

**Open Service Access (OSA);  
Application Programming Interface (API);  
Part 6: Mobility SCF  
(Parlay 5)**

---



---

Reference

DES/TISPAN-01005-06-OSA

---

Keywords

API, IDL, OSA, UML

**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, please send your comment to one of the following services:

[http://portal.etsi.org/chaicor/ETSI\\_support.asp](http://portal.etsi.org/chaicor/ETSI_support.asp)

---

**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 2005.

© The Parlay Group 2005.

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 .....	8
Foreword.....	8
1 Scope .....	9
2 References .....	9
3 Definitions and abbreviations.....	9
3.1 Definitions .....	9
3.2 Abbreviations .....	9
4 Mobility SCF.....	10
4.1 General requirements on support of methods.....	10
5 Sequence Diagrams .....	10
5.1 User Location Sequence Diagrams.....	10
5.1.1 User Location Interrogation - Triggered Request .....	10
5.1.2 User Location Interrogation - Periodic Request.....	12
5.1.3 User Location Interrogation - Interactive Request.....	13
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 .....	15
5.2.3 User Location Camel Interrogation - Interactive Request .....	16
5.3 User Location Emergency Sequence Diagrams .....	16
5.3.1 Subscription and Network Induced Location Reports .....	16
5.3.2 Network Induced Location Reports .....	17
5.3.3 Interactive Location Request .....	18
5.4 User Status Sequence Diagrams .....	18
5.4.1 Triggered Reporting.....	18
5.4.2 Interactive Request .....	19
5.5 User Binding Sequence Diagrams.....	20
5.5.1 Subscription and Binding Request Notifications .....	20
6 Class Diagrams.....	20
6.1 User Location Class Diagrams .....	20
6.2 User Location Camel Class Diagrams.....	22
6.3 User Location Emergency Class Diagrams .....	22
6.4 User Status Class Diagrams.....	23
6.5 User Binding Class Diagrams .....	23
7 The Service Interface Specifications .....	24
7.1 Interface Specification Format .....	24
7.1.1 Interface Class .....	24
7.1.2 Method descriptions.....	24
7.1.3 Parameter descriptions.....	24
7.1.4 State Model.....	24
7.2 Base Interface.....	24
7.2.1 Interface Class IpInterface .....	24
7.3 Service Interfaces .....	25
7.3.1 Overview .....	25
7.4 Generic Service Interface .....	25
7.4.1 Interface Class IpService .....	25
7.4.1.1 Method setCallback() .....	25
7.4.1.2 Method setCallbackWithSessionID().....	25
8 Mobility Interface Classes.....	26
8.1 User Location Interface Classes .....	26
8.1.1 Interface Class IpUserLocation.....	26
8.1.1.1 Method locationReportReq().....	27

8.1.1.2	Method extendedLocationReportReq() .....	27
8.1.1.3	Method periodicLocationReportingStartReq() .....	28
8.1.1.4	Method periodicLocationReportingStop() .....	28
8.1.1.5	Method <<new>> getNextPeriodicLocationRequest() .....	29
8.1.2	Interface Class IpAppUserLocation .....	29
8.1.2.1	Method locationReportRes() .....	30
8.1.2.2	Method locationReportErr() .....	30
8.1.2.3	Method extendedLocationReportRes() .....	31
8.1.2.4	Method extendedLocationReportErr() .....	31
8.1.2.5	Method periodicLocationReport() .....	31
8.1.2.6	Method periodicLocationReportErr() .....	31
8.1.3	Interface Class IpTriggeredUserLocation .....	32
8.1.3.1	Method triggeredLocationReportingStartReq() .....	32
8.1.3.2	Method triggeredLocationReportingStop() .....	33
8.1.3.3	Method <<new>> getNextTriggeredLocationRequest() .....	33
8.1.4	Interface Class IpAppTriggeredUserLocation .....	34
8.1.4.1	Method triggeredLocationReport() .....	34
8.1.4.2	Method triggeredLocationReportErr() .....	34
8.2	User Location Camel Interface Classes .....	35
8.2.1	Interface Class IpUserLocationCamel .....	35
8.2.1.1	Method locationReportReq() .....	35
8.2.1.2	Method periodicLocationReportingStartReq() .....	36
8.2.1.3	Method periodicLocationReportingStop() .....	36
8.2.1.4	Method triggeredLocationReportingStartReq() .....	37
8.2.1.5	Method triggeredLocationReportingStop() .....	37
8.2.1.6	Method <<new>> getNextPeriodicLocationRequest() .....	37
8.2.1.7	Method <<new>> getNextTriggeredLocationRequest() .....	38
8.2.2	Interface Class IpAppUserLocationCamel .....	38
8.2.2.1	Method locationReportRes() .....	39
8.2.2.2	Method locationReportErr() .....	39
8.2.2.3	Method periodicLocationReport() .....	40
8.2.2.4	Method periodicLocationReportErr() .....	40
8.2.2.5	Method triggeredLocationReport() .....	40
8.2.2.6	Method triggeredLocationReportErr() .....	40
8.3	User Location Emergency Interface Classes .....	41
8.3.1	Interface Class IpUserLocationEmergency .....	41
8.3.1.1	Method emergencyLocationReportReq() .....	41
8.3.1.2	Method subscribeEmergencyLocationReports() .....	42
8.3.1.3	Method unsubscribeEmergencyLocationReports() .....	42
8.3.2	Interface Class IpAppUserLocationEmergency .....	42
8.3.2.1	Method emergencyLocationReport() .....	43
8.3.2.2	Method emergencyLocationReportErr() .....	43
8.4	User Status Interface Classes .....	43
8.4.1	Interface Class IpAppUserStatus .....	44
8.4.1.1	Method statusReportRes() .....	44
8.4.1.2	Method statusReportErr() .....	44
8.4.1.3	Method triggeredStatusReport() .....	45
8.4.1.4	Method triggeredStatusReportErr() .....	45
8.4.1.5	Method <<new>> extendedStatusReportRes() .....	45
8.4.1.6	Method <<new>> extendedStatusReportErr() .....	45
8.4.1.7	Method <<new>> extTriggeredStatusReport() .....	46
8.4.1.8	Method <<new>> extTriggeredStatusReportErr() .....	46
8.4.2	Interface Class IpUserStatus .....	46
8.4.2.1	Method statusReportReq() .....	47
8.4.2.2	Method triggeredStatusReportingStartReq() .....	47
8.4.2.3	Method triggeredStatusReportingStop() .....	48
8.4.2.4	Method <<new>> getNextTriggeredStatusRequest() .....	48
8.4.2.5	Method <<new>> extendedStatusReportReq() .....	49
8.4.2.6	Method <<new>> extTriggeredStatusReportingStartReq() .....	49
8.4.2.7	Method <<new>> extTriggeredStatusReportingStop() .....	50
8.5	User Binding Interface Classes .....	50
8.5.1	Interface Class IpUserBinding .....	50

8.5.1.1	Method <<new>> triggeredBindingRequestNotificationStartReq()	50
8.5.1.2	Method <<new>> triggeredBindingRequestNotificationStop()	51
8.5.2	Interface Class IpAppUserBinding	51
8.5.2.1	Method <<new>> triggeredBindingRequestNotification()	51
8.5.2.2	Method <<new>> triggeredBindingRequestNotificationStartErr()	52
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	53
9.3	User Location Emergency	53
9.4	User Status	53
9.4.1	State Transition Diagrams for IpUserStatus	53
9.4.1.1	Active State	53
9.5	User Binding	53
10	Service Properties	54
10.1	Mobility Properties	54
10.1.1	Emergency Application Subtypes	54
10.1.2	Value Added Application Subtypes	54
10.1.3	PLMN Operator Application Subtypes	54
10.1.4	Lawful Intercept Application Subtypes	55
10.1.5	Altitude Obtainable	55
10.1.6	Location Methods	55
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	56
10.2	User Location Service Properties	56
10.3	User Location Camel Service Properties	56
10.4	User Location Emergency Service Properties	57
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	59
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 .....	64
11.2.18	TpTriggeredLocationRequestSetEntry .....	64
11.2.19	TpTriggeredLocationRequestSet .....	64
11.2.20	TpTriggeredLocationRequest .....	64
11.2.21	TpPeriodicLocationRequestSetEntry.....	64
11.2.22	TpPeriodicLocationRequestSet.....	64
11.2.23	TpPeriodicLocationRequest.....	64
11.3	User Location Camel Data Definitions .....	65
11.3.1	IpUserLocationCamel.....	65
11.3.2	IpUserLocationCamelRef .....	65
11.3.3	IpAppUserLocationCamel .....	65
11.3.4	IpAppUserLocationCamelRef .....	65
11.3.5	TpLocationCellIDOrLAI .....	65
11.3.6	TpLocationTriggerCamel .....	65
11.3.7	TpUserLocationCamel.....	66
11.3.8	TpUserLocationCamelSet.....	66
11.4	User Location Emergency Data Definitions.....	66
11.4.1	IpUserLocationEmergency .....	66
11.4.2	IpUserLocationEmergencyRef .....	66
11.4.3	IpAppUserLocationEmergency .....	66
11.4.4	IpAppUserLocationEmergencyRef.....	66
11.4.5	TpIMEI .....	66
11.4.6	TpNaESRD .....	67
11.4.7	TpNaESRK .....	67
11.4.8	TpUserLocationEmergencyRequest .....	67
11.4.9	TpUserLocationEmergency .....	68
11.4.10	TpUserLocationEmergencyTrigger .....	68
11.5	User Status Data Definitions .....	68
11.5.1	IpUserStatus.....	68
11.5.2	IpUserStatusRef .....	68
11.5.3	IpAppUserStatus .....	68
11.5.4	IpAppUserStatusRef .....	68
11.5.5	TpUserStatus.....	69
11.5.6	TpUserStatusSet .....	69
11.5.7	TpUserStatusIndicator .....	69
11.5.8	TpTriggeredStatusRequestSetEntry.....	69
11.5.9	TpTriggeredStatusRequestSet.....	69
11.5.10	TpTriggeredStatusRequest.....	69
11.5.11	TpUserStatusExtended.....	70
11.5.12	TpUserStatusExtendedSet .....	70
11.5.13	TpUserStatusIndicatorExtended .....	70
11.5.14	TpAuthStatusIndicator .....	70
11.5.15	TpUserInfo.....	71
11.5.16	TpNetworkStatusIndicator.....	71
11.5.17	TpAccessTechnology.....	71
11.5.18	TpRoamingStatus .....	71
11.6	User Binding Data Definitions .....	71
11.6.1	TpBindingSet.....	71
11.6.2	TpBindingEntrySet .....	71
11.6.3	TpBindingEntry .....	72
11.6.4	TpBindingEntryType .....	72
11.6.5	TpBindingNotificationCriteriaSet.....	72
11.6.6	TpBindingNotificationCriteria.....	72
11.6.7	TpBindingNotificationCriteriaType.....	72
11.7	Units and Validations of Parameters .....	73
12	Exception Classes.....	74
<b>Annex A (normative):</b>	<b>OMG IDL Description of Mobility SCF .....</b>	<b>75</b>

<b>Annex B (informative):</b>	<b>W3C WSDL Description of Mobility SCF.....</b>	<b>76</b>
<b>Annex C (informative):</b>	<b>Java API Description of the Mobility SCFs .....</b>	<b>77</b>
<b>Annex D (informative):</b>	<b>Contents of 3GPP OSA R6 Mobility .....</b>	<b>78</b>
<b>Annex E (informative):</b>	<b>Description of Mobility SCF for 3GPP2 cdma2000 networks .....</b>	<b>79</b>
E.1	General Exceptions.....	79
E.2	Specific Exceptions .....	79
E.2.1	Clause 1: Scope .....	79
E.2.2	Clause 2: References .....	79
E.2.3	Clause 3: Definitions and abbreviations .....	79
E.2.4	Clause 4: Mobility SCF .....	79
E.2.5	Clause 5: Sequence Diagrams .....	79
E.2.6	Clause 6 Class Diagrams.....	79
E.2.7	Clause 7: The Service Interface Specifications .....	79
E.2.8	Clause 8: Mobility Interface Classes .....	80
E.2.9	Clause 9: State Transition Diagrams .....	80
E.2.10	Clause 10: Service Properties.....	80
E.2.11	Clause 11: Data Definitions.....	80
E.2.12	Clause 12: Exception Classes.....	80
E.2.13	Annex A (normative): OMG IDL Description of Mobility SCF.....	80
E.2.14	Annex B (informative): W3C WSDL Description of Mobility SCF.....	80
E.2.15	Annex C (informative): Java™ API Description of Mobility SCF .....	80
<b>Annex F (informative):</b>	<b>Record of changes .....</b>	<b>81</b>
F.1	Interfaces .....	81
F.1.1	New .....	81
F.1.2	Deprecated.....	81
F.1.3	Removed.....	81
F.2	Methods.....	82
F.2.1	New .....	82
F.2.2	Deprecated.....	82
F.2.3	Modified.....	82
F.2.4	Removed.....	82
F.3	Data Definitions .....	83
F.3.1	New .....	83
F.3.2	Modified.....	83
F.3.3	Removed.....	83
F.4	Service Properties.....	84
F.4.1	New .....	84
F.4.2	Deprecated.....	84
F.4.3	Modified.....	84
F.4.4	Removed.....	84
F.5	Exceptions .....	84
F.5.1	New .....	84
F.5.2	Modified.....	84
F.5.3	Removed.....	85
F.6	Others .....	85
History .....		86

---

# 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 Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN), and is now submitted for the ETSI standards Membership Approval Procedure.

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 203 915) is structured in the following parts:

- Part 1: "Overview";
- Part 2: "Common Data Definitions";
- Part 3: "Framework";
- Part 4: "Call Control";
- 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";
- Part 13: "Policy Management SCF";
- Part 14: "Presence and Availability Management SCF";
- Part 15: "Multi-media Messaging SCF".

The present document has been defined jointly between ETSI, The Parlay Group (<http://www.parlay.org>) and the 3GPP, in co-operation with a number of JAIN™ Community (<http://www.java.sun.com/products/jain>) member companies.

**The present document forms part of the Parlay 5.0 set of specifications.**

**A subset of the present document is in 3GPP TS 29.198-6 V6.3.0 (Release 6).**



---

# 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.
- WSDL Description of the interfaces.
- Reference to the Java™ API 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 203 915-1 contain provisions which, through reference in this text, constitute provisions of the present document.

ETSI ES 203 915-1: "Open Service Access (OSA); Application Programming Interface (API); Part 1: Overview (Parlay 5)".

ETSI ES 203 915-2: "Open Service Access (OSA); Application Programming Interface (API); Part 2: Common Data Definitions (Parlay 5)".

---

## 3 Definitions and abbreviations

### 3.1 Definitions

For the purposes of the present document, the terms and definitions given in ES 203 915-1 apply.

### 3.2 Abbreviations

For the purposes of the present document, the abbreviations defined in ES 203 915-1 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 shows 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 shows 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 ES 203 915-2.

### 4.1 General requirements on support of methods

An implementation of this API which supports or implements a method described in the present document, shall support or implement the functionality described for that method, for at least one valid set of values for the parameters of that method.

Where a method is not supported by an implementation of a Service interface, the exception `P_METHOD_NOT_SUPPORTED` shall be returned to any call of that method.

Where a method is not supported by an implementation of an Application interface, a call to that method shall be possible, and no exception shall be returned.

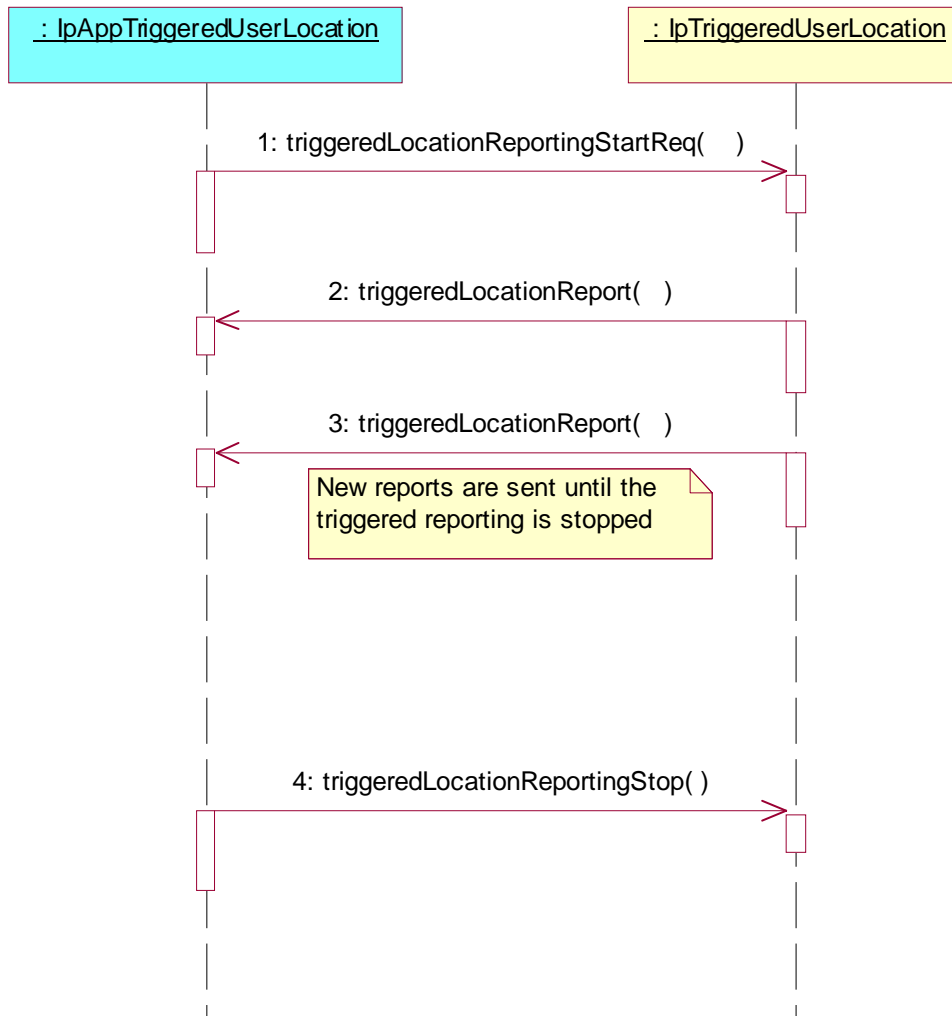
---

## 5 Sequence Diagrams

### 5.1 User Location Sequence Diagrams

#### 5.1.1 User Location Interrogation - Triggered Request

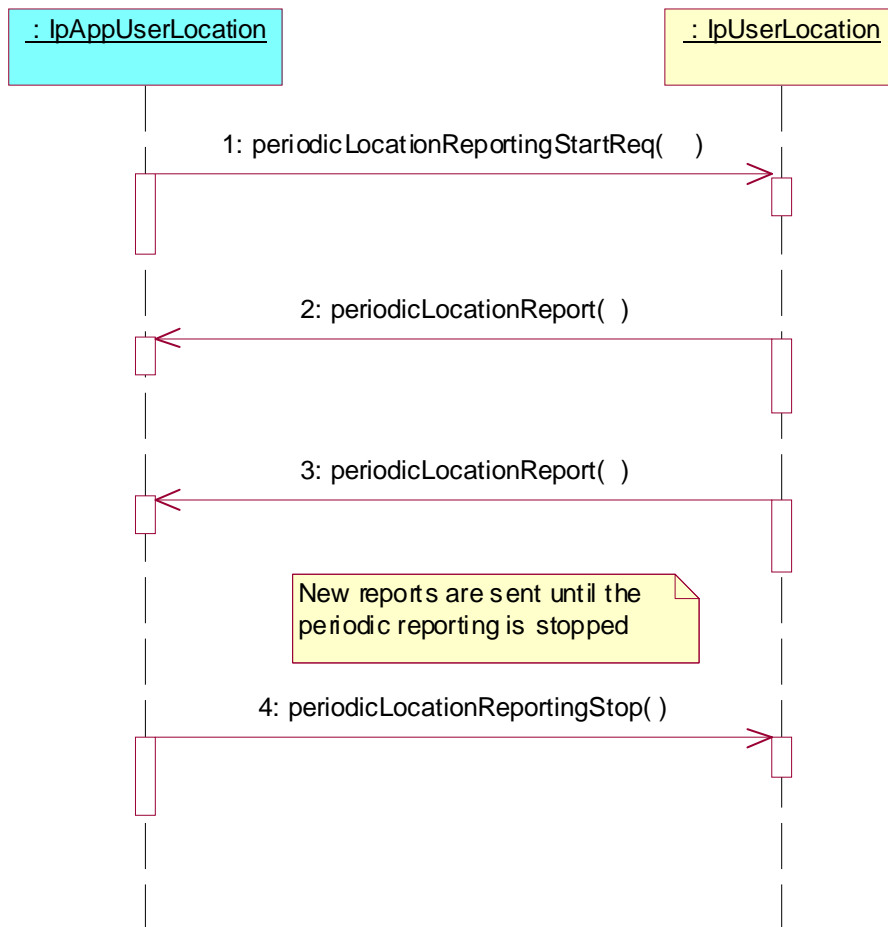
The following sequence diagram shows how an application requests triggered location reports from the User Location service. When users location changes, the service reports this to the application.



- 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.

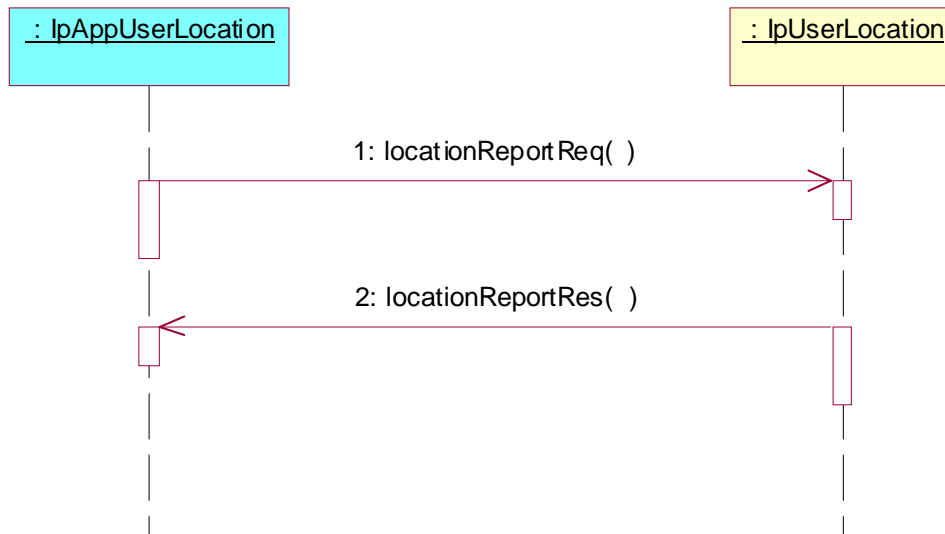
3: This message passes the location of one or several users to its callback object.

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 - Interactive Request

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



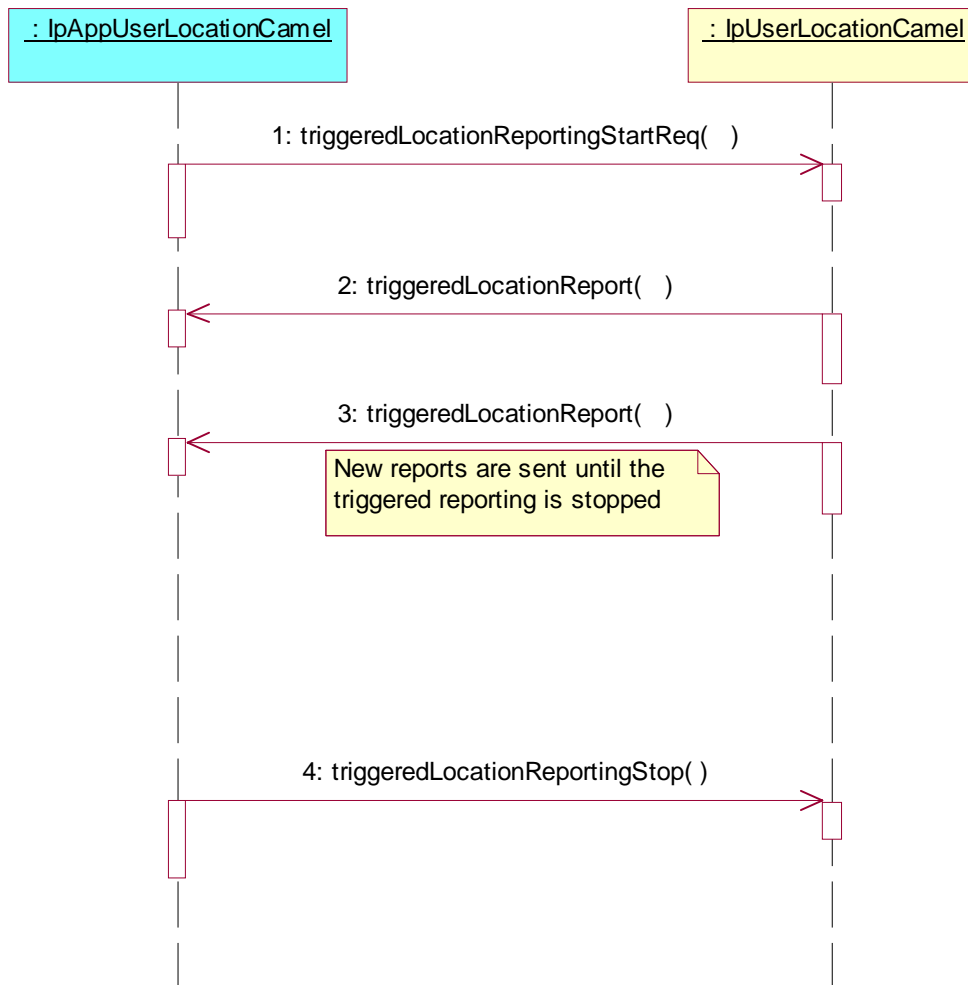
1: This message is used to request the location of one or several users.

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

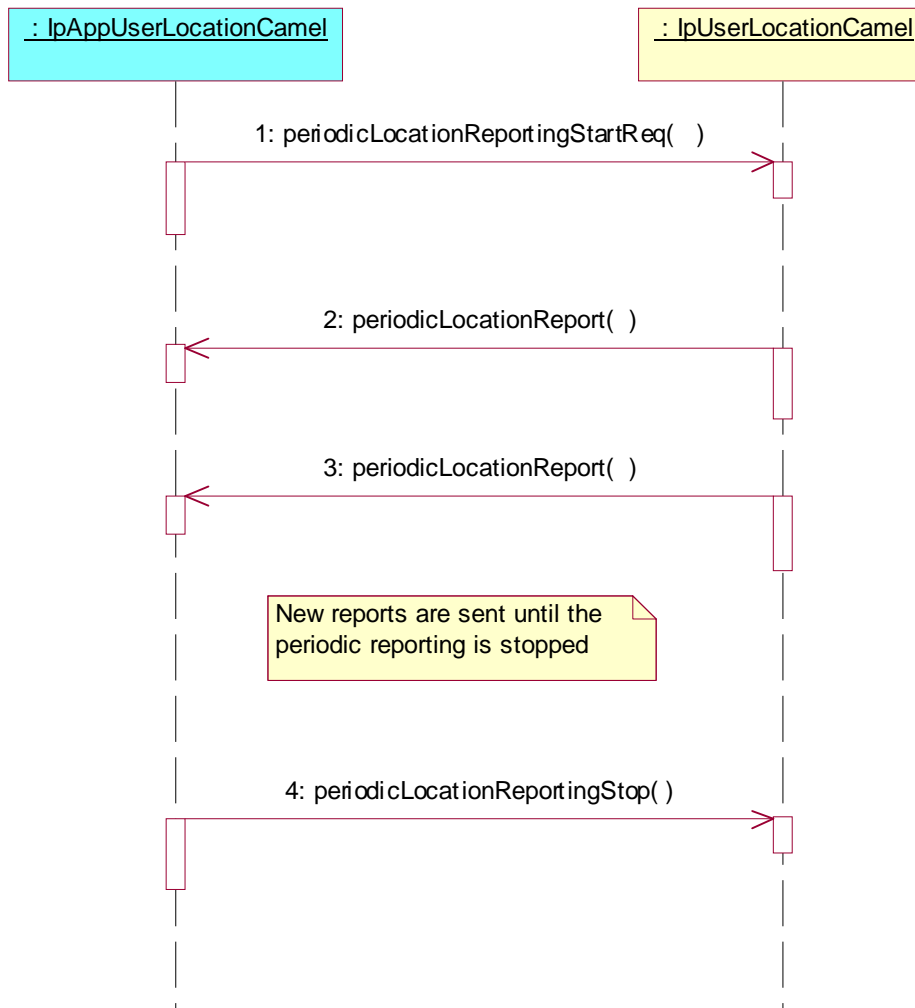
The following sequence diagram shows how an application requests triggered location reports from the User Location Camel service. When users location changes, the service reports this to the application.



- 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.

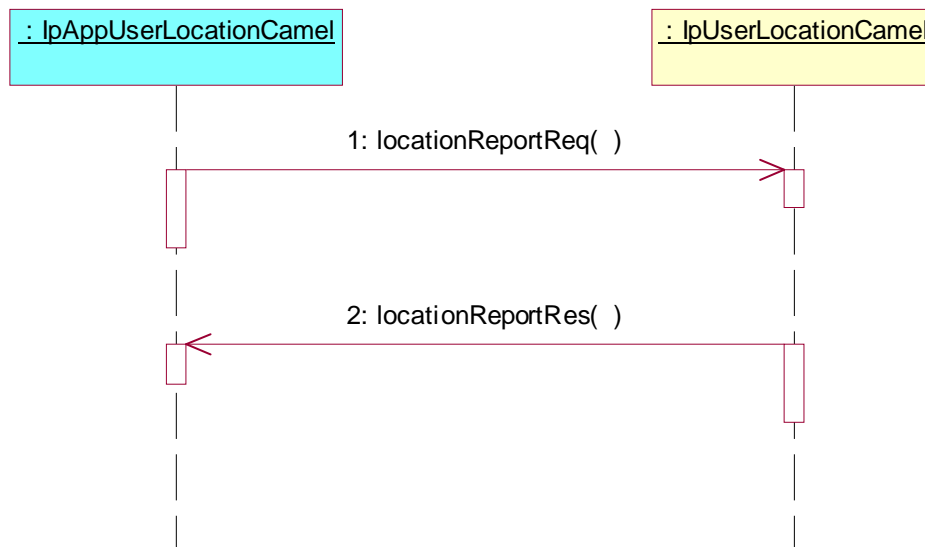
3: This message passes the location of one or several users to its callback object.

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 - 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.

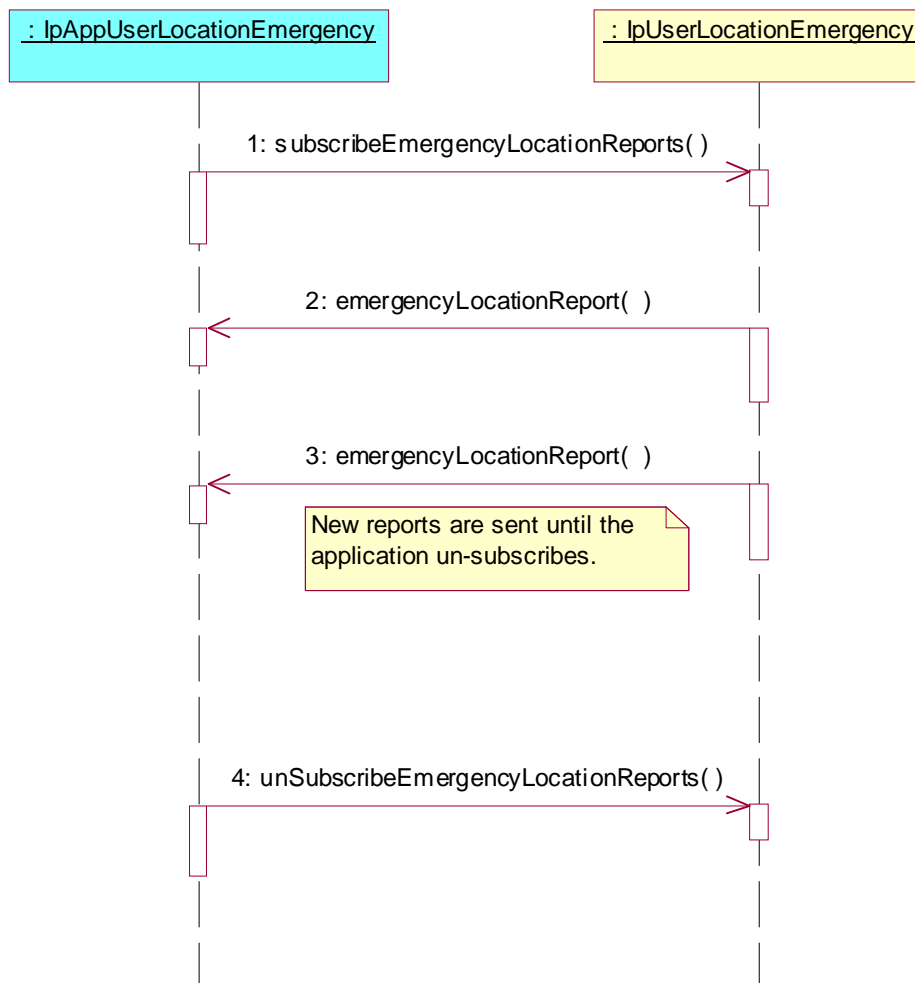
2: This message passes the result of the location request for one or several users to its callback object.

## 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.

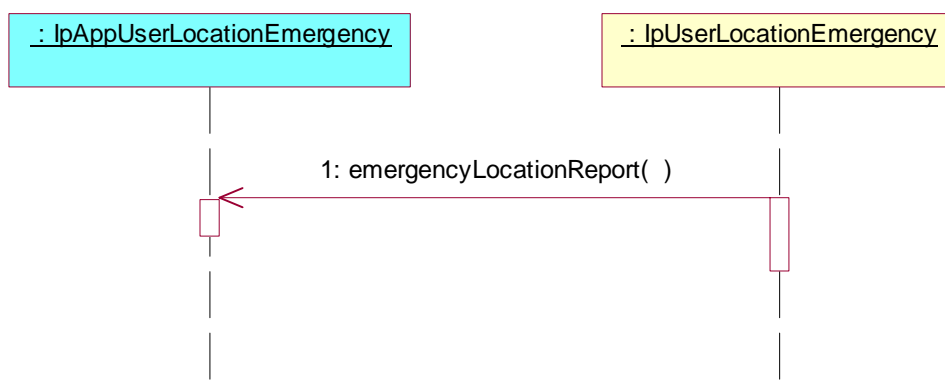
2: When the mobility service receives a Network Induced Location Request (triggered by an emergency service call), this message passes the location to its callback object.

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

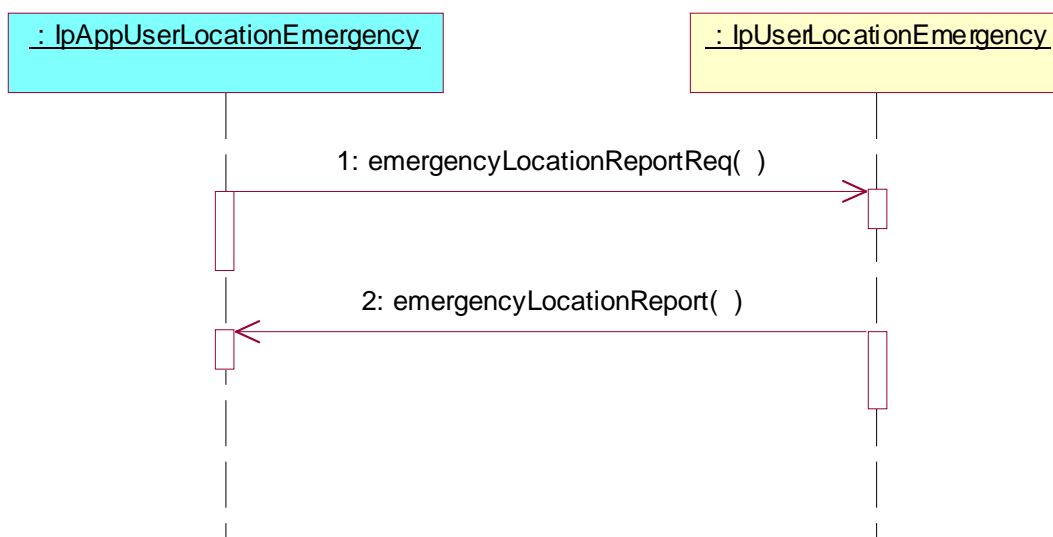
This sequence diagram is a simplified version of the previous diagram. This diagram shows only the part where an emergency call triggers a location report in the network and the location report is passed via the User Location Emergency service to a dedicated emergency application.



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 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.



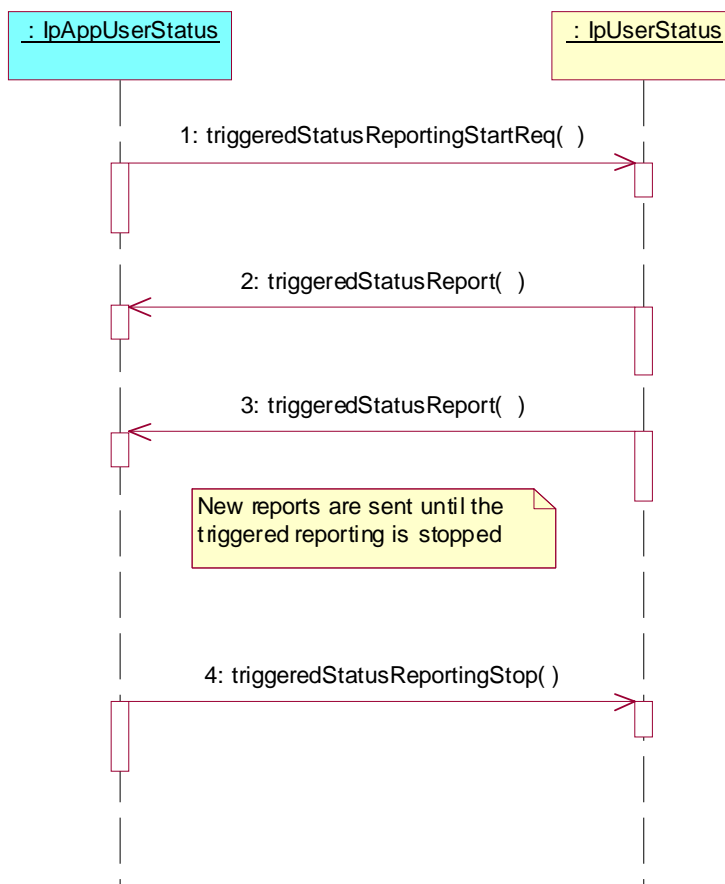
1: An emergency application makes an interactive request for a user location. This can sometimes be made with different parameters than the "ordinary" user location requests in the `IpUserLocation` interface, because a routing key and an IMSI might be used to identify the terminal (emergency call without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

2: The emergency location information is reported back to the application.

## 5.4 User Status Sequence Diagrams

### 5.4.1 Triggered Reporting

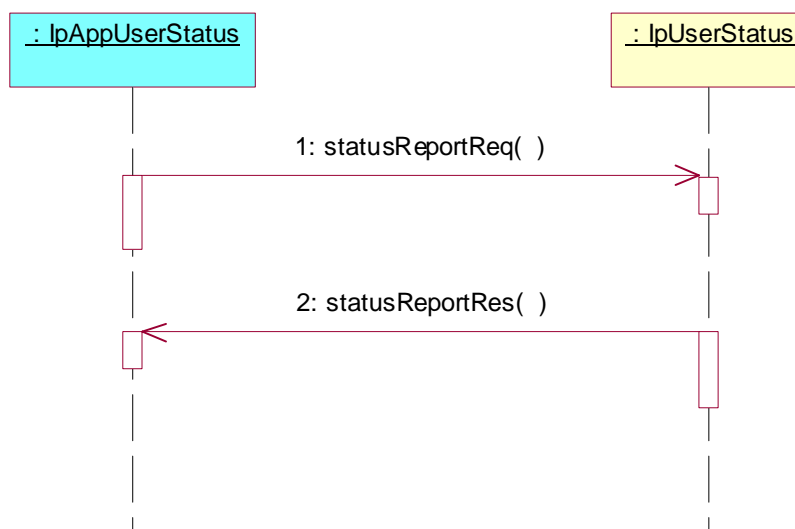
The following sequence diagram shows how an application requests triggered status reports from the Status Location service. When user's status changes, the service reports this to the application.



- 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

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.

## 5.5 User Binding Sequence Diagrams

### 5.5.1 Subscription and Binding Request Notifications

The following sequence diagram shows how an application subscribes to binding request notifications from the User Binding service. When the User Binding service receives Network Induced Binding requests (triggered by binding events) it reports that to the application.



- 1: This message is used to subscribe to notifications of binding requests.
- 2: When the mobility service receives a binding request, this message passes the binding details to its callback object.
- 3: This is repeated until the application cancels its subscription to binding request notifications (see next message).
- 4: This message is used to cancel the binding request notification subscription.

---

## 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.

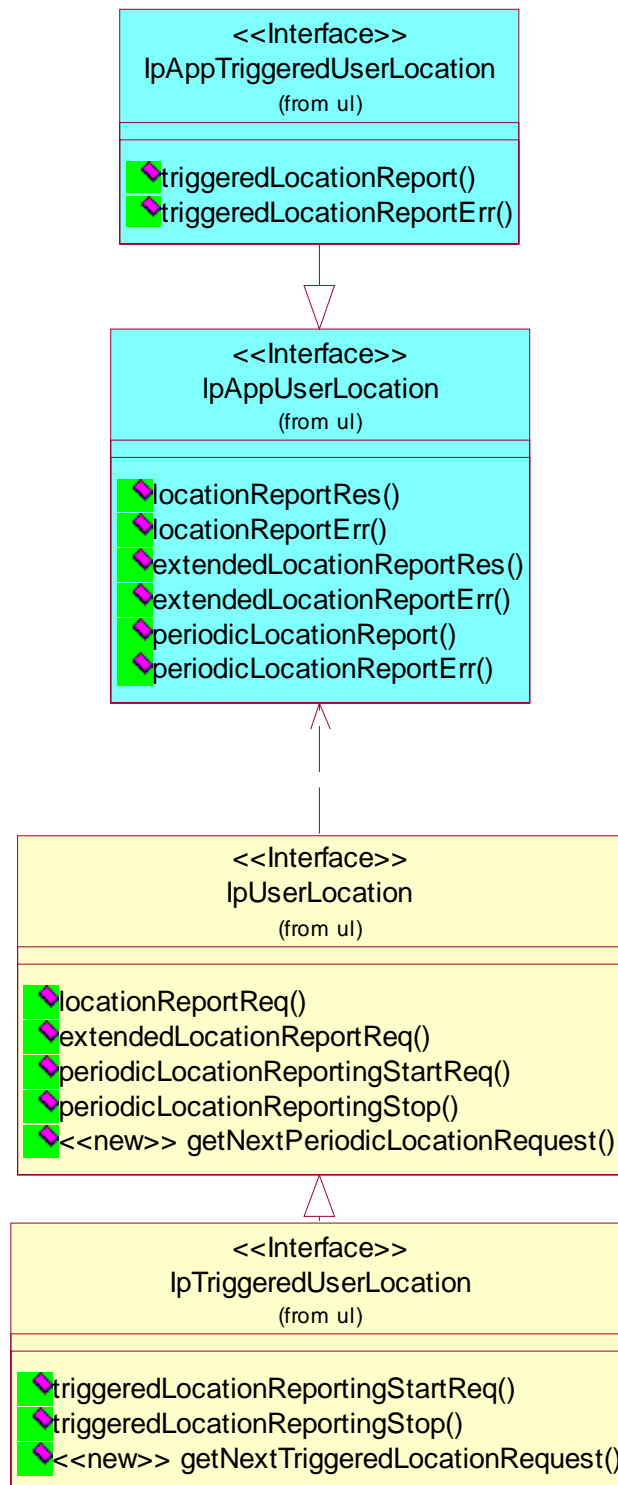


Figure 1: User Location Class Diagram

## 6.2 User Location Camel Class Diagrams

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

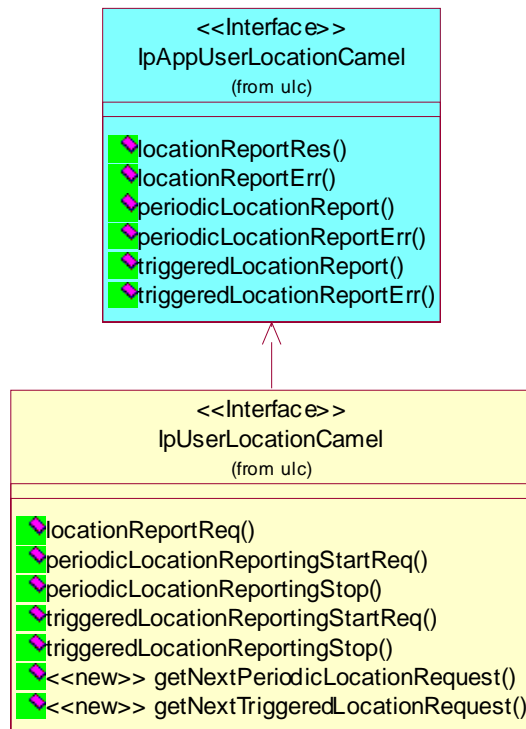


Figure 2: User Location Camel Class Diagram

## 6.3 User Location Emergency Class Diagrams

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

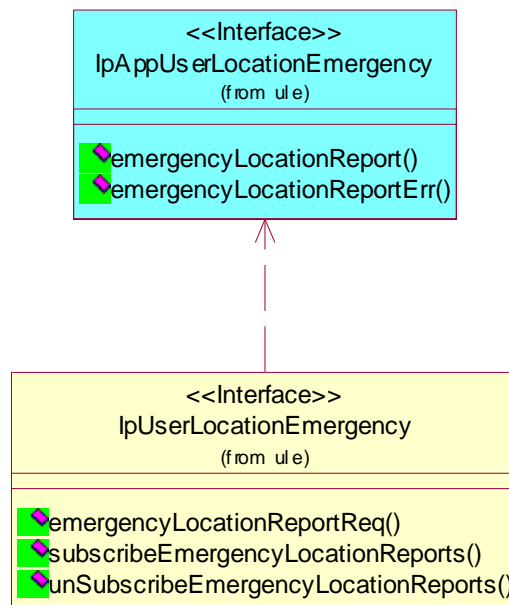


Figure 3: User Location Emergency Class Diagram

## 6.4 User Status Class Diagrams

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

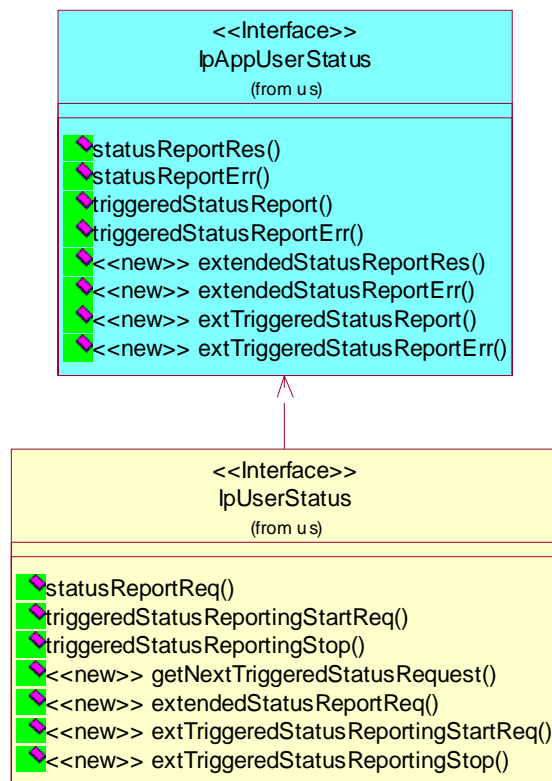


Figure 4: User Status Class Diagram

## 6.5 User Binding Class Diagrams

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

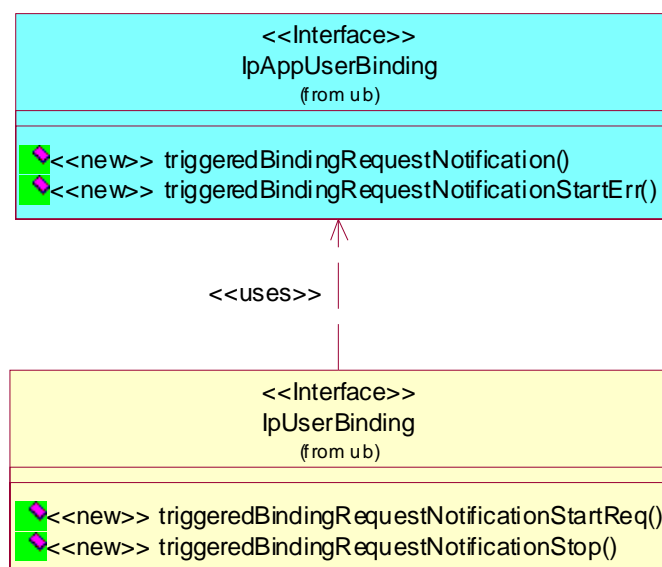


Figure 5: User Binding Class Diagram

## 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 below.

#### 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.

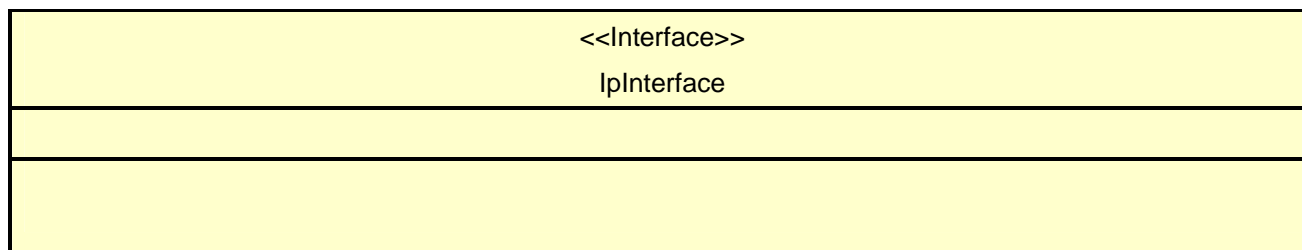
#### 7.1.4 State Model

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
<pre> setCallback (appInterface : in IpInterfaceRef) : void setCallbackWithSessionID (appInterface : in IpInterfaceRef, sessionID : in TpSessionID) : void </pre>

#### 7.4.1.1 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. Multiple invocations of this method on an interface shall result in multiple callback references being specified. The SCS shall use the most recent callback interface provided by the application using this method. In the event that a callback reference fails or is no longer available, the next most recent callback reference available shall be used.

#### *Parameters*

**appInterface : in IpInterfaceRef**

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

#### *Raises*

**TpCommonExceptions, P\_INVALID\_INTERFACE\_TYPE**

#### 7.4.1.2 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. Multiple invocations of this method on an interface shall result in multiple callback references being specified. The SCS shall use the most recent callback interface provided by the application using this method. In the event that a callback reference fails or is no longer available, the next most recent callback reference available shall be used.

*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.

This interface, or IpTriggeredUserLocation, shall be implemented by a User Location SCF as a minimum requirement.

The locationReportReq() method, or the extendedLocationReportReq() method, or both the periodicLocationReportingStartReq() and periodicLocationReportingStop() methods shall be implemented as a minimum requirement, if this interface is implemented.

<<Interface>> IpUserLocation
locationReportReq (appLocation : in IpAppUserLocationRef, users : in TpAddressSet) : TpAssignmentID extendedLocationReportReq (appLocation : in IpAppUserLocationRef, users : in TpAddressSet, request : in TpLocationRequest) : TpAssignmentID periodicLocationReportingStartReq (appLocation : in IpAppUserLocationRef, users : in TpAddressSet, request : in TpLocationRequest, reportingInterval : in TpDuration) : TpAssignmentID periodicLocationReportingStop (stopRequest : in TpMobilityStopAssignmentData) : void <<new>> getNextPeriodicLocationRequest (reset : in TpBoolean) : TpPeriodicLocationRequestSetEntry

### 8.1.1.1 Method locationReportReq()

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

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*

**TpAssignmentID**

#### *Raises*

**TpCommonExceptions, P\_APPLICATION\_NOT\_ACTIVATED, P\_INFORMATION\_NOT\_AVAILABLE, P\_INVALID\_INTERFACE\_TYPE**

### 8.1.1.2 Method extendedLocationReportReq()

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

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*

**TpAssignmentID**

*Raises*

**TpCommonExceptions, P\_APPLICATION\_NOT\_ACTIVATED,  
P\_REQUESTED\_ACCURACY\_CANNOT\_BE\_DELIVERED,  
P\_REQUESTED\_RESPONSE\_TIME\_CANNOT\_BE\_DELIVERED,  
P\_INFORMATION\_NOT\_AVAILABLE, P\_INVALID\_INTERFACE\_TYPE**

### 8.1.1.3 Method periodicLocationReportingStartReq()

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

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*

**TpAssignmentID**

*Raises*

**TpCommonExceptions, P\_INVALID\_REPORTING\_INTERVAL,  
P\_REQUESTED\_ACCURACY\_CANNOT\_BE\_DELIVERED,  
P\_REQUESTED\_RESPONSE\_TIME\_CANNOT\_BE\_DELIVERED,  
P\_APPLICATION\_NOT\_ACTIVATED, P\_INFORMATION\_NOT\_AVAILABLE,  
P\_INVALID\_INTERFACE\_TYPE**

### 8.1.1.4 Method periodicLocationReportingStop()

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

*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**

#### 8.1.1.5 Method <<new>> getNextPeriodicLocationRequest()

This method is used by the application to query the request set created with periodicLocationReportingStartReq. Since a lot of data can potentially be returned (which might cause problem in the middleware), this method must be used in an iterative way. Each method invocation may return part of the total set of requests if the set is too large to return it at once. The reset parameter permits the application to indicate whether an invocation to getNextPeriodicLocationRequest is requesting more requests from the total set of requests or is requesting that the total set of requests shall be returned from the beginning.

Returns the set of requests and an indication whether all off the requests have been obtained or if more requests are available that have not yet been obtained by the application.

Note that the (maximum) number of items provided to the application is determined by the gateway.

### *Parameters*

**reset : in TpBoolean**

TRUE: indicates that the application is intended to obtain the set of requests starting at the beginning.

FALSE: indicates that the application requests the next set of requests that have not (yet) been obtained since the last call to this method with this parameter set to TRUE.

The first time this method is invoked, reset shall be set to TRUE. Following the receipt of a final indication, for the next call to this method reset shall be set to TRUE. P\_TASK\_REFUSED may be thrown if these conditions are not met.

### *Returns*

**TpPeriodicLocationRequestSetEntry**

### *Raises*

**TpCommonExceptions**

#### 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.

<<Interface>> IpAppUserLocation
locationReportRes (assignmentId : in TpAssignmentID, locations : in TpUserLocationSet) : void locationReportErr (assignmentId : in TpAssignmentID, cause : in TpMobilityError, diagnostic : in TpMobilityDiagnostic) : void extendedLocationReportRes (assignmentId : in TpAssignmentID, locations : in TpUserLocationExtendedSet) : void extendedLocationReportErr (assignmentId : in TpAssignmentID, cause : in TpMobilityError, diagnostic : in TpMobilityDiagnostic) : void periodicLocationReport (assignmentId : in TpAssignmentID, locations : in TpUserLocationExtendedSet) : void periodicLocationReportErr (assignmentId : in TpAssignmentID, cause : in TpMobilityError, diagnostic : in TpMobilityDiagnostic) : void

### 8.1.2.1 Method locationReportRes()

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

#### *Parameters*

**assignmentId : in TpAssignmentID**

Specifies the assignment ID of the location-report request.

**locations : in TpUserLocationSet**

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

### 8.1.2.2 Method locationReportErr()

This method indicates that the location report request has failed.

#### *Parameters*

**assignmentId : in TpAssignmentID**

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.

### 8.1.2.3 Method extendedLocationReportRes()

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

#### *Parameters*

**assignmentId: in TpAssignmentID**

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

**locations: in TpUserLocationExtendedSet**

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

### 8.1.2.4 Method extendedLocationReportErr()

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

#### *Parameters*

**assignmentId: in TpAssignmentID**

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.

### 8.1.2.5 Method periodicLocationReport()

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

#### *Parameters*

**assignmentId: in TpAssignmentID**

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

**locations: in TpUserLocationExtendedSet**

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

### 8.1.2.6 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 TpAssignmentID**

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

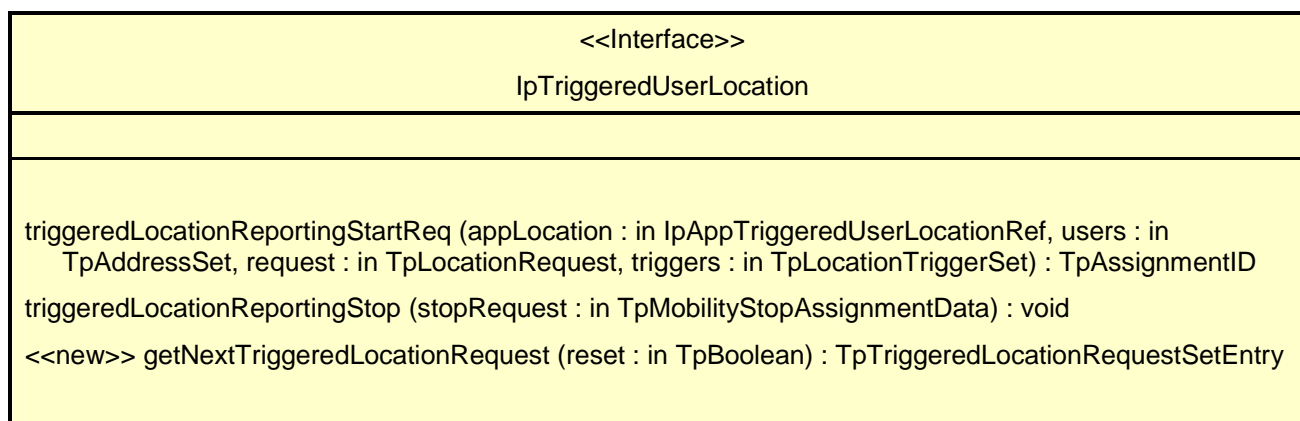
Inherits from: IpUserLocation.

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.

This interface, or IpUserLocation, shall be implemented by a User Location SCF as a minimum requirement.

The triggeredLocationReportingStartReq() and triggeredLocationReportingStop() methods shall be implemented as a minimum requirement. An implementation of IpTriggeredUserLocation is not required to implement the minimum mandatory methods of IpUserLocation.



#### 8.1.3.1 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***TpAssignmentID***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**

**8.1.3.2 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**

**8.1.3.3 Method <<new>> getNextTriggeredLocationRequest()**

This method is used by the application to query the request set created with triggeredLocationReportingStartReq. Since a lot of data can potentially be returned (which might cause problem in the middleware), this method must be used in an iterative way. Each method invocation may return part of the total set of requests if the set is too large to return it at once. The reset parameter permits the application to indicate whether an invocation to getNextTriggeredLocationRequest is requesting more requests from the total set of requests or is requesting that the total set of requests shall be returned from the beginning.

Returns the set of requests and an indication whether all off the requests have been obtained or if more requests are available that have not yet been obtained by the application.

Note that the (maximum) number of items provided to the application is determined by the gateway.

*Parameters*

**reset**: in **TpBoolean**

TRUE: indicates that the application is intended to obtain the set of requests starting at the beginning.

FALSE: indicates that the application requests the next set of requests that have not (yet) been obtained since the last call to this method with this parameter set to TRUE.

The first time this method is invoked, reset shall be set to TRUE. Following the receipt of a final indication, for the next call to this method reset shall be set to TRUE. P\_TASK\_REFUSED may be thrown if these conditions are not met.

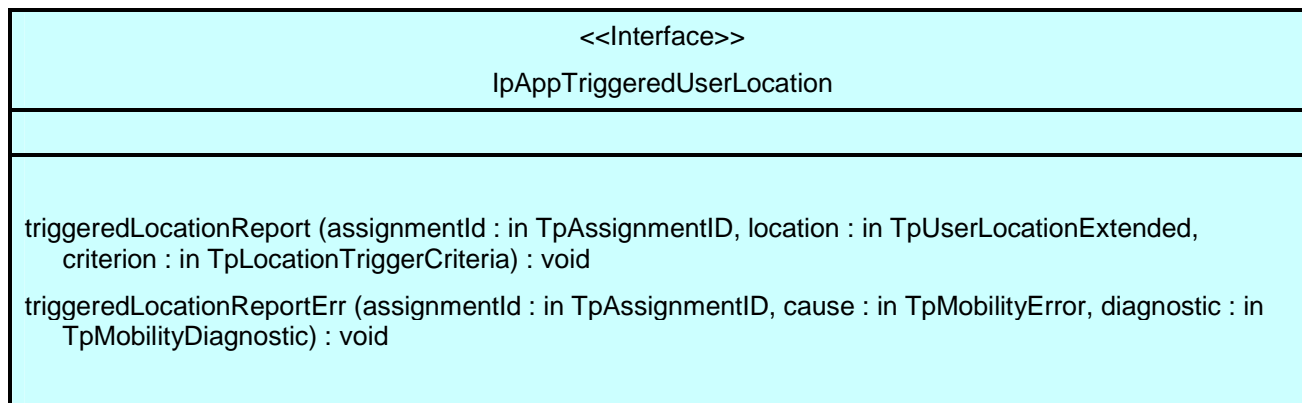
*Returns***TpTriggeredLocationRequestSetEntry***Raises***TpCommonExceptions**

## 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.



### 8.1.4.1 Method triggeredLocationReport()

A triggered report containing location for a user is delivered.

#### *Parameters*

**assignmentId : in TpAssignmentID**

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.

### 8.1.4.2 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 TpAssignmentID**

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 coordinates 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.

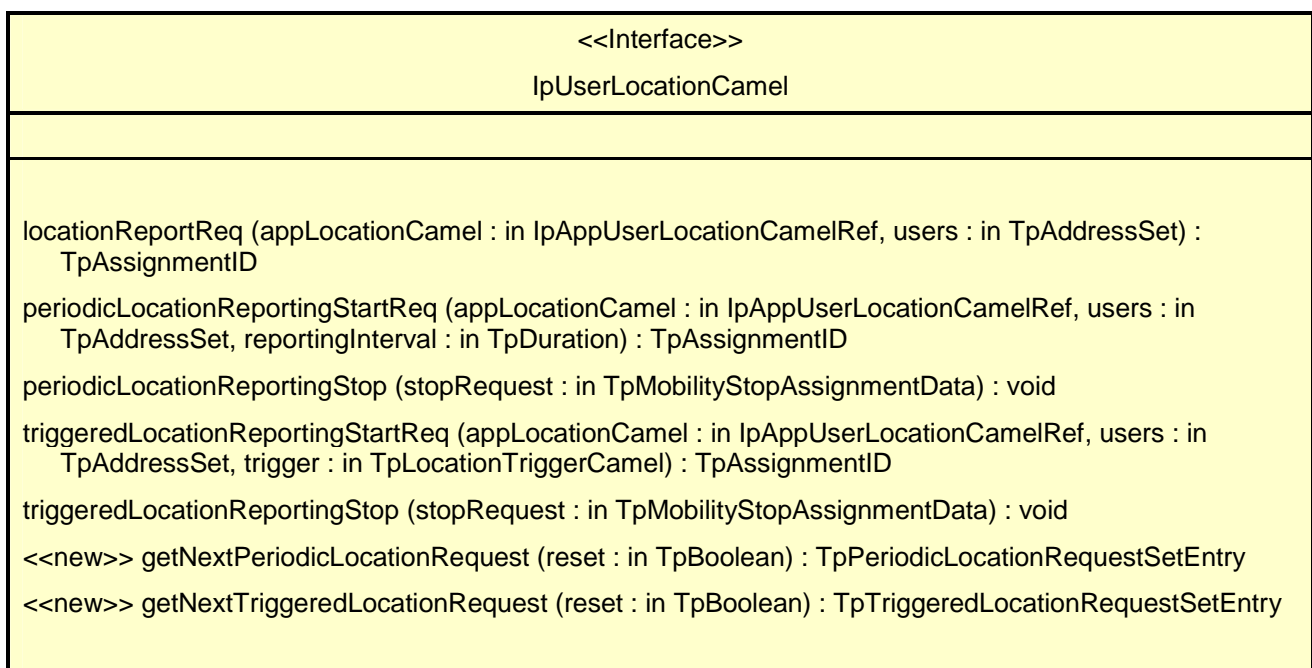
### 8.2.1 Interface Class `IpUserLocationCamel`

Inherits from: `IpService`.

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

This interface shall be implemented by a User Location Camel SCF.

The `locationReportReq()` method, or both the `periodicLocationReportingStartReq()` and `periodicLocationReportingStop()` methods, or both the `triggeredLocationReportingStartReq()` and `triggeredLocationReportingStop()` methods shall be implemented as a minimum requirement.



#### 8.2.1.1 Method `locationReportReq()`

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

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*

**TpAssignmentID**

*Raises*

**TpCommonExceptions, P\_UNKNOWN\_SUBSCRIBER, P\_APPLICATION\_NOT\_ACTIVATED, P\_INFORMATION\_NOT\_AVAILABLE, P\_INVALID\_INTERFACE\_TYPE**

### 8.2.1.2 Method periodicLocationReportingStartReq()

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

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*

**TpAssignmentID**

*Raises*

**TpCommonExceptions, P\_INVALID\_REPORTING\_INTERVAL, P\_UNKNOWN\_SUBSCRIBER, P\_APPLICATION\_NOT\_ACTIVATED, P\_INFORMATION\_NOT\_AVAILABLE, P\_INVALID\_INTERFACE\_TYPE**

### 8.2.1.3 Method periodicLocationReportingStop()

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

*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**

#### 8.2.1.4 Method triggeredLocationReportingStartReq()

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

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*

**TpAssignmentID**

##### *Raises*

**TpCommonExceptions, P\_UNKNOWN\_SUBSCRIBER, P\_APPLICATION\_NOT\_ACTIVATED, P\_INFORMATION\_NOT\_AVAILABLE, P\_INVALID\_INTERFACE\_TYPE**

#### 8.2.1.5 Method triggeredLocationReportingStop()

Request that triggered mobile location reporting should stop.

##### *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**

#### 8.2.1.6 Method <<new>> getNextPeriodicLocationRequest()

This method is used by the application to query the request set created with periodicLocationReportingStartReq. Since a lot of data can potentially be returned (which might cause problem in the middleware), this method must be used in an iterative way. Each method invocation may return part of the total set of requests if the set is too large to return it at once. The reset parameter permits the application to indicate whether an invocation to getNextPeriodicLocationRequest is requesting more requests from the total set of requests or is requesting that the total set of requests shall be returned from the beginning.

Returns the set of requests and an indication whether all off the requests have been obtained or if more requests are available that have not yet been obtained by the application.

Note that the (maximum) number of items provided to the application is determined by the gateway.

##### *Parameters*

**reset: in TpBoolean**

TRUE: indicates that the application is intended to obtain the set of requests starting at the beginning.

**FALSE:** indicates that the application requests the next set of requests that have not (yet) been obtained since the last call to this method with this parameter set to **TRUE**.

The first time this method is invoked, **reset** shall be set to **TRUE**. Following the receipt of a final indication, for the next call to this method **reset** shall be set to **TRUE**. **P\_TASK\_REFUSED** may be thrown if these conditions are not met.

### *Returns*

**TpPeriodicLocationRequestSetEntry**

### *Raises*

**TpCommonExceptions**

## 8.2.1.7 Method <<new>> getNextTriggeredLocationRequest()

This method is used by the application to query the request set created with **triggeredLocationReportingStartReq**. Since a lot of data can potentially be returned (which might cause problem in the middleware), this method must be used in an iterative way. Each method invocation may return part of the total set of requests if the set is too large to return it at once. The **reset** parameter permits the application to indicate whether an invocation to **getNextTriggeredLocationRequest** is requesting more requests from the total set of requests or is requesting that the total set of requests shall be returned from the beginning.

Returns the set of requests and an indication whether all off the requests have been obtained or if more requests are available that have not yet been obtained by the application.

Note that the (maximum) number of items provided to the application is determined by the gateway.

### *Parameters*

**reset : in TpBoolean**

**TRUE:** indicates that the application is intended to obtain the set of requests starting at the beginning.

**FALSE:** indicates that the application requests the next set of requests that have not (yet) been obtained since the last call to this method with this parameter set to **TRUE**.

The first time this method is invoked, **reset** shall be set to **TRUE**. Following the receipt of a final indication, for the next call to this method **reset** shall be set to **TRUE**. **P\_TASK\_REFUSED** may be thrown if these conditions are not met.

### *Returns*

**TpTriggeredLocationRequestSetEntry**

### *Raises*

**TpCommonExceptions**

## 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.

<<Interface>> IpAppUserLocationCamel
locationReportRes (assignmentId : in TpAssignmentID, locations : in TpUserLocationCamelSet) : void locationReportErr (assignmentId : in TpAssignmentID, cause : in TpMobilityError, diagnostic : in TpMobilityDiagnostic) : void periodicLocationReport (assignmentId : in TpAssignmentID, locations : in TpUserLocationCamelSet) : void periodicLocationReportErr (assignmentId : in TpAssignmentID, cause : in TpMobilityError, diagnostic : in TpMobilityDiagnostic) : void triggeredLocationReport (assignmentId : in TpAssignmentID, location : in TpUserLocationCamel, criterion : in TpLocationTriggerCamel) : void triggeredLocationReportErr (assignmentId : in TpAssignmentID, cause : in TpMobilityError, diagnostic : in TpMobilityDiagnostic) : void

### 8.2.2.1 Method locationReportRes()

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

#### *Parameters*

**assignmentId : in TpAssignmentID**

Specifies the assignment ID of the location-report request.

**locations : in TpUserLocationCamelSet**

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

### 8.2.2.2 Method locationReportErr()

This method indicates that the location report request has failed.

#### *Parameters*

**assignmentId : in TpAssignmentID**

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.

### 8.2.2.3 Method periodicLocationReport()

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

#### *Parameters*

**assignmentId: in TpAssignmentID**

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

**locations: in TpUserLocationCamelSet**

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

### 8.2.2.4 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 TpAssignmentID**

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.

### 8.2.2.5 Method triggeredLocationReport()

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

#### *Parameters*

**assignmentId: in TpAssignmentID**

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.

### 8.2.2.6 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 TpAssignmentID**

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.

This interface shall be implemented by a Mobility SCF.

The emergencyLocationReportReq() method, or the subscribeEmergencyLocationReports() and unsubscribeEmergencyLocationReports() methods shall be implemented as a minimum requirement.

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

#### 8.3.1.1 Method emergencyLocationReportReq()

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

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*

**TpAssignmentID**

*Raises*

**TpCommonExceptions, P\_UNKNOWN\_SUBSCRIBER, P\_INFORMATION\_NOT\_AVAILABLE, P\_APPLICATION\_NOT\_ACTIVATED, P\_INVALID\_INTERFACE\_TYPE**

**8.3.1.2 Method subscribeEmergencyLocationReports()**

Subscribe to network initiated emergency user location reports.

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*

**TpAssignmentID**

*Raises*

**TpCommonExceptions, P\_INVALID\_INTERFACE\_TYPE**

**8.3.1.3 Method unsubscribeEmergencyLocationReports()**

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

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

*Parameters*

**assignmentId:in TpAssignmentID**

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.

<<Interface>> IpAppUserLocationEmergency
emergencyLocationReport (assignmentId : in TpAssignmentID, location : in TpUserLocationEmergency) : void emergencyLocationReportErr (assignmentId : in TpAssignmentID, cause : in TpMobilityError, diagnostic : in TpMobilityDiagnostic) : void

### 8.3.2.1 Method emergencyLocationReport()

Delivery of an emergency user location report.

#### *Parameters*

**assignmentId : in TpAssignmentID**

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.

### 8.3.2.2 Method emergencyLocationReportErr()

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

#### *Parameters*

**assignmentId : in TpAssignmentID**

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

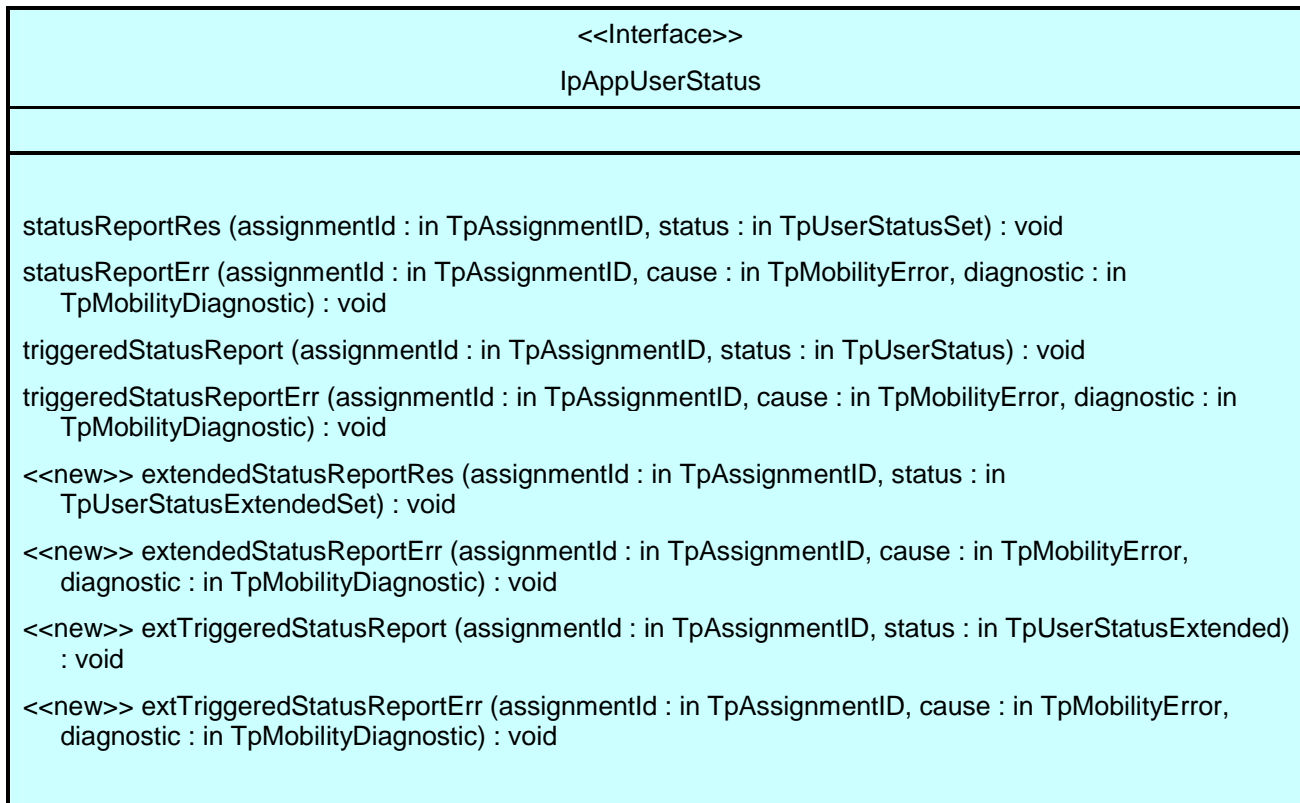
The User Status Service (US) provides a general user status service. US allow applications to obtain the status of fixed, mobile and IP-based telephony users.

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.



### 8.4.1.1 Method statusReportRes()

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

#### *Parameters*

**assignmentId : in TpAssignmentID**

Specifies the assignment ID of the status-report request.

**status : in TpUserStatusSet**

Specifies the status of one or several users.

### 8.4.1.2 Method statusReportErr()

This method indicates that the status report request has failed.

#### *Parameters*

**assignmentId : in TpAssignmentID**

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.

**8.4.1.3 Method triggeredStatusReport()**

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

*Parameters***assignmentId : in TpAssignmentID**

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

**status : in TpUserStatus**

Specifies the status of the user.

**8.4.1.4 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 TpAssignmentID**

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.

**8.4.1.5 Method <<new>> extendedStatusReportRes()**

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

*Parameters***assignmentId : in TpAssignmentID**

Specifies the assignment ID of the status-report request.

**status : in TpUserStatusExtendedSet**

Specifies the status of one or several users.

**8.4.1.6 Method <<new>> extendedStatusReportErr()**

This method indicates that the status report request has failed.

*Parameters***assignmentId : in TpAssignmentID**

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.

**8.4.1.7 Method <<new>> extTriggeredStatusReport()**

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

*Parameters***assignmentId : in TpAssignmentID**

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

**status : in TpUserStatusExtended**

Specifies the status of the user.

**8.4.1.8 Method <<new>> extTriggeredStatusReportErr()**

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

*Parameters***assignmentId : in TpAssignmentID**

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.

This interface shall be implemented by a User Status SCF.

The statusReportReq() method, or both the triggeredStatusReportingStartReq() and triggeredStatusReportingStop() methods shall be implemented as a minimum requirement.

<<Interface>> IpUserStatus
<pre> statusReportReq (appStatus : in IpAppUserStatusRef, users : in TpAddressSet) : TpAssignmentID triggeredStatusReportingStartReq (appStatus : in IpAppUserStatusRef, users : in TpAddressSet) :   TpAssignmentID triggeredStatusReportingStop (stopRequest : in TpMobilityStopAssignmentData) : void &lt;&lt;new&gt;&gt; getNextTriggeredStatusRequest (reset : in TpBoolean) : TpTriggeredStatusRequestSetEntry &lt;&lt;new&gt;&gt; extendedStatusReportReq (appStatus : in IpAppUserStatusRef, users : in TpAddressSet) :   TpAssignmentID &lt;&lt;new&gt;&gt; extTriggeredStatusReportingStartReq (appStatus : in IpAppUserStatusRef, users : in   TpAddressSet) : TpAssignmentID &lt;&lt;new&gt;&gt; extTriggeredStatusReportingStop (stopRequest : in TpMobilityStopAssignmentData) : void           </pre>

#### 8.4.2.1 Method statusReportReq()

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

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*

**TpAssignmentID**

##### *Raises*

**TpCommonExceptions, P\_UNKNOWN\_SUBSCRIBER, P\_INFORMATION\_NOT\_AVAILABLE, P\_APPLICATION\_NOT\_ACTIVATED, P\_INVALID\_INTERFACE\_TYPE**

#### 8.4.2.2 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.

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*

**TpAssignmentID**

*Raises*

**TpCommonExceptions, P\_UNKNOWN\_SUBSCRIBER, P\_INFORMATION\_NOT\_AVAILABLE, P\_APPLICATION\_NOT\_ACTIVATED, P\_INVALID\_INTERFACE\_TYPE**

**8.4.2.3 Method triggeredStatusReportingStop()**

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

*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**

**8.4.2.4 Method <<new>> getNextTriggeredStatusRequest()**

This method is used by the application to query the request set created with triggeredStatusReportingStartReq. Since a lot of data can potentially be returned (which might cause problem in the middleware), this method must be used in an iterative way. Each method invocation may return part of the total set of requests if the set is too large to return it at once. The reset parameter permits the application to indicate whether an invocation to getNextTriggeredStatusRequest is requesting more requests from the total set of requests or is requesting that the total set of requests shall be returned from the beginning.

Returns the set of requests and an indication whether all off the requests have been obtained or if more requests are available that have not yet been obtained by the application.

Note that the (maximum) number of items provided to the application is determined by the gateway.

*Parameters*

**reset : in TpBoolean**

TRUE: indicates that the application is intended to obtain the set of requests starting at the beginning.

FALSE: indicates that the application requests the next set of requests that have not (yet) been obtained since the last call to this method with this parameter set to TRUE.

The first time this method is invoked, reset shall be set to TRUE. Following the receipt of a final indication, for the next call to this method reset shall be set to TRUE. P\_TASK\_REFUSED may be thrown if these conditions are not met.

*Returns*

**TpTriggeredStatusRequestSetEntry**

*Raises*

**TpCommonExceptions**



#### 8.4.2.5 Method <<new>> extendedStatusReportReq()

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

A user can be identified by the following address types:

- MSISDN indicated with address plan P\_ADDRESS\_PLAN\_E164.
- IP Address indicated with address plan P\_ADDRESS\_PLAN\_IP.
- User Name indicated with address plan P\_ADDRESS\_PLAN\_UNDEFINED.

Returns: assignmentID.

##### *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*

**TpAssignmentID**

##### *Raises*

**TpCommonExceptions, P\_UNKNOWN\_SUBSCRIBER, P\_INFORMATION\_NOT\_AVAILABLE, P\_APPLICATION\_NOT\_ACTIVATED, P\_INVALID\_INTERFACE\_TYPE**

#### 8.4.2.6 Method <<new>> extTriggeredStatusReportingStartReq()

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.

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*

**TpAssignmentID**

##### *Raises*

**TpCommonExceptions, P\_UNKNOWN\_SUBSCRIBER, P\_INFORMATION\_NOT\_AVAILABLE, P\_APPLICATION\_NOT\_ACTIVATED, P\_INVALID\_INTERFACE\_TYPE**

### 8.4.2.7 Method <<new>> extTriggeredStatusReportingStop()

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

#### *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**

## 8.5 User Binding Interface Classes

In order to create a feature rich services creation environment for (mobile) "IP" networks where not only terminal mobility but also personal mobility is supported, User Binding notification functions, which enable applications to know UE binding requests, control and make use of them, are provided. The interfaces detailed below support UE binding notification functions.

### 8.5.1 Interface Class IpUserBinding

Inherits from: IpService.

The application programmer can use this interface to obtain binding details of users who bind to the network.

<<Interface>> IpUserBinding
<<new>> triggeredBindingRequestNotificationStartReq (appStatus : in IpAppUserBindingRef, users : in TpAddressSet, criteria : in TpBindingNotificationCriteriaSet) : TpAssignmentID <<new>> triggeredBindingRequestNotificationStop (assignmentId : in TpAssignmentID) : void

#### 8.5.1.1 Method <<new>> triggeredBindingRequestNotificationStartReq()

Request for triggered status reports when one or several user's binding details are requested to be changed. The User Binding service will send a notification when the user's binding details are requested to be changed.

Returns the assignment ID of the triggered binding request notification.

#### *Parameters*

**appStatus : in IpAppUserBindingRef**

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

**users : in TpAddressSet**

Specifies the user(s) for which the binding change requests shall be reported.

**criteria : in TpBindingNotificationCriteriaSet**

Specifies the criteria that must be matched by the binding change request notifications.

*Returns***TpAssignmentID***Raises***TpCommonExceptions, P\_UNKNOWN\_SUBSCRIBER, P\_INFORMATION\_NOT\_AVAILABLE, P\_APPLICATION\_NOT\_ACTIVATED, P\_INVALID\_INTERFACE\_TYPE****8.5.1.2 Method <<new>> triggeredBindingRequestNotificationStop()**

This method stops the sending of the notifications of binding change requests identified by the assignment ID.

*Parameters***assignmentId: in TpAssignmentID**

Specifies the assignment ID of the triggered binding-request notification.

*Raises***TpCommonExceptions, P\_INVALID\_ASSIGNMENT\_ID****8.5.2 Interface Class IpAppUserBinding**

Inherits from: IpInterface.

The user-binding application interface is implemented by the client application developer and is used to handle binding notification requests.

<<Interface>> IpAppUserBinding
<<new>> triggeredBindingRequestNotification (assignmentId : in TpAssignmentID, address : in TpAddress, contactInformation : in TpBindingSet) : TpBindingSet <<new>> triggeredBindingRequestNotificationStartErr (assignmentId : in TpAssignmentID, cause : in TpMobilityError, diagnostic : in TpMobilityDiagnostic) : void

**8.5.2.1 Method <<new>> triggeredBindingRequestNotification()**

Delivery of a report that is indicating that a user's binding details are requested to be changed. The application returns acceptable bindings.

*Parameters***assignmentId: in TpAssignmentID**

Specifies the assignment ID of the triggered binding-request notification.

**address: in TpAddress**

Specifies the address of the user whose binding details are requested to be changed.

**contactInformation: in TpBindingSet**

Specifies the binding details of the user.

*Returns*

**TpBindingSet**

### 8.5.2.2 Method <<new>> triggeredBindingRequestNotificationStartErr()

This method indicates that the request for notification of binding requests has failed.

*Parameters*

**assignmentId: in TpAssignmentID**

Specifies the assignment ID of the failed request for notification of binding requests.

**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.

## 9 State Transition Diagrams

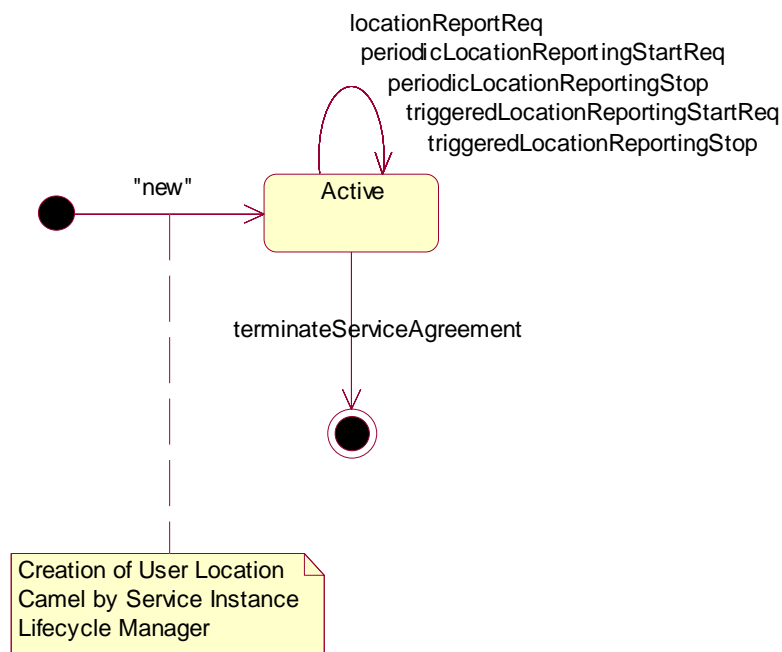
### 9.1 User Location

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 6: State Transition Diagram for User Location Camel**

### 9.2.1.1 Active State

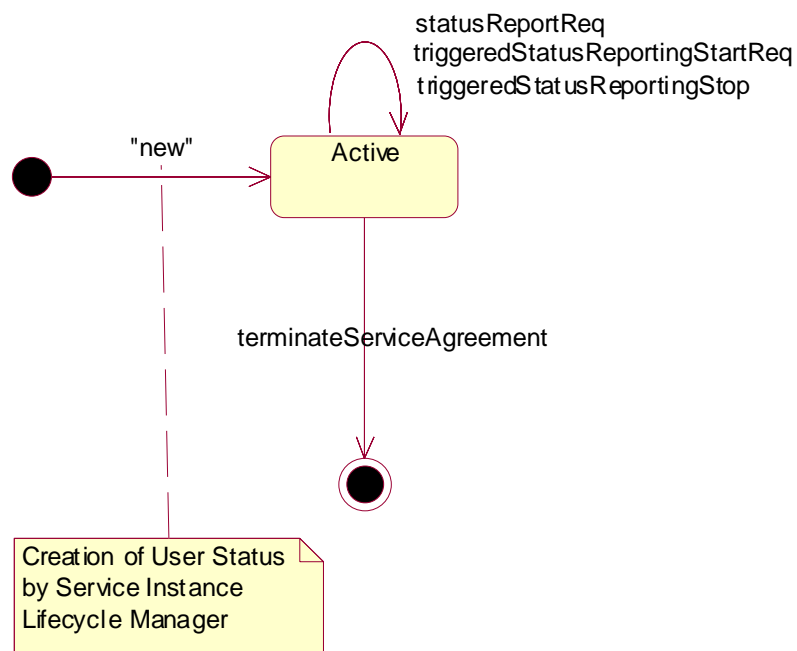
In this state, a relation between the Application and the Network User Location Service Capability Feature has been established. It allows the application to request a specific user location reports, subscribe to periodic user location reports or subscribe to triggers that generate location report when 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 7: 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.

## 9.5 User Binding

There are no State Transition Diagrams for User Binding.

---

## 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.](#)

EXAMPLE:

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 203 915-2.

## 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.

Type of uncertainty shape	Uncertainty Outer Semi Major	Uncertainty Outer Semi Minor	Uncertainty Inner Semi Major	Uncertainty Inner Semi Minor	Angle Of Semi Major	Segment Start Angle	Segment End Angle
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

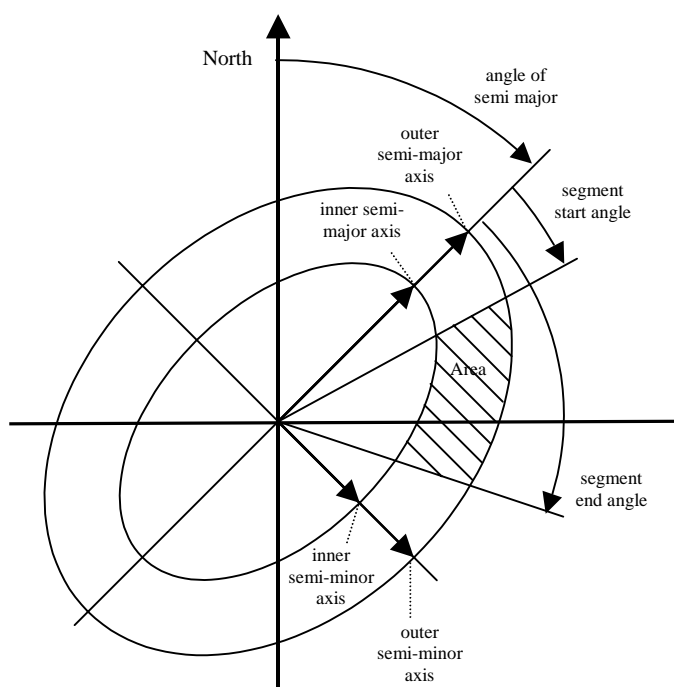


Figure 8: Description of an Ellipse Arc

## 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 binding details, location or status.	No
P_M_UNAUTHORIZED_APPLICATION	3	Unauthorised application. The application is not authorised to obtain the user's binding details, 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	TpAssignmentID	Identity of the assignment 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	TpUIExtendedDataSet	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.

### 11.2.18 TpTriggeredLocationRequestSetEntry

Defines the Sequence of Data Elements that specify a set of triggered requests and an indication whether more triggered location requests can be requested.

Sequence Element Name	Sequence Element Type	Description
Requests	TpTriggeredLocationRequestSet	Numbered set of requests.
Final	TpBoolean	Indication whether the set of triggered requests is the final set (TRUE) or if there are more triggered requests (FALSE).

### 11.2.19 TpTriggeredLocationRequestSet

Defines a numbered Set of Data Elements of TpTriggeredLocationRequest.

### 11.2.20 TpTriggeredLocationRequest

Defines the Sequence of Data Elements that specify the criteria relating to triggered location requests.

Sequence Element Name	Sequence Element Type
Users	TpAddressSet
Request	TpLocationRequest
AssignmentID	TpAssignmentID

### 11.2.21 TpPeriodicLocationRequestSetEntry

Defines the Sequence of Data Elements that specify a set of triggered requests and an indication whether more periodic location requests can be requested.

Sequence Element Name	Sequence Element Type	Description
Requests	TpPeriodicLocationRequestSet	Numbered set of requests.
Final	TpBoolean	Indication whether the set of triggered requests is the final set (TRUE) or if there are more triggered requests (FALSE).

### 11.2.22 TpPeriodicLocationRequestSet

Defines a numbered Set of Data Elements of TpPeriodicLocationRequest.

### 11.2.23 TpPeriodicLocationRequest

Defines the Sequence of Data Elements that specify the criteria relating to periodic location requests.

Sequence Element Name	Sequence Element Type
Users	TpAddressSet
Request	TpLocationRequest
ReportingInterval	TpDuration
AssignmentID	TpAssignmentID



## 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 Status Code 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.
NaEsrPresent	TpBoolean	Flag indicating if the NaESRD is present.
NaEsr	TpNaESRD	Current NaESRD for the user.
NaEsrkPresent	TpBoolean	Flag indicating if the NaESRK is present.
NaEsrk	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.
NaEsrPresent	TpBoolean	Flag indicating if the NaESRD is present.
NaEsr	TpNaESRD	Current NaESRD for the user.
NaEsrkPresent	TpBoolean	Flag indicating if the NaESRK is present.
NaEsrk	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.		

### 11.5.8 TpTriggeredStatusRequestSetEntry

Defines the Sequence of Data Elements that specify a set of triggered requests and an indication whether more triggered status requests can be requested.

Sequence Element Name	Sequence Element Type	Description
Requests	TpTriggeredStatusRequestSet	Numbered set of requests.
Final	TpBoolean	Indication whether the set of triggered requests is the final set (TRUE) or if there are more triggered requests (FALSE).

### 11.5.9 TpTriggeredStatusRequestSet

Defines a numbered Set of Data Elements of TpTriggeredStatusRequest.

### 11.5.10 TpTriggeredStatusRequest

Defines the Sequence of Data Elements that specify the criteria relating to triggered status requests.

Sequence Element Name	Sequence Element Type
Users	TpAddressSet
AssignmentID	TpAssignmentID

### 11.5.11 TpUserStatusExtended

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	TpUserStatusIndicatorExtended	The current status of the user.
TerminalType	TpTerminalType	The kind of terminal used by the user.
AuthenticationStatus	TpAuthStatusIndicator	The current authentication status of the user
NetworkStatus	TpNetworkStatusIndicator	Country code, Mobile network code, technology and Roaming status.
UserIPAddress	TpString	The allocated IP address for the user. If there is no IP address allocated then this is represented with the empty string.
UserMSISDN	TpString	The user MSISDN. If the MSISDN address is not known, then this is represented with the empty string.
UserInfo	TpUserInfo	Used to reference the user name and password supplied by the user during user-network authentication, if known by the network.
UserConnectionID	TpString	Combination of User IP Address and TCP port. This combination uniquely identifies the user's TCP connection, because one user can have several TCP connections ongoing at the same time. The User IP Address contained in UserConnectionID can be the same as the userIPAddress, but this is not always the case since a router in the data network might have translated the UserIPAddress to another IP-address. The latter is then contained in UserConnectionID.
AccessPointName	TpString	Provides the Access Point Name of the Internet Service Provider.

### 11.5.12 TpUserStatusExtendedSet

Defines a Numbered Set of Data Elements of TpUserStatusExtended.

### 11.5.13 TpUserStatusIndicatorExtended

Defines the status of a user.

Sequence Element Name	Sequence Element Type	Description
UserStatusIndicator	TpUserStatusIndicator	Indication whether the user is busy, reachable or not reachable
PDPContextActive	TpBoolean	Indication whether the user currently has an ongoing data session (i.e. PDP context established)

### 11.5.14 TpAuthStatusIndicator

Used to indicate if and how the user has been authenticated during data session establishment.

Name	Value	Description
P_AUTHENTICATED_FOR_NETWORK_ONLY	0	Authenticated only for the network
P_AUTHENTICATED_FOR_NETWORK_AND_IP_SERVICES	1	Authenticated for the network and IP services
P_NOT_AUTHENTICATED	2	Not authenticated

### 11.5.15 TpUserInfo

Used to reference the user name and password supplied by the user, if known by the network.

Sequence Element Name	Sequence Element Type	Description
UserName	TpString	User name as provided in a PPP/CHAP message.
Password	TpString	Password as provided in a PPP/CHAP message.

### 11.5.16 TpNetworkStatusIndicator

Specifies the country code, mobile network code, access network technology used by the user for the data session and roaming status.

Sequence Element Name	Sequence Element Type	Description
CountryCode	TpString	Network CC Identification, Country code.
MobileNetworkCode	TpString	Network MNC Identification, Mobile network code
AccessTechnology	TpAccessTechnology	Network access technology used by the user to connect
RoamingStatus	TpRoamingStatus	Roaming Status

### 11.5.17 TpAccessTechnology

Defines the derived access technology utilised by the user.

Name	Value	Description
P_MOBILE_ACCESS_PS	0	Access via packet switched connection in mobile network
P_MOBILE_ACCESS_CS	1	Access via a circuit switched connection in mobile network
P_FIXED_ACCESS	2	Access via a fixed network

### 11.5.18 TpRoamingStatus

Defines the roaming status of the user.

Name	Value	Description
P_UNKNOWN_ROAMING_STATUS	0	No information on Roaming status has been found
P_HOME_USER	1	User is in the home network
P_NATIONAL_ROAMING	2	User has roamed to another operator in the same country
P_INTERNATIONAL_ROAMING	3	User has roamed to another operator abroad
P_OPERATOR_ROAMING	4	User has roamed to a partner operator abroad

## 11.6 User Binding Data Definitions

### 11.6.1 TpBindingSet

Defines a Numbered Set of Data Elements of TpBindingEntrySet.

### 11.6.2 TpBindingEntrySet

Defines a Numbered Set of Data Elements of TpBindingEntry.

### 11.6.3 TpBindingEntry

Defines the Tagged Choice of Data Elements that specifies binding information.

Tag Element Type
TpBindingEntryType

Tag Element Value	Choice Element Type	Choice Element Name	Description
P_UB_EXPIRES	TpInt32	UBExpires	This field can be used to enforce time limits on session duration and is defined as an integral number of seconds.
P_UB_CONTACT_ADDRESS	TpAddress	UBContactAddress	This field informs the network of the subscriber's desired binding address.
P_UB_PREFERENCE	TpFloat	UBPreference	This field conveys a relative order among subscriber bindings.

### 11.6.4 TpBindingEntryType

Defines the type of binding information.

Name	Value	Description
P_UB_UNDEFINED	0	Undefined
P_UB_EXPIRES	1	A number indicating the number of seconds the binding is valid
P_UB_CONTACT_ADDRESS	2	The address representing the user's contact address
P_UB_PREFERENCE	3	A number between 0 and 1 indicating the contact address' relative preference

### 11.6.5 TpBindingNotificationCriteriaSet

Defines a Numbered Set of Data Elements of TpBindingNotificationCriteria.

### 11.6.6 TpBindingNotificationCriteria

Defines the Sequence of Data Elements that specifies the criteria relating to binding request notifications.

Sequence Element Name	Sequence Element Type
NotificationCriteriaType	TpBindingNotificationCriteriaType
ContactAddressSet	TpAddressSet

### 11.6.7 TpBindingNotificationCriteriaType

Defines the binding operations to be reported. The values may be combined by a logical 'OR' function.

Name	Value	Description
P_UB_NEW	1	Indicates attempt to create a new binding
P_UB_REMOVE	2	Indicates removal of an existing binding
P_UB_UPDATE	4	Indicates attempt to update an existing binding



## 11.7 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 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\_20391506v010101m0.zip) which accompanies the present document.

---

## Annex B (informative): W3C WSDL Description of Mobility SCF

The W3C WSDL representation of this interface specification is contained in a text file (mm.wsdl contained in archive es\_20391506v010101m0.zip) which accompanies the present document.

---

## Annex C (informative): Java API Description of the Mobility SCFs

The Java API realisation of this interface specification is produced in accordance with the Java Realisation rules defined in ES 203 915-1. These rules aim to deliver for Java, a developer API, provided as a realisation, supporting a Java API that represents the UML specifications. The rules support the production of both J2SE and J2EE versions of the API from the common UML specifications.

The J2SE representation of this interface specification is provided as Java Code, contained in archive 20391506J2SE.zip.

The J2EE representation of this interface specification is provided as Java Code, contained in archive 20391506J2EE.zip.

Both these archives can be found in es\_20391506v010101m0.zip which accompanies the present document.

---

## Annex D (informative): Contents of 3GPP OSA R6 Mobility

All parts of User Location, User Location Camel and User Status are relevant for TS 129 198-6 V6 (Release 6).

---

## Annex E (informative): Description of Mobility SCF for 3GPP2 cdma2000 networks

This annex is intended to define the OSA API Stage 3 interface definitions and it provides the complete OSA specifications. It is an extension of OSA API specifications capabilities to enable operation in cdma2000 systems environment. They are in alignment with 3GPP2 Stage 1 requirements and Stage 2 architecture defined in [52], [53] and [54] of ES 203 915-1, clause 2. These requirements are expressed as additions to and/or exclusions from the 3GPP Release 6 specification. The information given here is to be used by developers in 3GPP2 cdma2000 network architecture to interpret the 3GPP OSA specifications.

---

### E.1 General Exceptions

The terms 3GPP and UMTS are not applicable for the cdma2000 family of standards. Nevertheless these terms are used (TR 121 905) mostly in the broader sense of "3G Wireless System". If not stated otherwise there are no additions or exclusions required.

CAMEL and CAP mappings are not applicable for cdma2000 systems.

---

### E.2 Specific Exceptions

#### E.2.1 Clause 1: Scope

There are no additions or exclusions.

#### E.2.2 Clause 2: References

Normative references on TS 123 078 and on TS 129 078 are not applicable for cdma2000 systems.

#### E.2.3 Clause 3: Definitions and abbreviations

There are no additions or exclusions.

#### E.2.4 Clause 4: Mobility SCF

There are no additions or exclusions. UL-User Location UMTS term may be mapped onto LCS-Location Services cdma2000 term. CAMEL is not applicable for cdma2000 systems.

#### E.2.5 Clause 5: Sequence Diagrams

There are no additions. User Location Camel Sequence Diagrams are not applicable for cdma2000 systems.

#### E.2.6 Clause 6 Class Diagrams

There are no additions. User Location Camel Class Diagrams are not applicable for cdma2000 systems.

#### E.2.7 Clause 7: The Service Interface Specifications

There are no additions or exclusions.

## E.2.8 Clause 8: Mobility Interface Classes

There are no additions. User Location Camel Interface Classes are not applicable for cdma2000 systems.

## E.2.9 Clause 9: State Transition Diagrams

Since CAMEL protocol is not applicable for cdma2000 systems, User Location Camel state transition diagrams are not applicable for cdma2000 systems.

## E.2.10 Clause 10: Service Properties

Since CAMEL protocol is not applicable for cdma2000 systems, an SCS shall indicate support for the CAMEL feature through service properties. For cdma2000 systems the service properties shall be disabled (CAMEL shall be turned always off in the case of the 3GPP2 networks; e.g.: UserLocationCamel shall be set to false).

## E.2.11 Clause 11: Data Definitions

There are no additions. UserLocationCamel data definitions are not applicable for cdma2000 systems.

## E.2.12 Clause 12: Exception Classes

There are no additions or exclusions.

## E.2.13 Annex A (normative): OMG IDL Description of Mobility SCF

There are no additions or exclusions.

## E.2.14 Annex B (informative): W3C WSDL Description of Mobility SCF

There are no additions or exclusions.

## E.2.15 Annex C (informative): Java™ API Description of Mobility SCF

There are no additions or exclusions.



---

## Annex F (informative): Record of changes

The following is a list of the changes made to the present document for each release. The list contains the names of all changed, deprecated, added or removed items in the specifications and not the actual changes. Any type of change information that is important to the reader is put in the final clause of this annex.

Changes are specified as changes to the prior major release, but every minor release will have its own part of the table allowing the reader to know when the actual change was made.

---

### F.1 Interfaces

#### F.1.1 New

Identifier	Comments
<b>Interfaces added in ES 203 915-6 version 1.1.1 (Parlay 5.0)</b>	
IpUserBinding	
IpAppUserBinding	

#### F.1.2 Deprecated

Identifier	Comments
<b>Interfaces deprecated in ES 203 915-6 version 1.1.1 (Parlay 5.0)</b>	

#### F.1.3 Removed

Identifier	Comments
<b>Interfaces removed in ES 203 915-6 version 1.1.1 (Parlay 5.0)</b>	

## F.2 Methods

### F.2.1 New

Identifier	Comments
<b>Methods added in ES 203 915-6 version 1.1.1 (Parlay 5.0)</b>	
IpUserLocation.getNextPeriodicLocationRequest	
IpTriggeredUserLocation.getNextTriggeredLocationRequest	
IpUserLocationCamel.getNextPeriodicLocationRequest	
IpUserLocationCamel.getNextTriggeredLocationRequest	
IpUserStatus.getNextTriggeredStatusRequest	
IpAppUserStatus.extendedStatusReportRes	
IpAppUserStatus.extendedStatusReportErr	
IpAppUserStatus.extTriggeredStatusReport	
IpAppUserStatus.extTriggeredStatusReportErr	
IpUserStatus.extendedStatusReportReq	
IpUserStatus.extTriggeredStatusReportingStartReq	
IpUserStatus.extTriggeredStatusReportingStop	
IpUserBinding.triggeredBindingRequestNotificationStartReq()	
IpUserBinding.triggeredBindingRequestNotificationStop()	
IpAppUserBinding.triggeredBindingRequestNotification()	
IpAppUserBinding.triggeredBindingRequestNotificationStartErr()	

### F.2.2 Deprecated

Identifier	Comments
<b>Methods deprecated in ES 203 915-6 version 1.1.1 (Parlay 5.0)</b>	

### F.2.3 Modified

Identifier	Comments
<b>Methods modified in ES 203 915-6 version 1.1.1 (Parlay 5.0)</b>	

### F.2.4 Removed

Identifier	Comments
<b>Methods removed in ES 203 915-6 version 1.1.1 (Parlay 5.0)</b>	

## F.3 Data Definitions

### F.3.1 New

Identifier	Comments
<b>Data Definitions added in ES 203 915-6 version 1.1.1 (Parlay 5.0)</b>	
TpTriggeredLocationRequest	
TpTriggeredLocationRequestSet	
TpTriggeredLocationRequestSetEntry	
TpPeriodicLocationRequest	
TpPeriodicLocationRequestSet	
TpPeriodicLocationRequestSetEntry	
TpTriggeredStatusRequest	
TpTriggeredStatusRequestSet	
TpTriggeredStatusRequestSetEntry	
TpUserStatusExtended	
TpUserStatusExtendedSet	
TpUserStatusIndicatorExtended	
TpAuthStatusIndicator	
TpUserInfo	
TpNetworkStatusIndicator	
TpAccessTechnology	
TpRoamingStatus	
TpBindingSet	
TpBindingEntrySet	
TpBindingEntry	
TpBindingEntryType	
TpBindingNotificationCriteriaSet	
TpBindingNotificationCriteria	
TpBindingNotificationCriteriaType	

### F.3.2 Modified

Identifier	Comments
<b>Data Definitions modified in ES 203 915-6 version 1.1.1 (Parlay 5.0)</b>	

### F.3.3 Removed

Identifier	Comments
<b>Data Definitions removed in ES 203 915-6 version 1.1.1 (Parlay 5.0)</b>	

---

## F.4 Service Properties

### F.4.1 New

Identifier	Comments
Service Properties added in ES 203 915-6 version 1.1.1 (Parlay 5.0)	

### F.4.2 Deprecated

Identifier	Comments
Service Properties deprecated in ES 203 915-6 version 1.1.1 (Parlay 5.0)	

### F.4.3 Modified

Identifier	Comments
Service Properties modified in ES 203 915-6 version 1.1.1 (Parlay 5.0)	

### F.4.4 Removed

Identifier	Comments
Service Properties removed in ES 203 915-6 version 1.1.1 (Parlay 5.0)	

---

## F.5 Exceptions

### F.5.1 New

Identifier	Comments
Exceptions added in ES 203 915-6 version 1.1.1 (Parlay 5.0)	

### F.5.2 Modified

Identifier	Comments
Exceptions modified in ES 203 915-6 version 1.1.1 (Parlay 5.0)	

### F.5.3 Removed

Identifier	Comments
Exceptions removed in ES 203 915-6 version 1.1.1 (Parlay 5.0)	

---

### F.6 Others

Java code added.

---

## History

<b>Document history</b>		
V1.1.1	February 2005	Membership Approval Procedure MV 20050408: 2005-02-08 to 2005-04-08