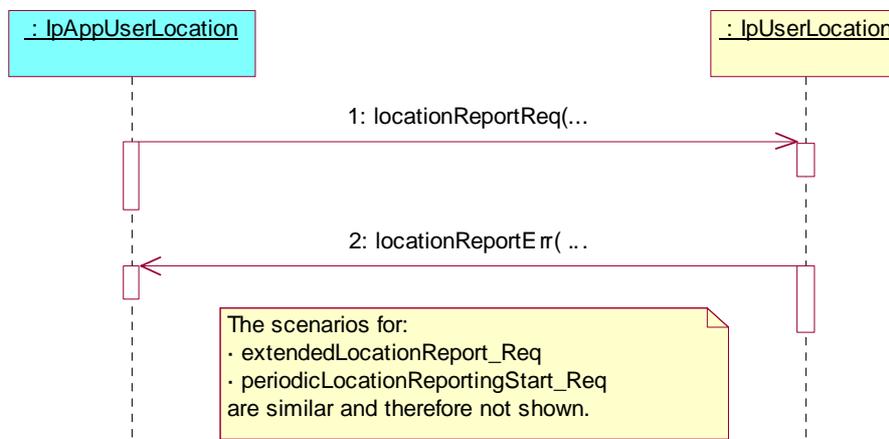
1: This message is used to request the location of one or several users, but the service returns an error and the execution of the request is aborted.

5.1.4 User Location Interrogation - Network Error

The following sequence diagram shows a scenario where the application is requesting a location report from the User Location service, but a network error occurs. The scenarios for:

- extendedLocationReportReq
- periodicLocationReportingStartReq

are similar and therefore not shown.

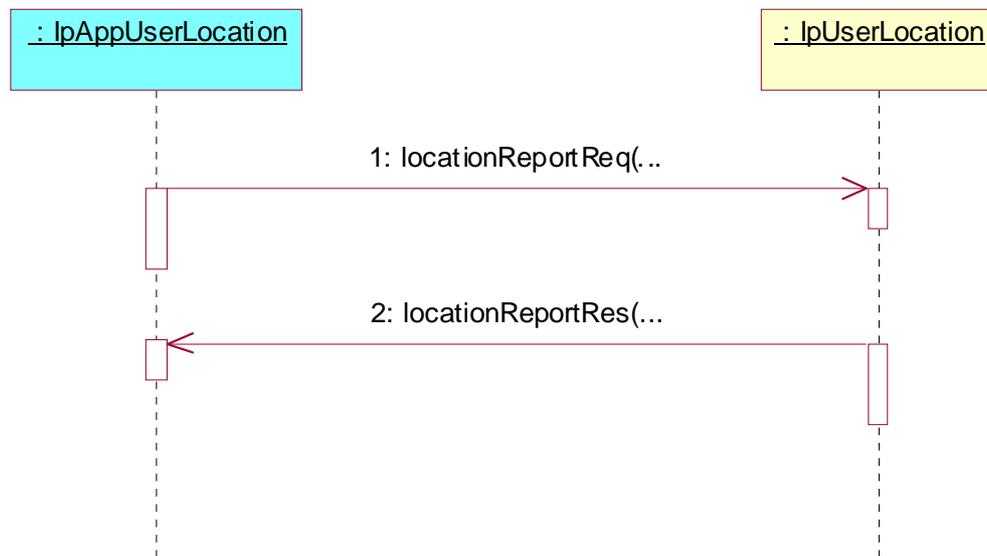


1: This message is used to request the location of one or several users.

2: This message passes information about the error in the location request from the network to the callback object.

5.1.5 User Location Interrogation - Interactive Request

The following sequence diagram shows how an application requests a location report from the User Location service.

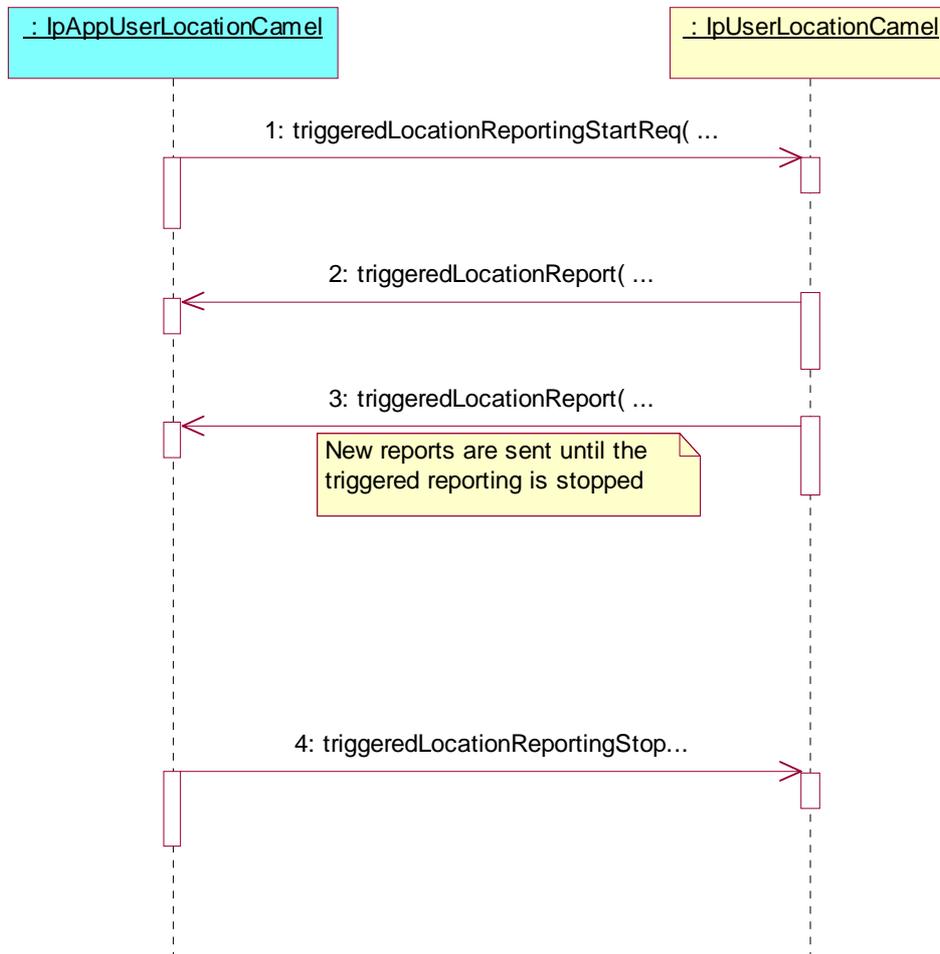


- 1: This message is used to request the location of one or several users.
- 2: This message passes the result of the location request for one or several users to its callback object.

5.2 User Location Camel Sequence Diagrams

5.2.1 User Location Camel Interrogation - Triggered Request

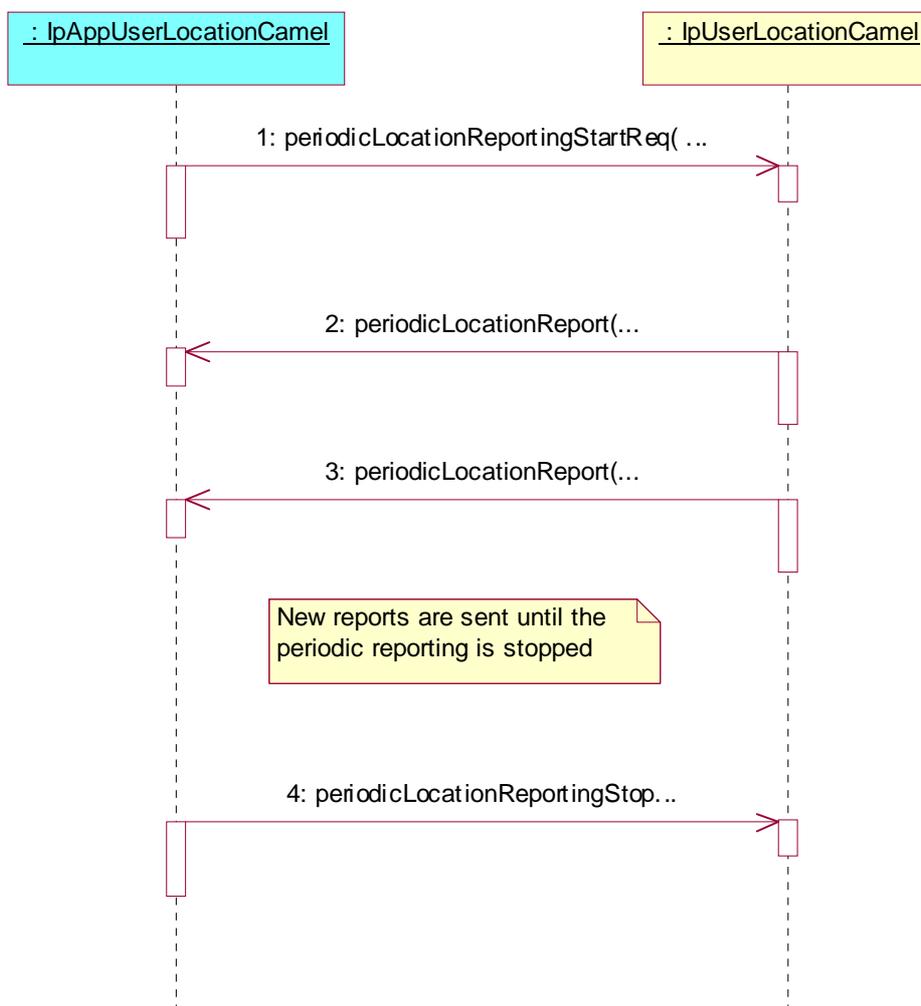
The following sequence diagram shows how an application requests triggered location reports from the User Location Camel service. When users location changes, the service reports this to the application.



- 1: This message is used to start triggered location reporting for one or several users.
- 2: When the trigger condition is fulfilled then this message passes the location of the affected user to its callback object.
- 3: This is repeated until the application stops triggered location reporting (see next message).
- 4: This message is used to stop triggered location reporting.

5.2.2 User Location Camel Interrogation - Periodic Request

The following sequence diagram shows how an application requests periodic location reports from the User Location Camel service.



- 1: This message is used to start periodic location reporting for one or several users.
- 2: This message passes the location of one or several users to its callback object.
- 3: This message passes the location of one or several users to its callback object.

This is repeated at regular intervals until the application stops periodic location reporting (see next message).

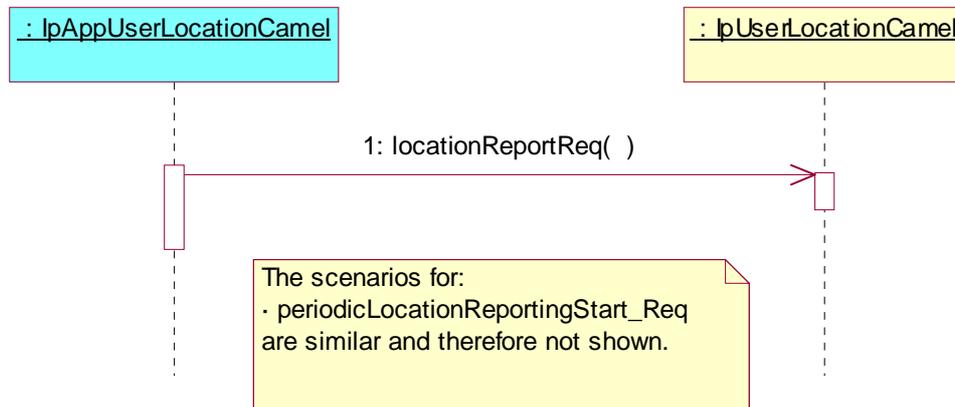
- 4: This message is used to stop periodic location reporting.

5.2.3 User Location Camel Interrogation - Parameter Error

The following sequence diagram show a scenario where the application is requesting a location report from the User Location Camel service but there is at least one error in the parameters that is detected by the service. The scenarios for:

- periodicLocationReportingStartReq

are similar and therefore not shown.



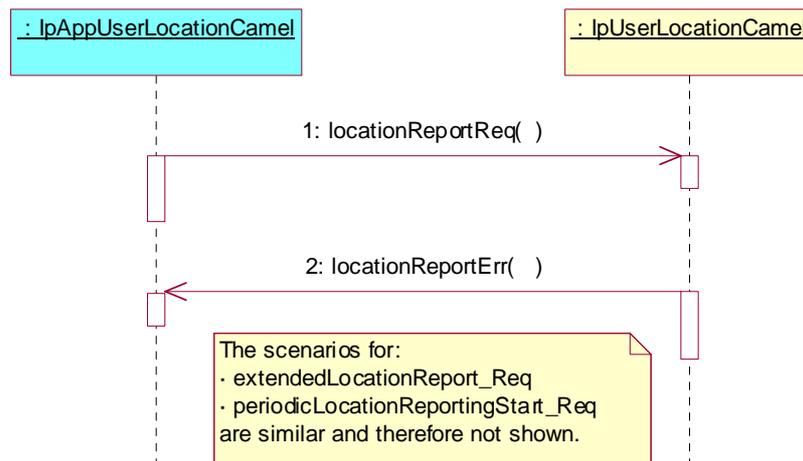
- 1: This message is used to request the location of one or several users, but the service returns an error and the execution of the request is aborted.

5.2.4 User Location Camel Interrogation - Network Error

The following sequence diagram shows a scenario where the application is requesting a location report from the User Location Camel service, but a network error occurs. The scenarios for:

- periodicLocationReportingStartReq

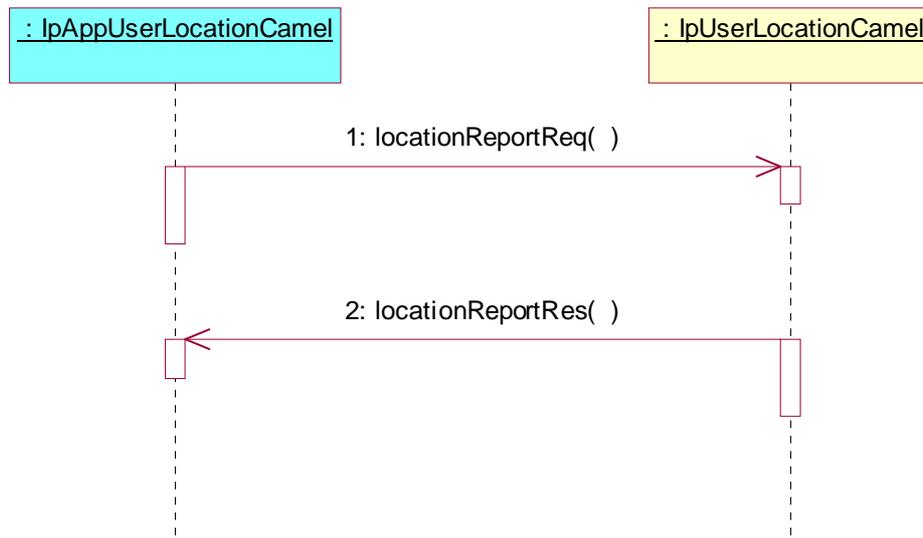
are similar and therefore not shown.



- 1: This message is used to request the location of one or several users.
- 2: This message passes information about the error in the location request from the network to the callback object.

5.2.5 User Location Camel Interrogation - Interactive Request

The following sequence diagram shows how an application requests a location report from the User Location Camel service.

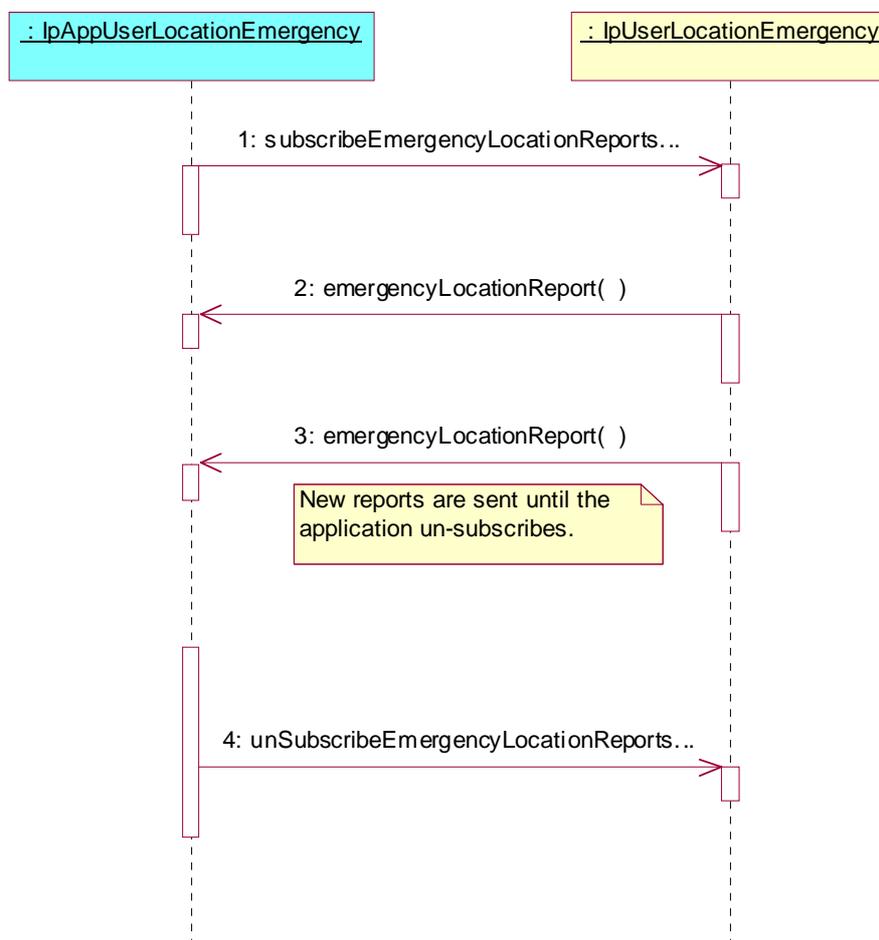


- 1: This message is used to request the location of one or several users.
- 2: This message passes the result of the location request for one or several users to its callback object.

5.3 User Location Emergency Sequence Diagrams

5.3.1 Subscription and Network Induced Location Reports

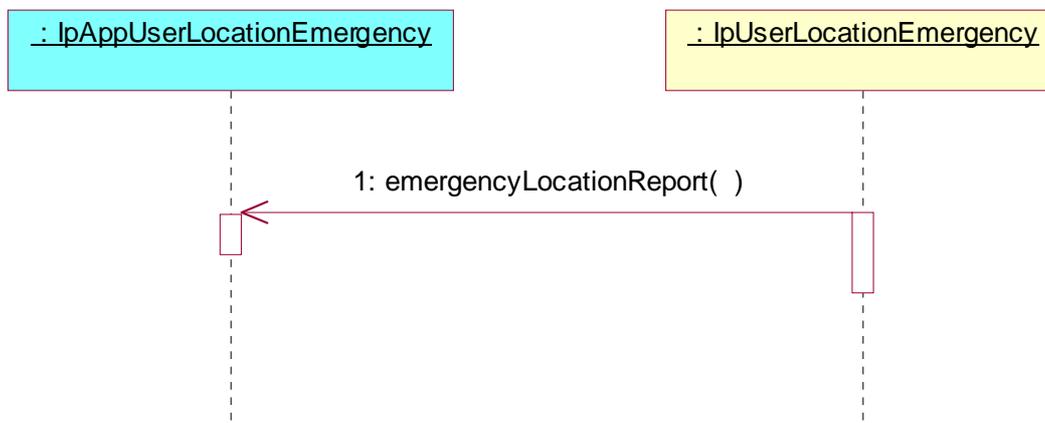
The following sequence diagram shows how an application subscribes to emergency location reports from the Emergency User Location service. When the User Location Emergency service receives Network Induced Location requests (triggered by emergency calls) it reports that to the application.



- 1: This message is used to subscribe to emergency location reports.
- 2: When the mobility service receives a Network Induced Location Request (triggered by an emergency service call), this message passes the location to its callback object.
- 3: This is repeated until the application cancels its subscription to emergency location reports (see next message).
- 4: This message is used to cancel the emergency user location report subscription.

5.3.2 Network Induced Location Reports

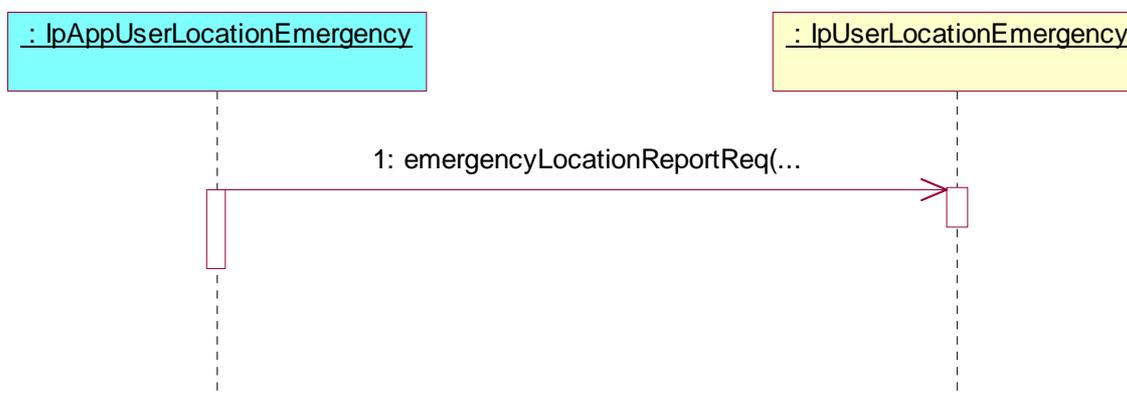
This sequence diagram is a simplified version of the previous diagram. This diagram shows only the part where an emergency call triggers a location report in the network and the location report is passed via the User Location Emergency service to a dedicated emergency application.



- 1: When the mobility service receives a Network Induced Location Request (triggered by an emergency service call), the available location information is passed to the emergency application in an emergency location report. Note that an emergency location report does not necessarily contain the user location, but in those cases, the user address, IMEI-address and/or a routing key (an identifier of the call) are passed to the application.

5.3.3 Interactive Request Parameter Error

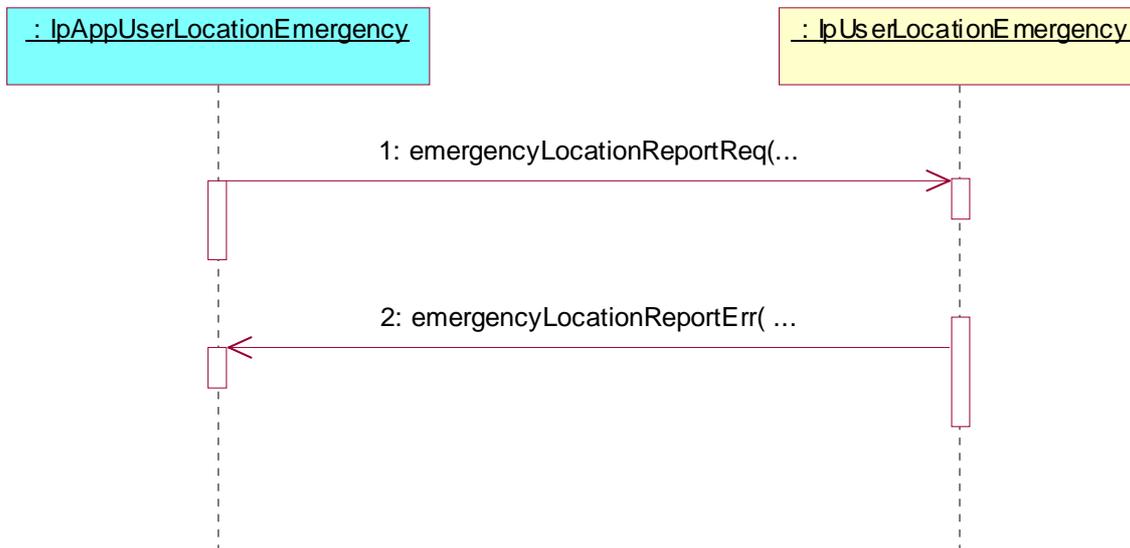
This sequence diagram shows the scenario, when an emergency application is requesting the location of a terminal based on information from a previous "incomplete" emergency location report, but the request cannot be processed because of a parameter error.



- 1: An emergency application makes an interactive request for a user location but the method returns an error that was discovered immediately.

5.3.4 Interactive Request Network Error

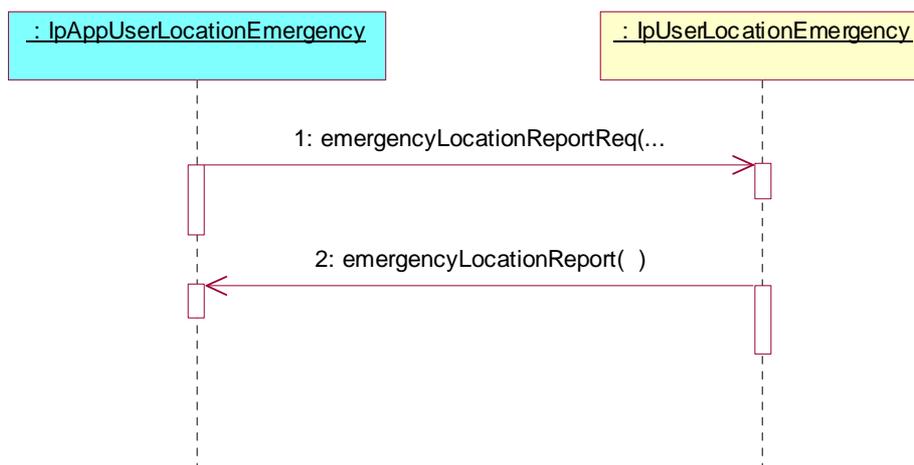
This sequence diagram shows the scenario where an emergency application requests the location of a terminal based on information from a previous "incomplete" emergency location report, but the request cannot be processed because of a network error.



- 1: An emergency application is making an interactive request for a user location.
- 2: Because of a network error, an error report is sent instead of the location report.

5.3.5 Interactive Location Request

This sequence diagram shows the scenario, when an emergency application requests the location of a terminal based on information from a previous incomplete emergency location report.

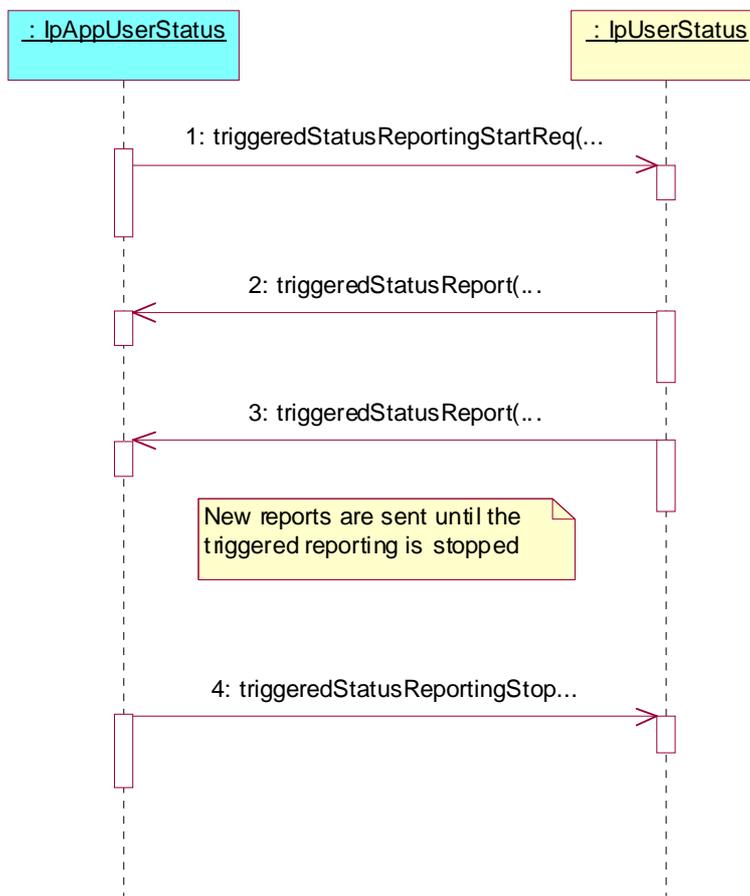


- 1: An emergency application makes an interactive request for a user location. This can sometimes be made with different parameters than the "ordinary" user location requests in the IpUserLocation interface, because a routing key and an IMSI might be used to identify the terminal (emergency call without SIM-card).
- 2: The emergency location information is reported back to the application.

5.4 User Status Sequence Diagrams

5.4.1 Triggered Reporting

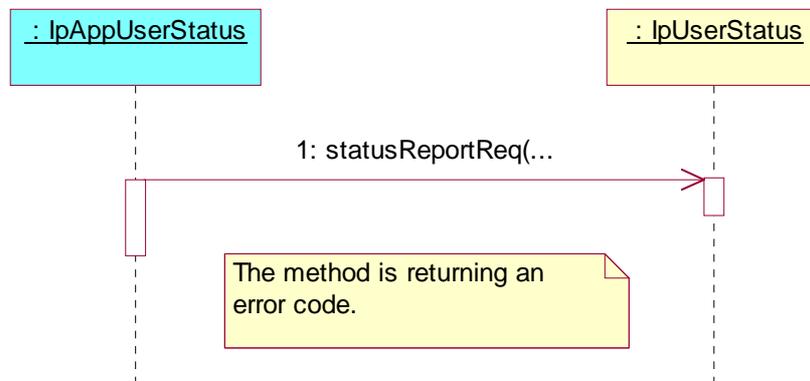
The following sequence diagram shows how an application requests triggered status reports from the Status Location service. When user's status changes, the service reports this to the application.



- 1: This message is used to start triggered status reporting for one or several users.
- 2: When a user's status changes, this message passes the status to its callback object.
- 3: This is repeated until the application stops triggered status reporting (see next message).
- 4: This message is used to stop triggered status reporting.

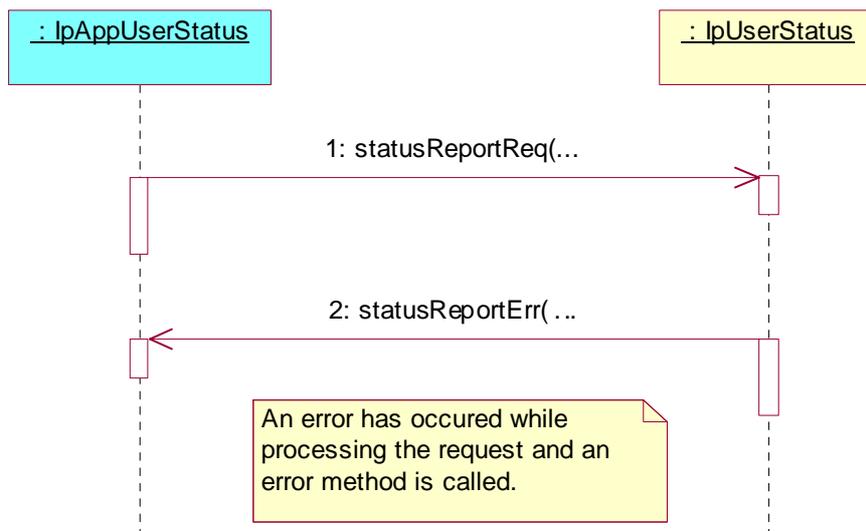
5.4.2 Interactive Request Parameter Error

The following sequence diagram shows, how an application requests a status report from the User Status service, but the service discovers an error and returns an error code.



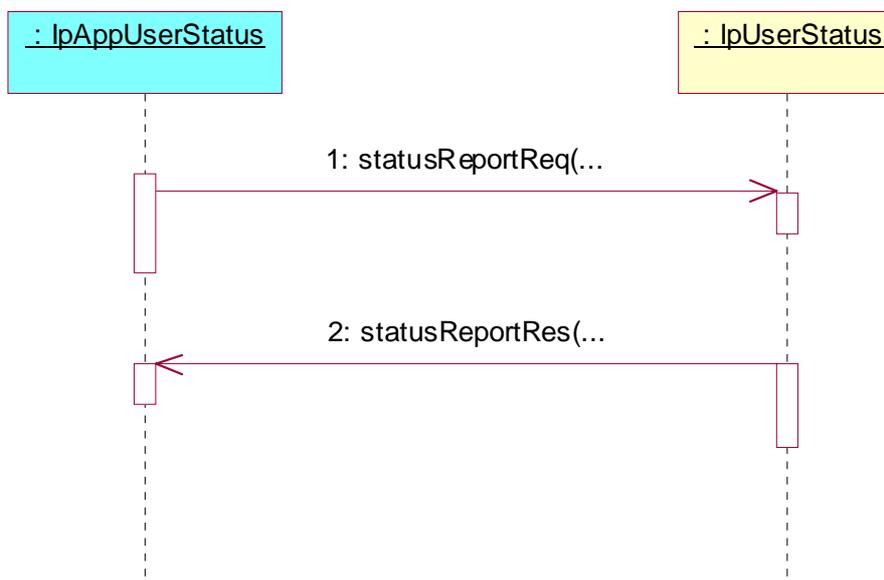
5.4.3 Interactive Request Network Error

The following sequence diagram shows, how an application requests a status report from the User Status service, but later, when the request is processed, the service discovers an error and calls an error method.



5.4.4 Interactive Request

The following sequence diagram shows how an application requests a status report from the User Status service.



- 1: This message is used to request the status of one or several users.
- 2: This message passes the result of the status request to its callback object.

6 Class Diagrams

6.1 User Location Class Diagrams

This class diagram shows the relationship between the interfaces in the User Location service. IpTriggeredUserLocation inherits from IpUserLocation, and IpAppTriggeredUserLocation inherits from IpAppUserLocation.

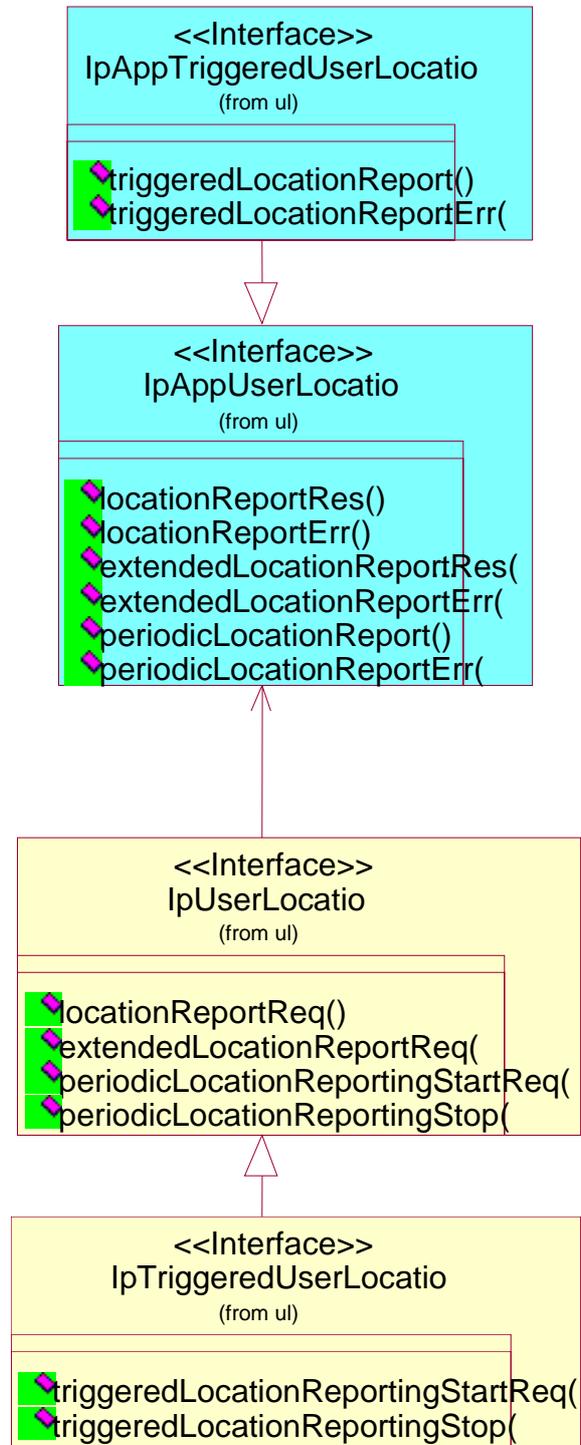


Figure 1: User Location Class Diagram

6.2 User Location Camel Class Diagrams

This class diagram shows the interfaces for the User Location Camel service.

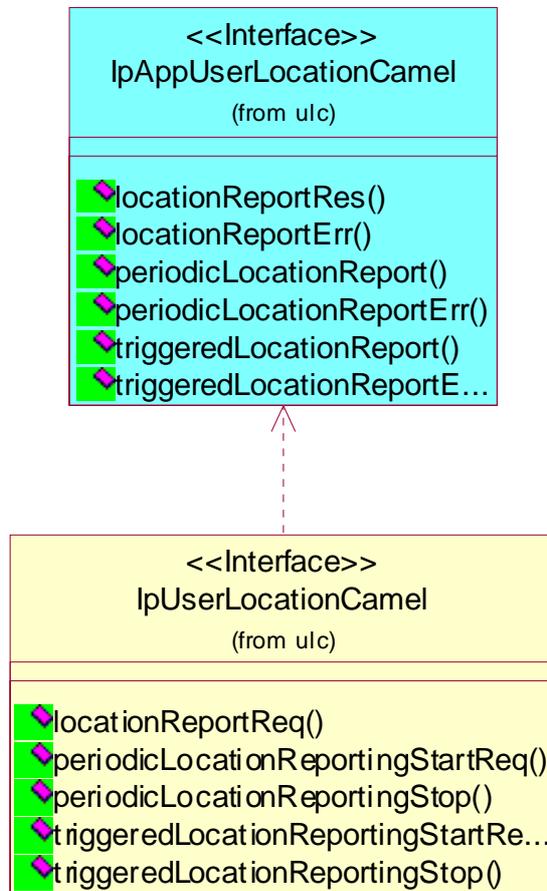


Figure 2: User Location Camel Class Diagram

6.3 User Location Emergency Class Diagrams

This class diagram shows the interfaces for the User Location Emergency service.

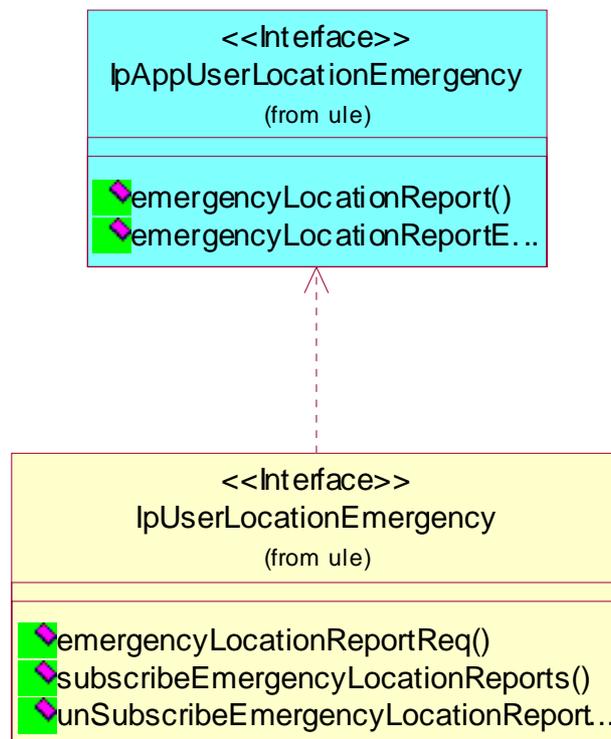


Figure 3: User Location Emergency Class Diagram

6.4 User Status Class Diagrams

This class diagram shows the interfaces for the User Status service.

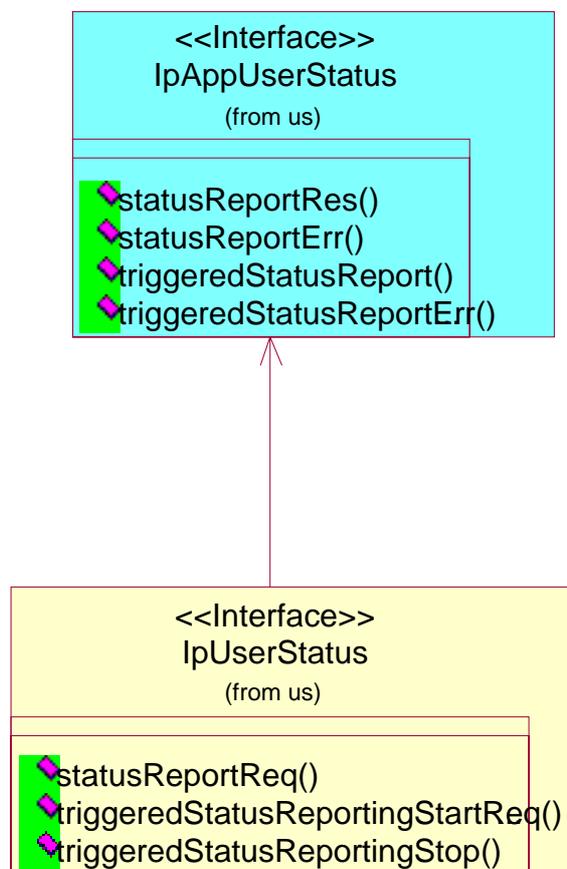


Figure 4: User Status Class Diagram

7 The Service Interface Specifications

7.1 Interface Specification Format

This clause defines the interfaces, methods and parameters that form a part of the API specification. The Unified Modelling Language (UML) is used to specify the interface classes. The general format of an interface specification is described in clause 7.1.1.

7.1.1 Interface Class

This shows a UML interface class description of the methods supported by that interface, and the relevant parameters and types. The Service and Framework interfaces for enterprise-based client applications are denoted by classes with name `Ip<name>`. The callback interfaces to the applications are denoted by classes with name `IpApp<name>`. For the interfaces between a Service and the Framework, the Service interfaces are typically denoted by classes with name `IpSvc<name>`, while the Framework interfaces are denoted by classes with name `IpFw<name>`.

7.1.2 Method descriptions

Each method (API method "call") is described. Both synchronous and asynchronous methods are used in the API. Asynchronous methods are identified by a "Req" suffix for a method request, and, if applicable, are served by asynchronous methods identified by either a "Res" or "Err" suffix for method results and errors, respectively. To handle responses and reports, the application or service developer must implement the relevant `IpApp<name>` or `IpSvc<name>` interfaces to provide the callback mechanism.

7.1.3 Parameter descriptions

Each method parameter and its possible values are described. Parameters described as "in" represent those that must have a value when the method is called. Those described as "out" are those that contain the return result of the method when the method returns.

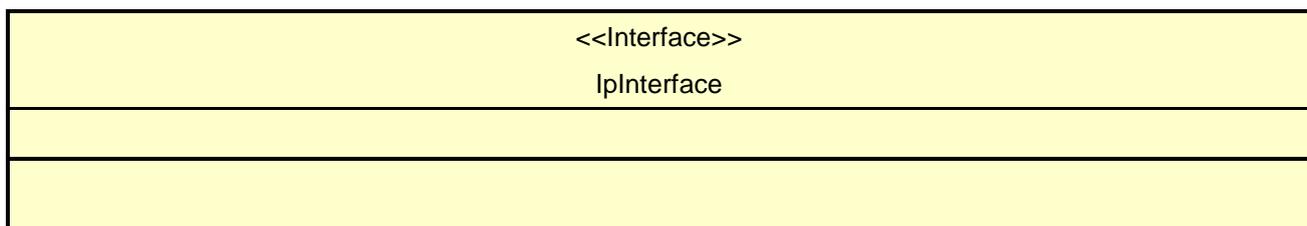
7.1.4 State Model

If relevant, a state model is shown to illustrate the states of the objects that implement the described interface.

7.2 Base Interface

7.2.1 Interface Class IpInterface

All application, framework and service interfaces inherit from the following interface. This API Base Interface does not provide any additional methods.



7.3 Service Interfaces

7.3.1 Overview

The Service Interfaces provide the interfaces into the capabilities of the underlying network - such as call control, user interaction, messaging, mobility and connectivity management.

The interfaces that are implemented by the services are denoted as "Service Interface". The corresponding interfaces that must be implemented by the application (e.g. for API callbacks) are denoted as "Application Interface".

7.4 Generic Service Interface

7.4.1 Interface Class IpService

Inherits from: IpInterface

All service interfaces inherit from the following interface.

<<Interface>> IpService
setCallback (appInterface: in IpInterfaceRef): void setCallbackWithSessionID (appInterface: in IpInterfaceRef, sessionID: in TpSessionID): void

Method

setCallback()

This method specifies the reference address of the callback interface that a service uses to invoke methods on the application. It is not allowed to invoke this method on an interface that uses SessionIDs.

Parameters

appInterface: in IpInterfaceRef

Specifies a reference to the application interface, which is used for callbacks

Raises

TpCommonExceptions, P_INVALID_INTERFACE_TYPE

Method

setCallbackWithSessionID()

This method specifies the reference address of the application's callback interface that a service uses for interactions associated with a specific session ID: e.g. a specific call, or call leg. It is not allowed to invoke this method on an interface that does not use SessionIDs.

Parameters

appInterface: in IpInterfaceRef

Specifies a reference to the application interface, which is used for callbacks

sessionID: in TpSessionID

Specifies the session for which the service can invoke the application's callback interface.

Raises

TpCommonExceptions, P_INVALID_SESSION_ID, P_INVALID_INTERFACE_TYPE

8 Mobility Interface Classes

8.1 User Location Interface Classes

The User Location service (UL) provides a general geographic location service. UL has functionality to allow applications to obtain the geographical location and the status of fixed, mobile and IP based telephony users.

UL is supplemented by User Location Camel service (ULC) to provide information about network related information. There is also some specialised functionality to handle emergency calls in the User Location Emergency service (ULE).

The UL service provides the IpUserLocation and IpTriggeredUserLocation interfaces. Most methods are asynchronous, in that they do not lock a thread into waiting whilst a transaction performs. In this way, the client machine can handle many more calls, than one that uses synchronous message calls. To handle responses and reports, the developer must implement IpAppUserLocation and IpAppTriggeredUserLocation interfaces to provide the callback mechanism.

When periodic or triggered location reporting is used, errors may be reported either when the recurrent reporting is requested, as an error per user in reports or in the corresponding err-method when the error concerns all subscribers in an assignment.

8.1.1 Interface Class IpUserLocation

Inherits from: IpService.

This interface is the "service manager" interface for the User Location Service.

The user location interface provides the management functions to the user location service. The application programmer can use this interface to obtain the geographical location of users.

<<Interface>> IpUserLocation
locationReportReq (appLocation: in IpAppUserLocationRef, users: in TpAddressSet): TpSessionID extendedLocationReportReq (appLocation: in IpAppUserLocationRef, users: in TpAddressSet, request: in TpLocationRequest): TpSessionID periodicLocationReportingStartReq (appLocation: in IpAppUserLocationRef, users: in TpAddressSet, request: in TpLocationRequest, reportingInterval: in TpDuration): TpSessionID periodicLocationReportingStop (stopRequest: in TpMobilityStopAssignmentData): void

*Method***locationReportReq()**

Request of a report on the location for one or several users.

Raises the following exceptions:

P_NO_CALLBACK_ADDRESS_SET

The requested method has been refused, because no callback address is set.

P_RESOURCES_UNAVAILABLE

The required resources in the network are not available. The application may try to invoke the method at a later time.

P_UNKNOWN_SUBSCRIBER

The end-user is not subscribed to the application.

P_APPLICATION_NOT_ACTIVATED

The end-user has de-activated the application.

P_INFORMATION_NOT_AVAILABLE

The request violates the end-user's privacy setting.

Returns: assignmentId

Specifies the assignment ID of the location-report request.

Parameters

appLocation: in IpAppUserLocationRef

Specifies the application interface for callbacks from the User Location service.

users: in TpAddressSet

Specifies the user(s) for which the location shall be reported.

Returns

TpSessionID

Raises

**TpCommonExceptions, P_APPLICATION_NOT_ACTIVATED,
P_INFORMATION_NOT_AVAILABLE, P_UNKNOWN_SUBSCRIBER,
P_INVALID_INTERFACE_TYPE**

*Method***extendedLocationReportReq()**

Advanced request of report on the location for one or several users.

Raises the following exceptions:

P_NO_CALLBACK_ADDRESS_SET

The requested method has been refused, because no callback address is set.

P_RESOURCES_UNAVAILABLE

The required resources in the network are not available. The application may try to invoke the method at a later time.

P_UNKNOWN_SUBSCRIBER

The end-user is not subscribed to the application.

P_APPLICATION_NOT_ACTIVATED

The end-user has de-activated the application.

P_INFORMATION_NOT_AVAILABLE

The request violates the end-user's privacy setting.

Returns: assignmentId

Specifies the assignment ID of the extended location-report request.

Parameters

appLocation: in IpAppUserLocationRef

Specifies the application interface for callbacks from the User Location service.

users: in TpAddressSet

Specifies the user(s) for which the location shall be reported

request: in TpLocationRequest

Specifies among others the requested location type, accuracy, response time and priority.

Returns

TpSessionID

Raises

TpCommonExceptions, P_APPLICATION_NOT_ACTIVATED, P_REQUESTED_ACCURACY_CANNOT_BE_DELIVERED, P_REQUESTED_RESPONSE_TIME_CANNOT_BE_DELIVERED, P_UNKNOWN_SUBSCRIBER, P_INFORMATION_NOT_AVAILABLE, P_INVALID_INTERFACE_TYPE

Method

periodicLocationReportingStartReq()

Request of periodic reports on the location for one or several users.

Raises the following exceptions:

P_NO_CALLBACK_ADDRESS_SET

The requested method has been refused, because no callback address is set.

P_RESOURCES_UNAVAILABLE

The required resources in the network are not available. The application may try to invoke the method at a later time.

P_UNKNOWN_SUBSCRIBER

The end-user is not subscribed to the application.

P_APPLICATION_NOT_ACTIVATED

The end-user has de-activated the application.

P_INFORMATION_NOT_AVAILABLE

The request violates the end-user's privacy setting.

Returns: assignmentId

Specifies the assignment ID of the periodic location-reporting request.

Parameters

appLocation: in IpAppUserLocationRef

Specifies the application interface for callbacks from the User Location service.

users: in TpAddressSet

Specifies the user(s) for which the location shall be reported.

request: in TpLocationRequest

Specifies among others the requested location type, accuracy, response time and priority.

reportingInterval: in TpDuration

Specifies the requested interval in seconds between the reports.

Returns

TpSessionID

Raises

**TpCommonExceptions, P_INVALID_REPORTING_INTERVAL,
P_REQUESTED_ACCURACY_CANNOT_BE_DELIVERED,
P_REQUESTED_RESPONSE_TIME_CANNOT_BE_DELIVERED, P_UNKNOWN_SUBSCRIBER,
P_APPLICATION_NOT_ACTIVATED, P_INFORMATION_NOT_AVAILABLE,
P_INVALID_INTERFACE_TYPE**

Method

periodicLocationReportingStop()

Termination of periodic reports on the location for one or several users.

Raises the following exceptions:

P_INVALID_ASSIGNMENT_ID

The assignment ID does not correspond to one of a valid assignment.

Parameters

stopRequest: in TpMobilityStopAssignmentData

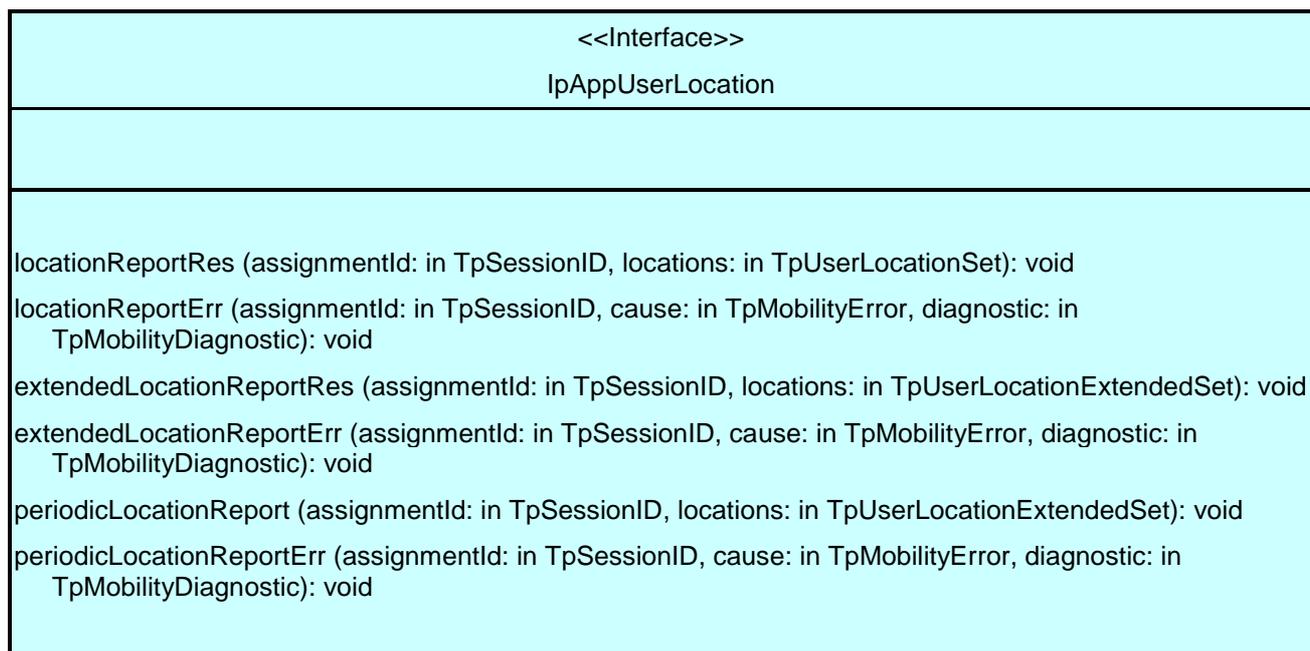
Specifies how the assignment shall be stopped, i.e. if whole or just parts of the assignment should be stopped.

*Raises***TpCommonExceptions, P_INVALID_ASSIGNMENT_ID**

8.1.2 Interface Class IpAppUserLocation

Inherits from: IpInterface.

The user-location application interface is implemented by the client application developer and is used to handle user location request responses.

*Method***locationReportRes ()**

A report containing locations for one or several users is delivered.

*Parameters***assignmentId: in TpSessionID**

Specifies the assignment ID of the location-report request.

locations: in TpUserLocationSet

Specifies the location(s) of one or several users.

*Method***locationReportErr ()**

This method indicates that the location report request has failed.

*Parameters***assignmentId: in TpSessionID**

Specifies the assignment ID of the failed location report request.

cause: in TpMobilityError

Specifies the error that led to the failure.

diagnostic: in TpMobilityDiagnostic

Specifies additional information about the error that led to the failure.

*Method***extendedLocationReportRes()**

A report containing extended location information for one or several users is delivered.

*Parameters***assignmentId: in TpSessionID**

Specifies the assignment ID of the extended location-report request.

locations: in TpUserLocationExtendedSet

Specifies the location(s) of one or several users.

*Method***extendedLocationReportErr()**

This method indicates that the extended location report request has failed.

*Parameters***assignmentId: in TpSessionID**

Specifies the assignment ID of the failed extended location report request.

cause: in TpMobilityError

Specifies the error that led to the failure.

diagnostic: in TpMobilityDiagnostic

Specifies additional information about the error that led to the failure.

*Method***periodicLocationReport()**

A report containing periodic location information for one or several users is delivered.

*Parameters***assignmentId: in TpSessionID**

Specifies the assignment ID of the periodic location-reporting request.

locations: in TpUserLocationExtendedSet

Specifies the location(s) of one or several users.

*Method***periodicLocationReportErr()**

This method indicates that a requested periodic location report has failed. Note that errors only concerning individual users are reported in the ordinary periodicLocationReport() message.

*Parameters***assignmentId: in TpSessionID**

Specifies the assignment ID of the failed periodic location reporting start request.

cause: in TpMobilityError

Specifies the error that led to the failure.

diagnostic: in TpMobilityDiagnostic

Specifies additional information about the error that led to the failure.

8.1.3 Interface Class IpTriggeredUserLocation

Inherits from: IpUserLocation.

This interface can be used as an extended version of the User Location: Service Interface.

The triggered user location interface represents the interface to the triggered user location functions. The application programmer can use this interface to request user location reports that are triggered by location change.

<<Interface>> IpTriggeredUserLocation
triggeredLocationReportingStartReq (appLocation: in IpAppTriggeredUserLocationRef, users: in TpAddressSet, request: in TpLocationRequest, triggers: in TpLocationTriggerSet): TpSessionID triggeredLocationReportingStop (stopRequest: in TpMobilityStopAssignmentData): void

*Method***triggeredLocationReportingStartReq()**

Request for user location reports when the location is changed (reports are triggered by location change).

Returns: assignmentId

Specifies the assignment ID of the triggered location-reporting request.

*Parameters***appLocation: in IpAppTriggeredUserLocationRef**

Specifies the application interface for callbacks from the User Location service.

users: in TpAddressSet

Specifies the user(s) for which the location shall be reported.

request: in TpLocationRequest

Specifies among others the requested location type, accuracy, response time and priority.

triggers: in TpLocationTriggerSet

Specifies the trigger conditions.

*Returns***TpSessionID***Raises***TpCommonExceptions, P_REQUESTED_ACCURACY_CANNOT_BE_DELIVERED,
P_REQUESTED_RESPONSE_TIME_CANNOT_BE_DELIVERED,
P_TRIGGER_CONDITIONS_NOT_SUBSCRIBED, P_UNKNOWN_SUBSCRIBER,
P_APPLICATION_NOT_ACTIVATED, P_INFORMATION_NOT_AVAILABLE,
P_INVALID_INTERFACE_TYPE***Method***triggeredLocationReportingStop()**

Stop triggered user location reporting.

*Parameters***stopRequest: in TpMobilityStopAssignmentData**

Specifies how the assignment shall be stopped, i.e. if whole or just parts of the assignment should be stopped.

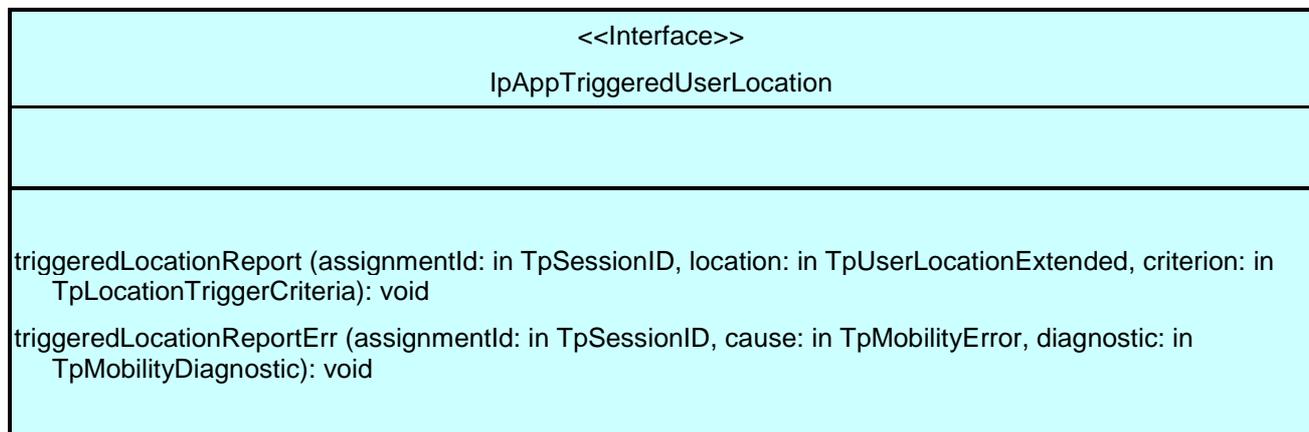
*Raises***TpCommonExceptions, P_INVALID_ASSIGNMENT_ID**

8.1.4 Interface Class IpAppTriggeredUserLocation

Inherits from: IpAppUserLocation.

This interface must be used as a specialised version of the User Location: Application Interface if the Triggered User Location: Service Interface is used.

The triggered user location application interface is implemented by the client application developer and is used to handle triggered location reports.



Method

triggeredLocationReport ()

A triggered report containing location for a user is delivered.

Parameters

assignmentId: in TpSessionID

Specifies the assignment ID of the triggered location-reporting request.

location: in TpUserLocationExtended

Specifies the location of the user.

criterion: in TpLocationTriggerCriteria

Specifies the criterion that triggered the report.

Method

triggeredLocationReportErr ()

This method indicates that a requested triggered location report has failed. Note that errors only concerning individual users are reported in the ordinary triggeredLocationReport() message.

Parameters

assignmentId: in TpSessionID

Specifies the assignment ID of the failed triggered location reporting start request.

cause: in TpMobilityError

Specifies the error that led to the failure.

diagnostic: in TpMobilityDiagnostic

Specifies additional information about the error that led to the failure.

8.2 User Location Camel Interface Classes

The ULC provides location information, based on network-related information, rather than the geographical co-ordinates that can be retrieved via the general User Location Service.

Using the ULC functions, an application programmer can request the VLR Number, the location Area Identification and the Cell Global Identification and other mobile-telephony-specific location information

The ULC provides the IpUserLocationCamel interface. Most methods are asynchronous, in that they do not lock a thread into waiting whilst a transaction performs. In this way, the client machine can handle many more calls, than one that uses synchronous message calls. To handle responses and reports, the developer must implement IpAppUserLocationCamel interface to provide the callback mechanism.

8.2.1 Interface Class IpUserLocationCamel

Inherits from: IpService.

This interface is the "service manager" interface for ULC.

<<Interface>> IpUserLocationCamel
locationReportReq (appLocationCamel: in IpAppUserLocationCamelRef, users: in TpAddressSet): TpSessionID periodicLocationReportingStartReq (appLocationCamel: in IpAppUserLocationCamelRef, users: in TpAddressSet, reportingInterval: in TpDuration): TpSessionID periodicLocationReportingStop (stopRequest: in TpMobilityStopAssignmentData): void triggeredLocationReportingStartReq (appLocationCamel: in IpAppUserLocationCamelRef, users: in TpAddressSet, trigger: in TpLocationTriggerCamel): TpSessionID triggeredLocationReportingStop (stopRequest: in TpMobilityStopAssignmentData): void

Method

locationReportReq()

Request for mobile-related location information on one or several camel users.

Raises the following exceptions:

P_NO_CALLBACK_ADDRESS_SET

The requested method has been refused, because no callback address is set.

P_RESOURCES_UNAVAILABLE

The required resources in the network are not available. The application may try to invoke the method at a later time.

P_UNKNOWN_SUBSCRIBER

The end-user is not subscribed to the application.

P_APPLICATION_NOT_ACTIVATED

The end-user has de-activated the application.

P_INFORMATION_NOT_AVAILABLE

The request violates the end-user's privacy setting.

Returns: assignmentId

Specifies the assignment ID of the location-report request.

Parameters

appLocationCamel: in IpAppUserLocationCamelRef

Specifies the application interface for callbacks from the User Location Camel service.

users: in TpAddressSet

Specifies the user(s) for which the location shall be reported.

Returns

TpSessionID

Raises

TpCommonExceptions, P_UNKNOWN_SUBSCRIBER, P_APPLICATION_NOT_ACTIVATED, P_INFORMATION_NOT_AVAILABLE, P_INVALID_INTERFACE_TYPE

Method

periodicLocationReportingStartReq()

Request for periodic mobile location reports on one or several users.

Raises the following exceptions:

P_NO_CALLBACK_ADDRESS_SET

The requested method has been refused, because no callback address is set.

P_RESOURCES_UNAVAILABLE

The required resources in the network are not available. The application may try to invoke the method at a later time.

P_UNKNOWN_SUBSCRIBER

The end-user is not subscribed to the application.

P_APPLICATION_NOT_ACTIVATED

The end-user has de-activated the application.

P_INFORMATION_NOT_AVAILABLE

The request violates the end-user's privacy setting.

Returns: assignmentId

Specifies the assignment ID of the periodic location-reporting request.

Parameters

appLocationCamel: in IpAppUserLocationCamelRef

Specifies the application interface for callbacks from the User Location Camel service.

users: in TpAddressSet

Specifies the user(s) for which the location shall be reported.

reportingInterval: in TpDuration

Specifies the requested interval in seconds between the reports.

Returns

TpSessionID

Raises

**TpCommonExceptions, P_INVALID_REPORTING_INTERVAL,
P_REQUESTED_ACCURACY_CANNOT_BE_DELIVERED,
P_REQUESTED_RESPONSE_TIME_CANNOT_BE_DELIVERED, P_UNKNOWN_SUBSCRIBER,
P_APPLICATION_NOT_ACTIVATED, P_INFORMATION_NOT_AVAILABLE,
P_INVALID_INTERFACE_TYPE**

Method

periodicLocationReportingStop()

This method stops the sending of periodic mobile location reports for one or several users.

Raises the following exceptions:

P_INVALID_ASSIGNMENT_ID

The assignment ID does not correspond to one of a valid assignment.

Parameters

stopRequest: in TpMobilityStopAssignmentData

Specifies how the assignment shall be stopped, i.e. if whole or just parts of the assignment should be stopped.

Raises

TpCommonExceptions, P_INVALID_ASSIGNMENT_ID

Method

triggeredLocationReportingStartReq()

Request for user location reports, containing mobile related information, when the location is changed (the report is triggered by the location change).

Raises the following exceptions:

P_NO_CALLBACK_ADDRESS_SET

The requested method has been refused, because no callback address is set.

P_RESOURCES_UNAVAILABLE

The required resources in the network are not available. The application may try to invoke the method at a later time.

P_UNKNOWN_SUBSCRIBER

The end-user is not subscribed to the application.

P_APPLICATION_NOT_ACTIVATED

The end-user has de-activated the application.

P_INFORMATION_NOT_AVAILABLE

The request violates the end-user's privacy setting.

Returns: `assignmentId`

Specifies the assignment ID of the triggered location-reporting request.

Parameters

appLocationCamel: in IpAppUserLocationCamelRef

Specifies the application interface for callbacks from the User Location Camel service.

users: in TpAddressSet

Specifies the user(s) for which the location shall be reported.

trigger: in TpLocationTriggerCamel

Specifies the trigger conditions.

Returns

TpSessionID

Raises

TpCommonExceptions, P_UNKNOWN_SUBSCRIBER, P_APPLICATION_NOT_ACTIVATED, P_INFORMATION_NOT_AVAILABLE, P_INVALID_INTERFACE_TYPE

Method

triggeredLocationReportingStop()

Request that triggered mobile location reporting should stop.

Raises the following exceptions:

P_INVALID_ASSIGNMENT_ID

The assignment ID does not correspond to one of a valid assignment.

*Parameters***stopRequest: in TpMobilityStopAssignmentData**

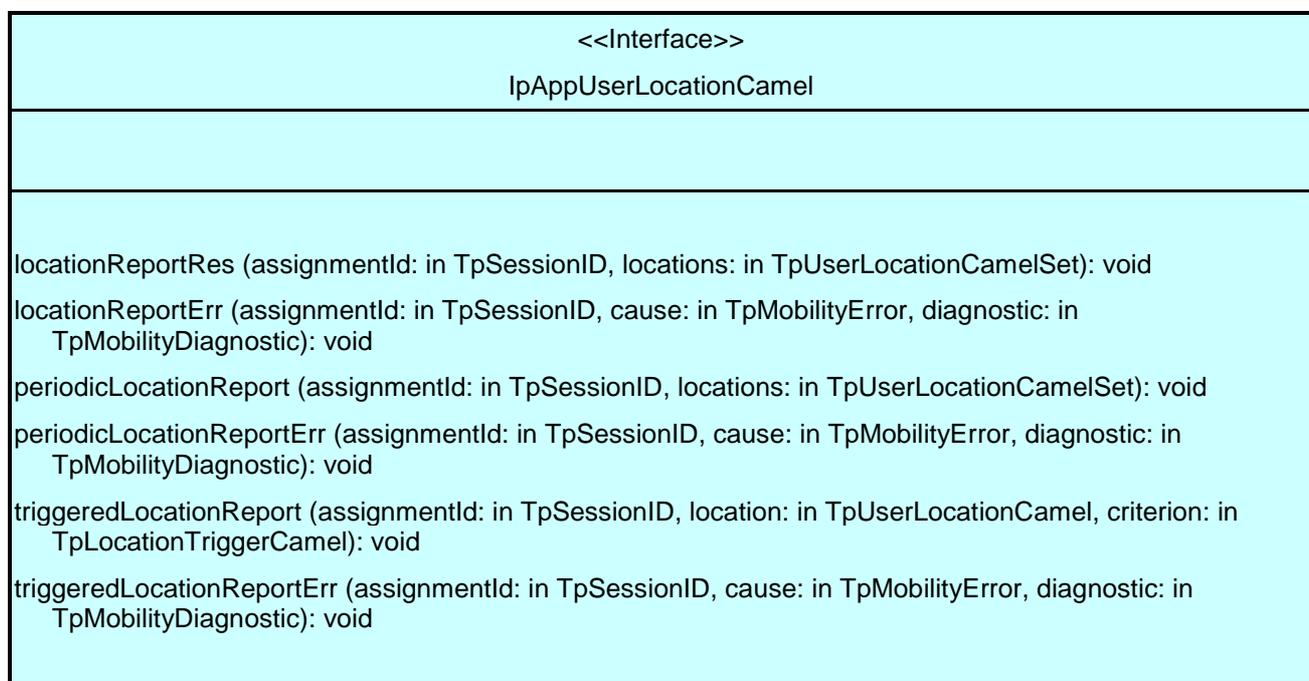
Specifies how the assignment shall be stopped, i.e. if whole or just parts of the assignment should be stopped.

*Raises***TpCommonExceptions, P_INVALID_ASSIGNMENT_ID**

8.2.2 Interface Class IpAppUserLocationCamel

Inherits from: IpInterface.

The user location Camel application interface is implemented by the client application developer and is used to handle location reports that are specific for mobile telephony users.

*Method***locationReportRes()**

Delivery of a mobile location report. The report is containing mobile-related location information for one or several users.

*Parameters***assignmentId: in TpSessionID**

Specifies the assignment ID of the location-report request.

locations: in TpUserLocationCamelSet

Specifies the location(s) of one or several users.

*Method***locationReportErr()**

This method indicates that the location report request has failed.

Parameters

assignmentId: in TpSessionID

Specifies the assignment ID of the failed location report request.

cause: in TpMobilityError

Specifies the error that led to the failure.

diagnostic: in TpMobilityDiagnostic

Specifies additional information about the error that led to the failure.

*Method***periodicLocationReport()**

Periodic delivery of mobile location reports. The reports are containing mobile-related location information for one or several users.

Parameters

assignmentId: in TpSessionID

Specifies the assignment ID of the periodic location-reporting request.

locations: in TpUserLocationCamelSet

Specifies the location(s) of one or several users.

*Method***periodicLocationReportErr()**

This method indicates that a requested periodic location report has failed. Note that errors only concerning individual users are reported in the ordinary periodicLocationReport() message.

Parameters

assignmentId: in TpSessionID

Specifies the assignment ID of the failed periodic location reporting start request.

cause: in TpMobilityError

Specifies the error that led to the failure.

diagnostic: in TpMobilityDiagnostic

Specifies additional information about the error that led to the failure.

*Method***triggeredLocationReport()**

Delivery of a report that is indicating that the user's mobile location has changed.

*Parameters***assignmentId: in TpSessionID**

Specifies the assignment ID of the triggered location-reporting request.

location: in TpUserLocationCamel

Specifies the location of the user.

criterion: in TpLocationTriggerCamel

Specifies the criterion that triggered the report.

*Method***triggeredLocationReportErr ()**

This method indicates that a requested triggered location report has failed. Note that errors only concerning individual users are reported in the ordinary triggeredLocationReport() message.

*Parameters***assignmentId: in TpSessionID**

Specifies the assignment ID of the failed triggered location reporting start request.

cause: in TpMobilityError

Specifies the error that led to the failure.

diagnostic: in TpMobilityDiagnostic

Specifies additional information about the error that led to the failure.

8.3 User Location Emergency Interface Classes

In the case of an emergency call, the network may locate the caller automatically. The resulting location is sent directly to an application that is dedicated to handle emergency user location. If the dedicated emergency call application is using the API, the location is sent to the application using a callback method in the IpAppUserLocationEmergency interface.

However, the network does not always send the location immediately (probably when the location procedure is not finished when the call is set up). In this case the network will send an identifier of the caller that can be used to locate the caller using the interface IpUserLocationEmergency.

8.3.1 Interface Class IpUserLocationEmergency

Inherits from: IpService.

The application programmer can use this interface to obtain the location of users who initiate emergency calls.

<<Interface>> IpUserLocationEmergency
emergencyLocationReportReq (appEmergencyLocation: in IpAppUserLocationEmergencyRef, request: in TpUserLocationEmergencyRequest): TpSessionID subscribeEmergencyLocationReports (appEmergencyLocation: in IpAppUserLocationEmergencyRef): TpSessionID unsubscribeEmergencyLocationReports (assignmentId: in TpSessionID): void

Method

emergencyLocationReportReq ()

Request of report on the location for one user that is making an emergency call.

Raises the following exceptions:

P_NO_CALLBACK_ADDRESS_SET

The requested method has been refused, because no callback address is set.

P_RESOURCES_UNAVAILABLE

The required resources in the network are not available. The application may try to invoke the method at a later time.

Returns: assignmentId

Specifies the assignment ID of the emergency location-report request.

Parameters

appEmergencyLocation: in IpAppUserLocationEmergencyRef

Specifies the application interface for callbacks from the User Location Emergency service.

request: in TpUserLocationEmergencyRequest

Specifies among others the identity of the user or terminal, requested location type, accuracy, response time and priority.

Returns **TpSessionID** *Raises* **TpCommonExceptions, P_UNKNOWN_SUBSCRIBER, P_INFORMATION_NOT_AVAILABLE, P_APPLICATION_NOT_ACTIVATED, P_INVALID_INTERFACE_TYPE** *Method* **subscribeEmergencyLocationReports()**

Subscribe to network initiated emergency user location reports.

A method parameter has an invalid value.

 P_NO_CALLBACK_ADDRESS_SET

The requested method has been refused, because no callback address is set.

 P_RESOURCES_UNAVAILABLE

The required resources in the network are not available. The application may try to invoke the method at a later time.

Returns: assignmentId

Specifies the assignment ID of the subscription.

Parameters **appEmergencyLocation: in IpAppUserLocationEmergencyRef**

Specifies the application interface for callbacks from the User Location Emergency service.

Returns **TpSessionID** *Raises* **TpCommonExceptions, P_INVALID_INTERFACE_TYPE** *Method* **unsubscribeEmergencyLocationReports()**

This method cancels a subscription to network initiated emergency user location reports.

Raises the following exceptions:

 P_INVALID_ASSIGNMENT_ID

The assignment ID does not correspond to one of a valid assignment.

Parameters **assignmentId: in TpSessionID**

Specifies the assignment ID of the subscription.

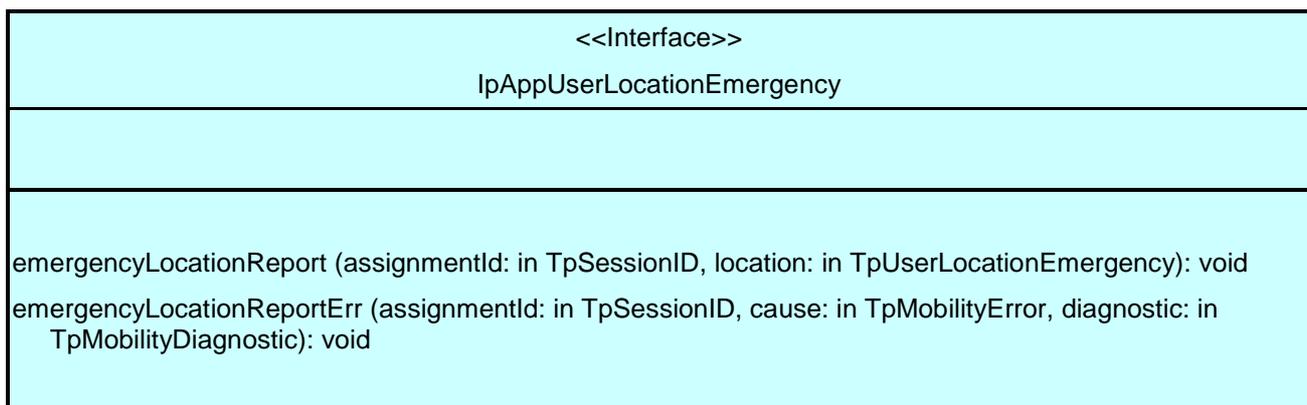
Raises

TpCommonExceptions, P_INVALID_ASSIGNMENT_ID

8.3.2 Interface Class IpAppUserLocationEmergency

Inherits from: IpInterface.

The user-location emergency application interface is implemented by the client application developer and is used to handle emergency user location reports.



Method

emergencyLocationReport ()

Delivery of an emergency user location report.

Parameters

assignmentId: in TpSessionID

Specifies the assignment ID of the emergency location-report request or the emergency location report subscription.

location: in TpUserLocationEmergency

Specifies the identity and location of the user that makes the emergency call.

Method

emergencyLocationReportErr ()

This method indicates that the emergency location report request has failed.

Parameters

assignmentId: in TpSessionID

Specifies the assignment ID of the failed emergency location report request.

cause: in TpMobilityError

Specifies the error that led to the failure.

diagnostic: in TpMobilityDiagnostic

Specifies additional information about the error that led to the failure.

8.4 User Status Interface Classes

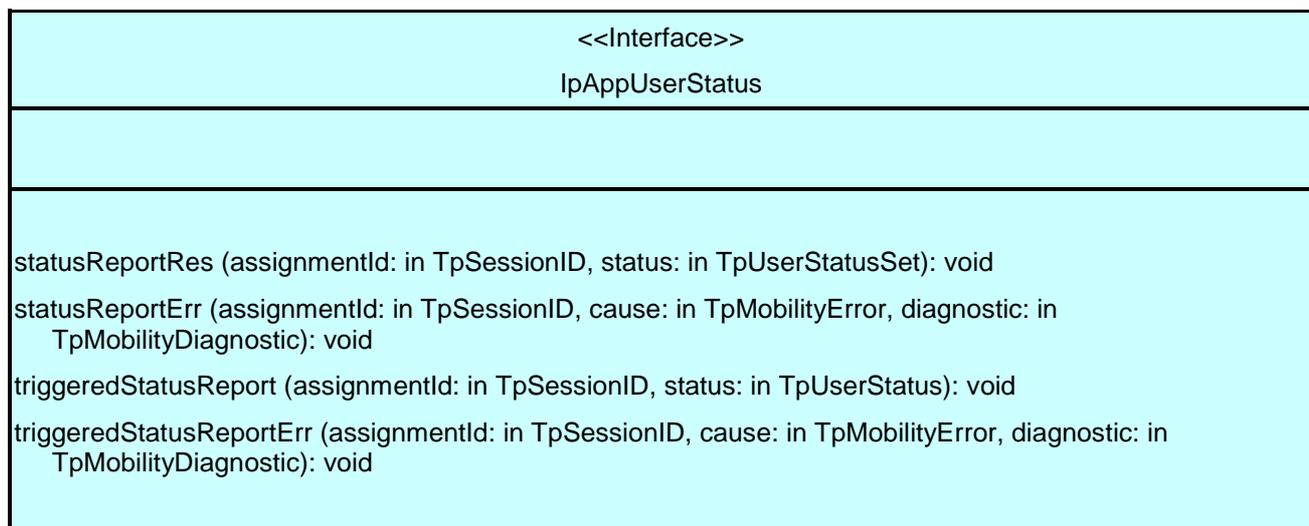
The User Status Service (US) provides a general user status service. US allow applications to obtain the status of fixed, mobile and IP-based telephony users.

The US provides the IpUserStatus interface. Most methods are asynchronous, in that they do not lock a thread into waiting whilst a transaction performs. In this way, the client machine can handle many more calls, than one that uses synchronous message calls. To handle responses and reports, the developer must implement IpAppUserStatus interface to provide the callback mechanism.

8.4.1 Interface Class IpAppUserStatus

Inherits from: IpInterface.

The user-status application interface is implemented by the client application developer and is used to handle user status reports.



Method

statusReportRes ()

Delivery of a report, that is containing one or several user's status.

Parameters

assignmentId: in TpSessionID

Specifies the assignment ID of the status-report request.

status: in TpUserStatusSet

Specifies the status of one or several users.

*Method***statusReportErr()**

This method indicates that the status report request has failed.

Parameters

assignmentId: in TpSessionID

Specifies the assignment ID of the failed status report request.

cause: in TpMobilityError

Specifies the error that led to the failure.

diagnostic: in TpMobilityDiagnostic

Specifies additional information about the error that led to the failure.

*Method***triggeredStatusReport()**

Delivery of a report that is indicating that a user's status has changed.

Parameters

assignmentId: in TpSessionID

Specifies the assignment ID of the triggered status-reporting request.

status: in TpUserStatus

Specifies the status of the user.

*Method***triggeredStatusReportErr()**

This method indicates that a requested triggered status reporting has failed. Note that errors only concerning individual users are reported in the ordinary triggeredStatusReport() message.

Parameters

assignmentId: in TpSessionID

Specifies the assignment ID of the failed triggered status reporting start request.

cause: in TpMobilityError

Specifies the error that led to the failure.

diagnostic: in TpMobilityDiagnostic

Specifies additional information about the error that led to the failure.

8.4.2 Interface Class IpUserStatus

Inherits from: IpService.

The application programmer can use this interface to obtain the status of fixed, mobile and IP-based telephony users.

<<Interface>> IpUserStatus
<pre> statusReportReq (appStatus: in IpAppUserStatusRef, users: in TpAddressSet): TpSessionID triggeredStatusReportingStartReq (appStatus: in IpAppUserStatusRef, users: in TpAddressSet): TpSessionID triggeredStatusReportingStop (stopRequest: in TpMobilityStopAssignmentData): void </pre>

Method

statusReportReq ()

Request for a report on the status of one or several users.

Raises the following exceptions:

P_NO_CALLBACK_ADDRESS_SET

The requested method has been refused, because no callback address is set.

P_RESOURCES_UNAVAILABLE

The required resources in the network are not available. The application may try to invoke the method at a later time.

Returns: assignmentId

Specifies the assignment ID of the status-report request.

Parameters

appStatus: in IpAppUserStatusRef

Specifies the application interface for callbacks from the User Status service.

users: in TpAddressSet

Specifies the user(s) for which the status shall be reported.

Returns

TpSessionID

Raises

TpCommonExceptions, P_UNKNOWN_SUBSCRIBER, P_INFORMATION_NOT_AVAILABLE, P_APPLICATION_NOT_ACTIVATED, P_INVALID_INTERFACE_TYPE

*Method***triggeredStatusReportingStartReq()**

Request for triggered status reports when one or several user's status is changed. The user status service will send a report when the status changes.

Raises the following exceptions:

P_NO_CALLBACK_ADDRESS_SET

The requested method has been refused, because no callback address is set.

P_RESOURCES_UNAVAILABLE

The required resources in the network are not available. The application may try to invoke the method at a later time.

Returns: assignmentId

Specifies the assignment ID of the triggered status-reporting request.

Parameters

appStatus: in IpAppUserStatusRef

Specifies the application interface for callbacks from the User Status service.

users: in TpAddressSet

Specifies the user(s) for which the status changes shall be reported.

Returns

TpSessionID

Raises

TpCommonExceptions, P_UNKNOWN_SUBSCRIBER, P_INFORMATION_NOT_AVAILABLE, P_APPLICATION_NOT_ACTIVATED, P_INVALID_INTERFACE_TYPE

*Method***triggeredStatusReportingStop()**

This method stops the sending of status reports for one or several users.

Raises the following exceptions:

P_INVALID_ASSIGNMENT_ID

The assignment ID does not correspond to one of a valid assignment.

Parameters

stopRequest: in TpMobilityStopAssignmentData

Specifies how the assignment shall be stopped, i.e. if whole or just parts of the assignment should be stopped.

Raises

TpCommonExceptions, P_INVALID_ASSIGNMENT_ID

9 State Transition Diagrams

9.1 User Location

There are no State Transition Diagrams for User Location.

9.2 User Location Camel

9.2.1 State Transition Diagrams for IpUserLocationCamel

During the signServiceAgreement a new user location interface reference is created, which is user as the initial point of contact for the application.

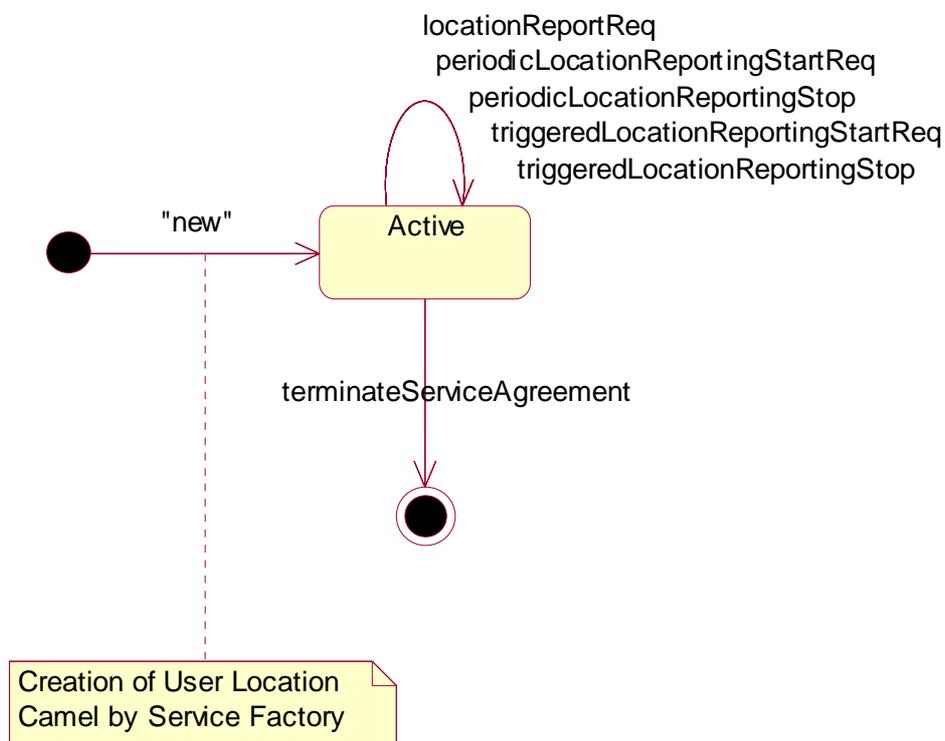


Figure 5: State Transition Diagram for User Location Camel

9.2.1.1 Active State

In this state, a relation between the Application and the Network User Location Service Capability Feature has been established. It allows the application to request a specific user location reports, subscribe to periodic user location reports or subscribe to triggers that generate location report when a location update occurs inside the current VLR area or when the user moves to another VLR area or both.

9.3 User Location Emergency

There are no State Transition Diagrams for User Location Emergency.

9.4 User Status

9.4.1 State Transition Diagrams for IpUserStatus

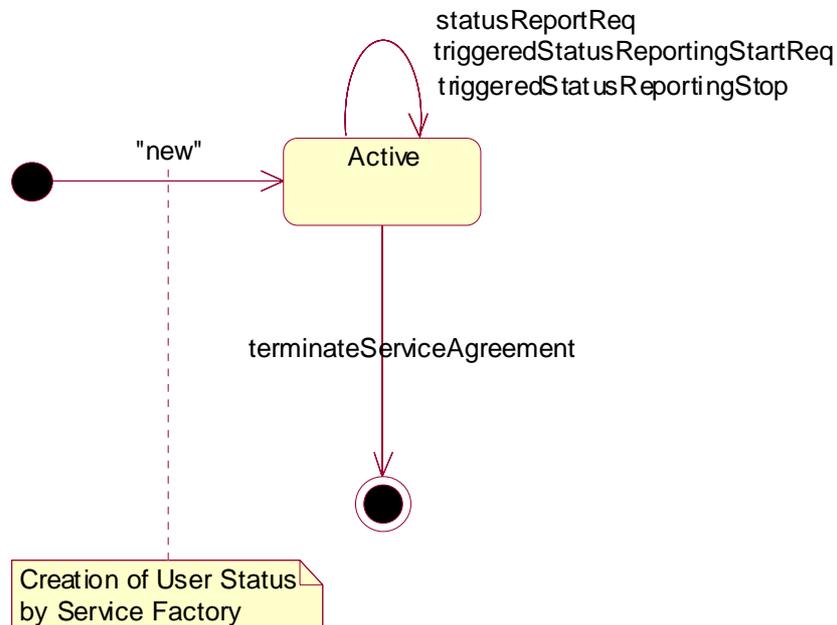


Figure 6: State Transition Diagram for User Status

9.4.1.1 Active State

In this state, a relation between the Application and the User Status Service Capability Feature has been established. It allows the application to request a specific user status report or subscribe to triggers that generate status reports when the status of one of the monitored user changes.

10 Service Properties

10.1 Mobility Properties

10.1.1 Emergency Application Subtypes

NOTE: See definition of "LCS Client Type" in TS 123 271.

This property contains a list of application subtypes that are permitted to use the service. The possible subtypes are (see definition of "LCS Client Internal ID" in TS 123 271):

- "Broadcast service"
- "O&M HPLMN service"
- "O&M VPLMN service"
- "Anonymous location"
- "Target MS subscribed service"

10.1.2 Value Added Application Subtypes

Note: See definition of "LCS Client Type" in TS 123 271.

This property contains a list of application subtypes that are permitted to use the service. The possible subtypes are (see definition of "LCS Client Internal ID" in TS 123 271):

- "Broadcast service"
- "O&M HPLMN service"
- "O&M VPLMN service"
- "Anonymous location"
- "Target MS subscribed service"

10.1.3 PLMN Operator Application Subtypes

NOTE: See definition of "LCS Client Type" in TS 123 271.

This property contains a list of application subtypes that are permitted to use the service. The possible subtypes are (see definition of "LCS Client Internal ID" in TS 123 271):

- "Broadcast service"
- "O&M HPLMN service"
- "O&M VPLMN service"
- "Anonymous location"
- "Target MS subscribed service"

10.1.4 Lawful Intercept Application Subtypes

NOTE: See definition of "LCS Client Type" in TS 123 271.

This property contains a list of application subtypes that are permitted to use the service. The possible subtypes are (see definition of "LCS Client Internal ID" in TS 123 271):

- "Broadcast service"
- "O&M HPLMN service"
- "O&M VPLMN service"
- "Anonymous location"
- "Target MS subscribed service"

10.1.5 Altitude Obtainable

Indicates whether it is possible to obtain a user's altitude.

10.1.6 Location Methods

List of supported location methods. Possible values (other values are permitted):

- "Time of Arrival"
- "Timing Advance"
- "GPS"
- "User Data Lookup"
- "Any Time Interrogation"

10.1.7 Priorities

List of supported priorities for location requests. Possible values (no other values are permitted):

- "Normal"
- "High"

10.1.8 Max Interactive Requests

The maximum number of parallel outstanding location or status requests allowed per application. It shall be possible to convert the value to a 32-bit integer.

10.1.9 Max Triggered Users

The maximum number of users allowed per application for which triggered location reporting can be requested. It shall be possible to convert the value to a 32-bit integer.

10.1.10 Max Periodic Users

The maximum number of users allowed per application for which periodic location reporting can be requested. It shall be possible to convert the value to a 32-bit integer.

10.1.11 Min Periodic Interval Duration

The minimal time in seconds allowed between two periodic reports. It shall be possible to convert the value to a 32-bit integer.

10.2 User Location Service Properties

A specific User Location service shall set the following properties:

- General Properties applicable to all SCFs (in Framework)
- [Permitted application types](#)
- [Permitted application subtypes](#)
- [Priorities](#)

NOTE: See definition of "LCS Client Type" in TS 123 271.

- [Altitude obtainable](#)
- [Location methods](#)
- [Max interactive requests](#)
- [Max triggered users](#)
- [Max periodic users](#)
- [Min periodic interval duration](#)

The example below describes the capabilities of two fictive User Location services:

Property Name	Property Value Service 1	Property Value Service 2
Service instance ID	0x80923AD0	0xF0ED85CB
Service name	UserLocation	UserLocation
Service version	2.1	2.1
Service description	Basic user location service.	Advanced high-performance user location service.
Product name	Find It	Locate.com
Product version	1.3	3.1
Supported interfaces	"IpUserLocation"	"IpUserLocation"
Permitted application types	"Emergency service", "Value added service"	"Emergency service", "Value added service", "Lawful intercept service"
Permitted application subtypes	?	?
Priorities	"Normal"	"Normal", "High"
Altitude obtainable	False	True
Location methods	"Timing Advance"	"GPS", "Time Of Arrival"
Max interactive requests	2 000	10 000
Max triggered users	0	2 000
Max periodic users	300	2 000
Min periodic interval duration	600	30

10.3 User Location Camel Service Properties

A specific User Location Camel service shall set the following properties:

- General Properties applicable to all SCFs (in Framework)
- [Max interactive requests](#)
- [Max triggered users](#)
- [Max periodic users](#)
- [Min periodic interval duration](#)

10.4 User Location Emergency Service Properties

A specific User Location Emergency service shall set the following properties:

- General Properties applicable to all SCFs (in Framework)
- [Permitted application types](#)
- [Permitted application subtypes](#)
- [Altitude obtainable](#)
- [Location methods](#)
- [Priorities](#)

NOTE: See definition of "LCS Client Type" in TS 123 271.

- [Max interactive requests](#)

10.5 User Status Service Properties

A specific User Location service shall set the following properties:

- General Properties applicable to all SCFs (in Framework)
- [Max interactive requests](#)
- [Max triggered users](#)

11 Data Definitions

All data types referenced but not defined in this clause are common data definitions which may be found in ES 201 915-2.

11.1 Common Mobility Data Definitions

The following data definitions are used for several of the mobility services.

11.1.1 TpGeographicalPosition

Defines the [Sequence of Data Elements](#) that specify a geographical position.

The horizontal location is defined by an "ellipsoid point with uncertainty shape". The reference system chosen for the coding of locations is the World Geodetic System 1984 (WGS 84).

TypeOfUncertaintyShape describes the type of the uncertainty shape and *Longitude/Latitude* defines the position of the uncertainty shape. The following table defines the meaning of the data elements that describe the uncertainty shape for each uncertainty shape type.

<i>Type of uncertainty shape</i>	<i>Uncertainty Outer Semi Major</i>	<i>Uncertainty Outer Semi Minor</i>	<i>Uncertainty Inner Semi Major</i>	<i>Uncertainty Inner Semi Minor</i>	<i>Angle Of Semi Major</i>	<i>Segment Start Angle</i>	<i>Segment End Angle</i>
None	-	-	-	-	-	-	-
Circle	radius of circle	-	-	-	-	-	-
Circle Sector	radius of circle	-	-	-	-	start angle of circle segment	end angle of circle segment
Circle Arc Stripe	radius of outer circle	-	radius of inner circle	-	-	start angle of circle arc stripe	end angle of circle arc stripe
Ellipse	length of semi-major axis	length of semi-minor axis	-	-	rotation of ellipse measured clockwise from north	-	-
Ellipse Sector	length of semi-major axis	length of semi-minor axis	-	-	rotation of ellipse measured clockwise from north	start angle of ellipse segment	end angle of ellipse segment
Ellipse Arc Stripe	length of semi-major axis, outer ellipse	length of semi-minor axis, outer ellipse	length of semi-major axis, inner ellipse	length of semi-minor axis, inner ellipse	rotation of ellipse measured clockwise from north	start angle of ellipse arc stripe	end angle of ellipse arc stripe

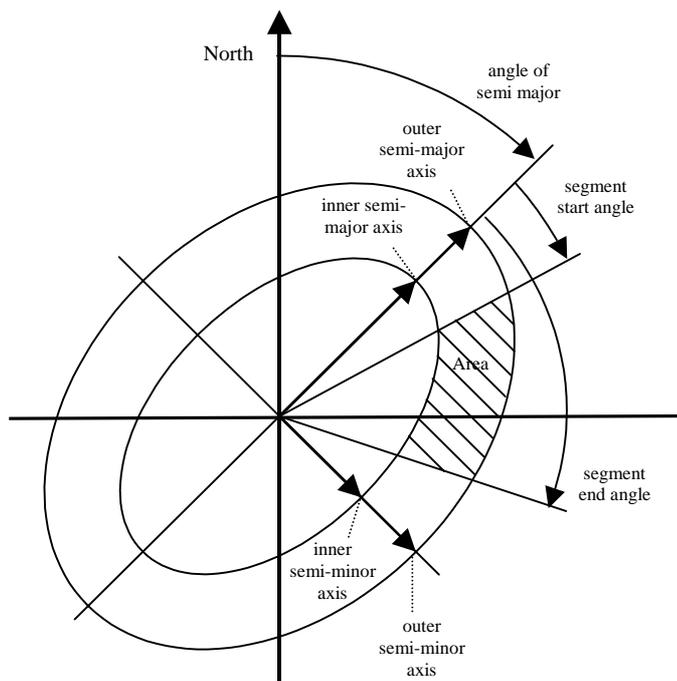


Figure 7: Description of an Ellipse Arc

TpGeographicalPosition:

Sequence Element Name	Sequence Element Type
Longitude	TpFloat
Latitude	TpFloat
TypeOfUncertaintyShape	TpLocationUncertaintyShape
UncertaintyInnerSemiMajor	TpFloat
UncertaintyOuterSemiMajor	TpFloat
UncertaintyInnerSemiMinor	TpFloat
UncertaintyOuterSemiMinor	TpFloat
AngleOfSemiMajor	TpInt32
SegmentStartAngle	TpInt32
SegmentEndAngle	TpInt32

11.1.2 TpLocationPriority

Defines the priority of a location request.

Name	Value	Description
P_M_NORMAL	0	Normal
P_M_HIGH	1	High

11.1.3 TpLocationRequest

Defines the [Sequence of Data Elements](#) that specify a location request.

Sequence Element Name	Sequence Element Type	Description
RequestedAccuracy	TpFloat	Requested accuracy in meters.
RequestedResponseTime	TpLocationResponseTime	Requested response time as a classified requirement or as an absolute timer.
AltitudeRequested	TpBoolean	Altitude request flag.
Type	TpLocationType	The kind of location that is requested.
Priority	TpLocationPriority	Priority of location request.
RequestedLocationMethod	TpString	The kind of location method that is requested.

11.1.4 TpLocationResponseIndicator

Defines a response time requirement.

Name	Value	Description
P_M_NO_DELAY	0	No delay: return either initial or last known location of the user.
P_M_LOW_DELAY	1	Low delay: return the current location with minimum delay. The mobility service shall attempt to fulfil any accuracy requirement, but in doing so shall not add any additional delay.
P_M_DELAY_TOLERANT	2	Delay tolerant: obtain the current location with regard to fulfilling the accuracy requirement.
P_M_USE_TIMER_VALUE	3	Use timer value: obtain the current location with regard to fulfilling the response time requirement.

11.1.5 TpLocationResponseTime

Defines the [Sequence of Data Elements](#) that specify the application's requirements on the mobility service's response time.

Sequence Element Name	Sequence Element Type	Description
ResponseTime	TpLocationResponseIndicator	Indicator for which kind of response time that is required. see TpLocationResponseIndicator.
TimerValue	TpInt32	Optional timer used in combination when ResponseTime equals P_M_USE_TIMER_VALUE.

11.1.6 TpLocationType

Defines the type of location requested.

Name	Value	Description
P_M_CURRENT	0	Current location
P_M_CURRENT_OR_LAST_KNOWN	1	Current or last known location
P_M_INITIAL	2	Initial location for an emergency services call

11.1.7 TpLocationUncertaintyShape

Defines the type of uncertainty shape.

Name	Value	Description
P_M_SHAPE_NONE	0	No uncertainty shape present.
P_M_SHAPE_CIRCLE	1	Uncertainty shape is a circle.
P_M_SHAPE_CIRCLE_SECTOR	2	Uncertainty shape is a circle sector.
P_M_SHAPE_CIRCLE_ARC_STRIPE	3	Uncertainty shape is a circle arc stripe.
P_M_SHAPE_ELLIPSE	4	Uncertainty shape is an ellipse.
P_M_SHAPE_ELLIPSE_SECTOR	5	Uncertainty shape is an ellipse sector.
P_M_SHAPE_ELLIPSE_ARC_STRIPE	6	Uncertainty shape is an ellipse arc stripe.

11.1.8 TpMobilityDiagnostic

Defines a diagnostic value that is reported in addition to an error by one of the mobility services.

Name	Value	Description
P_M_NO_INFORMATION	0	No diagnostic information present. Valid for all type of errors.
P_M_APPL_NOT_IN_PRIV_EXCEPT_LST	1	Application not in privacy exception list. Valid for "Unauthorised Application" error.
P_M_CALL_TO_USER_NOT_SETUP	2	Call to user not set-up. Valid for "Unauthorised Application" error.
P_M_PRIVACY_OVERRIDE_NOT_APPLIC	3	Privacy override not applicable. Valid for "Unauthorised Application" error.
P_M_DISALL_BY_LOCAL_REGULAT_REQ	4	Disallowed by local regulatory requirements. Valid for "Unauthorised Application" error.
P_M_CONGESTION	5	Congestion. Valid for "Position Method Failure" error.
P_M_INSUFFICIENT_RESOURCES	6	Insufficient resources. Valid for "Position Method Failure" error.
P_M_INSUFFICIENT_MEAS_DATA	7	Insufficient measurement data. Valid for "Position Method Failure" error.
P_M_INCONSISTENT_MEAS_DATA	8	Inconsistent measurement data. Valid for "Position Method Failure" error.
P_M_LOC_PROC_NOT_COMPLETED	9	Location procedure not completed. Valid for "Position Method Failure" error.
P_M_LOC_PROC_NOT_SUPP_BY_USER	10	Location procedure not supported by user. Valid for "Position Method Failure" error.
P_M_QOS_NOT_ATTAINABLE	11	Quality of service not attainable. Valid for "Position Method Failure" error.

11.1.9 TpMobilityError

Defines an error that is reported by one of the mobility services.

Name	Value	Description	Fatal
P_M_OK	0	No error occurred while processing the request.	N/A
P_M_SYSTEM_FAILURE	1	System failure. The request cannot be handled because of a general problem in the mobility service or the underlying network.	Yes
P_M_UNAUTHORIZED_NETWORK	2	Unauthorised network, The requesting network is not authorised to obtain the user's location or status.	No
P_M_UNAUTHORIZED_APPLICATION	3	Unauthorised application. The application is not authorised to obtain the user's location or status.	Yes
P_M_UNKNOWN_SUBSCRIBER	4	Unknown subscriber. The user is unknown, i.e. no such subscription exists.	Yes
P_M_ABSENT_SUBSCRIBER	5	Absent subscriber. The user is currently not reachable.	No
P_M_POSITION_METHOD_FAILURE	6	Position method failure. The mobility service failed to obtain the user's position.	No

11.1.10 TpMobilityStopAssignmentData

Defines the [Sequence of Data Elements](#) that specify a request to stop whole or parts of an assignment. Assignments are used for periodic or triggered reporting of a user's location or status.

Note that the parameter "Users" is optional. If the parameter "StopScope" is set to P_M_ALL_IN_ASSIGNMENT the parameter "Users" is undefined. If the parameter "StopScope" is set to P_M_SPECIFIED_USERS, then the assignment shall be stopped only for those users specified in the "Users" list.

Sequence Element Name	Sequence Element Type	Description
AssignmentId	TpSessionID	Identity of the session that shall be stopped.
StopScope	TpMobilityStopScope	Specify if only a part of the assignment or if all the assignment shall be stopped.
Users	TpAddressSet	Optional parameter describing which users a stop request is addressing, when only a part of an assignment is to be stopped.

11.1.11 TpMobilityStopScope

This enumeration is used in requests to stop mobility reports that are sent from a mobility service to an application.

Name	Value	Description
P_M_ALL_IN_ASSIGNMENT	0	The request concerns all users in an assignment.
P_M_SPECIFIED_USERS	1	The request concerns only the users that are explicitly specified in a list.

11.1.12 TpTerminalType

Defines which kind of terminal is used.

Name	Value	Description
P_M_FIXED	0	Fixed terminal.
P_M_MOBILE	1	Mobile terminal.
P_M_IP	2	IP terminal.

11.2 User Location Data Definitions

11.2.1 IpUserLocation

Defines the address of an IpUserLocation Interface.

11.2.2 IpUserLocationRef

Defines a [Reference](#) to type IpUserLocation.

11.2.3 IpAppUserLocation

Defines the address of an IpAppUserLocation Interface.

11.2.4 IpAppUserLocationRef

Defines a [Reference](#) to type IpAppUserLocation.

11.2.5 IpTriggeredUserLocation

Defines the address of an IpTriggeredUserLocation Interface.

11.2.6 IpTriggeredUserLocationRef

Defines a [Reference](#) to type IpTriggeredUserLocation.

11.2.7 IpAppTriggeredUserLocation

Defines the address of an IpAppTriggeredUserLocation Interface.

11.2.8 IpAppTriggeredUserLocationRef

Defines a [Reference](#) to type IpAppTriggeredUserLocation.

11.2.9 TpUIExtendedData

Defines the [Sequence of Data Elements](#) that specify a location (extended format).

The optional vertical location is defined by the data element *Altitude*, which contains the altitude in meters above sea level, and the data element *AltitudeAccuracy*, which contains the accuracy of the altitude.

Sequence Element Name	Sequence Element Type	Description
GeographicalPosition	TpGeographicalPosition	Specification of a position and an area of uncertainty.
TerminalType	TpTerminalType	Kind of terminal.
AltitudePresent	TpBoolean	Flag indicating if the altitude is present.
Altitude	TpFloat	Decimal altitude in meters.
UncertaintyAltitude	TpFloat	Uncertainty of the altitude.
TimestampPresent	TpBoolean	Flag indicating if the timestamp is present.
Timestamp	TpDateAndTime	Timestamp indicating when the position was measured.
UsedLocationMethod	TpString	Specifying which location method was used.

11.2.10 TpUIExtendedDataSet

Defines a [Numbered Set of Data Elements](#) of TpUIExtendedData.

11.2.11 TpUserLocationExtended

Defines the [Sequence of Data Elements](#) that specify the identity and location(s) of a user (extended format). In general the data element *Locations* will contain only one location, but in case of IP-telephony users this data element might continue several locations (the locations of all communication end-points, where the user is currently registered).

Sequence Element Name	Sequence Element Type	Description
UserID	TpAddress	The address of the user.
StatusCode	TpMobilityError	Indicator of error.
Locations	TpUlExtendedDataSet	Optional list of locations. If StatusCode is indicating an error, this value is undefined.

11.2.12 TpUserLocationExtendedSet

Defines a [Numbered Set of Data Elements](#) of [TpUserLocationExtended](#).

11.2.13 TpLocationTrigger

Defines the [Sequence of Data Elements](#) that specify the criteria for a triggered location report to be generated. The area is defined by an ellipse.

Sequence Element Name	Sequence Element Type	Description
Longitude	TpFloat	Longitude of the position used in the trigger.
Latitude	TpFloat	Latitude of the position used in the trigger.
AreaSemiMajor	TpFloat	Semi major of ellipse area used in the trigger.
AreaSemiMinor	TpFloat	Semi minor of ellipse area used in the trigger.
AngleOfSemiMajor	TpInt32	Angle of the semi major of the ellipse area used in the trigger.
Criterion	TpLocationTriggerCriteria	Trigger criteria with regard to the ellipse area.
ReportingInterval	TpDuration	Duration between generated location reports.

11.2.14 TpLocationTriggerSet

Defines a [Numbered Set of Data Elements](#) of [TpLocationTrigger](#).

11.2.15 TpLocationTriggerCriteria

Defines the criteria that trigger a location report.

Name	Value	Description
P_UL_ENTERING_AREA	0	User enters the area
P_UL_LEAVING_AREA	1	User leaves the area

11.2.16 TpUserLocation

Defines the [Sequence of Data Elements](#) that specify the identity and location of a user (basic format).

Sequence Element Name	Sequence Element Type	Description
UserID	TpAddress	The address of the user.
StatusCode	TpMobilityError	Indicator of error.
GeographicalPosition	TpGeographicalPosition	Specification of a position and an area of uncertainty. If StatusCode is indicating an error, this value is undefined.

11.2.17 TpUserLocationSet

Defines a [Numbered Set of Data Elements](#) of [TpUserLocation](#).

11.3 User Location Camel Data Definitions

11.3.1 IpUserLocationCamel

Defines the address of an IpUserLocationCamel Interface.

11.3.2 IpUserLocationCamelRef

Defines a [Reference](#) to type IpUserLocationCamel.

11.3.3 IpAppUserLocationCamel

Defines the address of an IpAppUserLocationCamel Interface.

11.3.4 IpAppUserLocationCamelRef

Defines a [Reference](#) to type IpAppUserLocationCamel.

11.3.5 TpLocationCellIDOrLAI

This data type is identical to a [TpString](#). It specifies the Cell Global Identification or the Location Area Identification (LAI).

The Cell Global Identification (CGI) is defined as a string of characters in the following format:

MCC-MNC-LAC-CI

where:

MCC Mobile Country Code (three decimal digits)
 MNC Mobile Network Code (two or three decimal digits)
 LAC Location Area Code (four hexadecimal digits)
 CI Cell Identification (four hexadecimal digits)

The Location Area Identification (LAI) is defined as a string of characters in the following format:

MCC-MNC-LAC

where:

MCC Mobile Country Code (three decimal digits)
 MNC Mobile Network Code (two or three decimal digits)
 LAC Location Area Code (four hexadecimal digits)

The length of the parameter indicates, which format is used. See TS 129 002 for the detailed coding.

11.3.6 TpLocationTriggerCamel

Defines the [Sequence of Data Elements](#) that specify the criteria for a triggered location report to be generated.

Sequence Element Name	Sequence Element Type	Description
UpdateInsideVlr	TpBoolean	Generate location report, when a location update occurs inside the current VLR area.
UpdateOutsideVlr	TpBoolean	Generate location report, when the user moves to another VLR area.

11.3.7 TpUserLocationCamel

Defines the [Sequence of Data Elements](#) that specify the location of a mobile telephony user. Note that if the `StatusCode` is indicating an error, then neither `GeographicalPosition`, `Timestamp`, `VlrNumber`, `LocationNumber`, `CellIdOrLai` nor their associated presence flags are defined.

Sequence Element Name	Sequence Element Type	Description
UserID	TpAddress	The address of the user.
StatusCode	TpMobilityError	Indicator of error.
GeographicalPositionPresent	TpBoolean	Flag indicating if the geographical position is present.
GeographicalPosition	TpGeographicalPosition	Specification of a position and an area of uncertainty.
TimestampPresent	TpBoolean	Flag indicating if the timestamp is present.
Timestamp	TpDateAndTime	Timestamp indicating when the request was processed.
VlrNumberPresent	TpBoolean	Flag indicating if the VLR number is present.
VlrNumber	TpAddress	Current VLR number for the user.
LocationNumberPresent	TpBoolean	Flag indicating if the location number is present.
LocationNumber (see Note)	TpAddress	Current location number.
CellIdOrLaiPresent	TpBoolean	Flag indicating if cell-id or LAI of the user is present.
CellIdOrLai	TpLocationCellIDOrLAI	Cell-id or LAI of the user.

NOTE: The location number is the number to the MSC or in rare cases the roaming number.

11.3.8 TpUserLocationCamelSet

Defines a [Numbered Set of Data Elements](#) of `TpUserLocationCamel`.

11.4 User Location Emergency Data Definitions

11.4.1 IpUserLocationEmergency

Defines the address of an `IpUserLocationEmergency` Interface.

11.4.2 IpUserLocationEmergencyRef

Defines a [Reference](#) to type `IpUserLocationEmergency`.

11.4.3 IpAppUserLocationEmergency

Defines the address of an `IpAppUserLocationEmergency` Interface.

11.4.4 IpAppUserLocationEmergencyRef

Defines a [Reference](#) to type `IpAppUserLocationEmergency`.

11.4.5 TpIMEI

This data type is identical to a [TpString](#). It specifies the International Mobile Equipment Identity (IMEI).

11.4.6 TpNaESRD

This data type is identical to a [TpString](#). It specifies the North American Emergency Services Routing Digits (NA-ESRD).

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 its associated Location Services client. The NA-ESRD also identifies the base station, cell site or sector from which a North American emergency call originates.

11.4.7 TpNaESRK

This data type is identical to a [TpString](#). It specifies the North American Emergency Services Routing Key (NA-ESRK).

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.

11.4.8 TpUserLocationEmergencyRequest

Defines the [Sequence of Data Elements](#) that specify the request for the location of an emergency service user. The emergency service user is identified by a combination of *user address*, *NaESRD*, *NaESRK* and *IMEI*. *NaESRD*, *NaESRK* and *IMEI* may be provided, if the emergency service user has originated the emergency service call in North America.

Sequence Element Name	Sequence Element Type	Description
UserAddressPresent	TpBoolean	Flag indicating if the user address is present.
UserAddress	TpAddress	The address of the user.
NaEsrdrPresent	TpBoolean	Flag indicating if the NaESRD is present.
NaEsrdr	TpNaESRD	Current NaESRD for the user.
NaEsrkrPresent	TpBoolean	Flag indicating if the NaESRK is present.
NaEsrkr	TpNaESRK	Current NaESRK for the user.
ImeiPresent	TpBoolean	Flag indicating if the IMEI is present.
Imei	TpIMEI	IMEI for the user.
LocationReq	TpLocationRequest	The actual location request.

11.4.9 TpUserLocationEmergency

Defines the [Sequence of Data Elements](#) that specify the identity and location of an emergency service user. The emergency service user is identified by a combination of *UserID*, *NaESRD*, *NaESRK* and *IMEI*.

NaESRD, *NaESRK* and *IMEI* may be provided, if the emergency service user has originated the emergency service call in North America.

The horizontal location is defined by an "ellipsoid point with uncertainty ellipse" (see [TpUIExtendedData](#)).

Sequence Element Name	Sequence Element Type	Description
StatusCode	TpMobilityError	Indicator of error.
UserIdPresent	TpBoolean	Flag indicating if the user address is present.
UserId	TpAddress	The user address.
NaEsrdrPresent	TpBoolean	Flag indicating if the NaESRD is present.
NaEsrdr	TpNaESRD	Current NaESRD for the user.
NaEsrkrPresent	TpBoolean	Flag indicating if the NaESRK is present.
NaEsrkr	TpNaESRK	Current NaESRK for the user.
ImeiPresent	TpBoolean	Flag indicating if the IMEI is present.
Imei	TpIMEI	IMEI for the user.
TriggeringEvent	TpUserLocationEmergencyTrigger	The reason for this location report.
GeographicalPositionPresent	TpBoolean	Flag indicating if the geographical position is present.
GeographicalPosition	TpGeographicalPosition	Specification of a position and an area of uncertainty.
AltitudePresent	TpBoolean	Flag indicating if the altitude is present.
Altitude	TpFloat	Decimal altitude in meters.
UncertaintyAltitude	TpFloat	Uncertainty of the altitude.
TimestampPresent	TpBoolean	Flag indicating if a timestamp is present.
Timestamp	TpDateAndTime	Timestamp indicating when the request was processed.
UsedLocationMethod	TpString	Specifying which location method was used.

11.4.10 TpUserLocationEmergencyTrigger

Defines which event triggered the emergency User Location report.

Name	Value	Description
P_ULE_CALL_ORIGINATION	0	An emergency service user originated an emergency call.
P_ULE_CALL_RELEASE	1	An emergency service user released an emergency call.
P_ULE_LOCATION_REQUEST	2	The report is a response to an emergency location report request.

11.5 User Status Data Definitions

11.5.1 IpUserStatus

Defines the address of an IpUserStatus Interface.

11.5.2 IpUserStatusRef

Defines a [Reference](#) to type IpUserStatus.

11.5.3 IpAppUserStatus

Defines the address of an IpAppUserStatus Interface.

11.5.4 IpAppUserStatusRef

Defines a [Reference](#) to type IpAppUserStatus.

11.5.5 TpUserStatus

Defines the [Sequence of Data Elements](#) that specify the identity and status of a user.

Sequence Element Name	Sequence Element Type	Description
UserID	TpAddress	The user address.
StatusCode	TpMobilityError	Indicator of error.
Status	TpUserStatusIndicator	The current status of the user.
TerminalType	TpTerminalType	The kind of terminal used by the user.

11.5.6 TpUserStatusSet

Defines a [Numbered Set of Data Elements](#) of TpUserStatus.

11.5.7 TpUserStatusIndicator

Defines the status of a user.

Name	Value	Description
P_US_REACHABLE	0	User is reachable
P_US_NOT_REACHABLE	1	User is not reachable
P_US_BUSY (see Note)	2	User is busy (only applicable for interactive user status request, not when triggers are used)
NOTE: Only applicable to mobile (Camel) telephony users.		

11.6 Units and Validations of Parameters

This clause describes the units that shall be used for data elements, where this is not obvious.

Altitude

Unit: Metric meter

Angle

Unit: Degrees

Value constraint: $0 \leq \text{'Angle'} \leq 360$

AreaSemiMajor and AreaSemiMinor

Unit: Metric meter

Value constraint: $0 \leq \text{'AreaSemi...'} \leq 360$

ReportingInterval

Unit: Seconds

Value constraint: $0 < \text{'ReportingInterval'}$

UncertaintyAltitude

Unit: Metric meter

Value constraint: $0 \leq \text{'UncertaintyAltitude'}$

Semantic: $(\text{Altitude} - \text{UncertaintyAltitude}) \leq \text{'Terminal actual altitude'} \leq (\text{'Altitude'} + \text{'UncertaintyAltitude'})$

UncertaintyInnerSemiMajor and UncertaintyInnerSemiMinor

Unit: Metric meter

Value constraint: $0 \leq \text{'UncertaintyInner...'}$

UncertaintyOuterSemiMajor and UncertaintyOuterSemiMinor

Unit: Metric meter

Value constraint: $0 \leq \text{'UncertaintyInner...'}$

UsedLocationMethod

Predefined strings are listed in clause 10.1.6 Location Methods.

12 Exception Classes

The following are the list of exception classes which are used in this interface of the API.

Name	Description
P_INVALID_REPORTING_INTERVAL	The requested reporting interval is not valid
P_REQUESTED_ACCURACY_CANNOT_BE_DELIVERED	The requested location accuracy cannot be delivered
P_REQUESTED_RESPONSE_TIME_CANNOT_BE_DELIVERED	The requested response time cannot be delivered
P_TRIGGER_CONDITIONS_NOT_SUBSCRIBED	Trigger conditions not subscribed

Each exception class contains the following structure:

Structure Element Name	Structure Element Type	Structure Element Description
ExtraInformation	TpString	Carries extra information to help identify the source of the exception, e.g. a parameter name

Annex A (normative): OMG IDL Description of Mobility SCF

The OMG IDL representation of this interface specification is contained in a text file (mm.idl contained in archive es_20191506v010201p0.ZIP) which accompanies the present document.

Annex B (informative): Contents of 3GPP OSA R4 Mobility

All parts of User Location, User Location Camel and User Status are relevant for TS 129 198-6 V4 (Release 4).

Annex C (informative): Summary of differences between V1.1.1 (Parlay 3.0) and V1.2.1 (Parlay 3.1)

C.1 IpUserLocation

Exception P_INVALID_INTERFACE_TYPE added to the following methods:

locationReportReq(), extendedLocationReportReq(), periodicLocationReportingStartReq()

C.2 IpTriggeredUserLocation

triggeredLocationReportingStartReq (appLocation: in ~~IpAppTriggeredUserLocationRef~~IpAppUserLocationRef, users:
in TpAddressSet, request: in TpLocationRequest, triggers: in TpLocationTriggerSet): TpSessionID

Exception P_INVALID_INTERFACE_TYPE added to the following methods:

triggeredLocationReportingStartReq()

C.3 IpUserLocationCamel

Exception P_INVALID_INTERFACE_TYPE added to the following methods:

locationReportReq(), extendedLocationReportReq(), periodicLocationReportingStartReq(),
triggeredLocationReportingStartReq()

C.4 IpUserLocationEmergency

Exception P_INVALID_INTERFACE_TYPE added to the following methods:

emergencyLocationReportReq(), subscribeEmergencyLocationReports()

C.5 IpUserStatus

Exception P_INVALID_INTERFACE_TYPE added to the following methods:

statusReportReq(), triggeredStatusReportingStartReq()

C.6 IpService

setCallback() and setCallbackWithSessionID() now both raise P_INVALID_INTERFACE_TYPE.

History

Document history		
V1.1.1	February 2002	Publication
V1.2.1	May 2002	Membership Approval Procedure MV 20020705: 2002-05-07 to 2002-07-05
V1.2.1	July 2002	Publication