3GPP TSG CN Plenary Meeting #20 **NP-030249**
04-06 June 2003. Hämeenlinna, FINLAND

Source: **CN5 (OSA)**

Title: **Rel-6 CR 29.198-06 OSA API Part 6: Mobility**

Agenda item: **9.7**

Document for: **APPROVAL**

Doc-1st-Level	Spec	CR	R	Ph	Subject	Cat	Ver-Curr	Doc-2nd-Level	WI
NP-030249	29.198-06	021	-	Rel-6	Add methods to prevent unnecessary strict requirements on application persistency for Mobility	C	5.2.0	N5-021036	OSA3
NP-030249	29.198-06	022	-	Rel-6	Extension to User Status	B	5.2.0	N5-030278	OSA3

joint API group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) **N5-021036**
Meeting #21, Dublin, IRELAND, 28 – 31 Oct 2002

CR-Form-v5

CHANGE REQUEST

⌘ **29.198-06** **CR** **021** ⌘**rev** **-** ⌘ Current version: **5.2.0** ⌘

For ***HELP*** *on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM □ ME/UE □ Radio Access Network □ Core Network ☒

Title:	⌘	Add methods to prevent unnecessary strict requirements on application persistency for Mobility		
Source:	⌘	Ericsson		
Work item code:	⌘	OSA3	***Date:*** ⌘	15/10/2002
Category:	⌘	**C**	***Release:*** ⌘	*REL-6*

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change:	⌘	Consider an application that has started several triggered status requests and that this application crashes and restarts. If the application would not consider the requests it had started before the application crash, and instead would simply restart them, the result would be that the 'old' (inactive) requests remain the Parlay gateway. The Parlay gateway does not have an infallible means of judging which requests are old and which are new. As a result, all old trigger requests would accumulate in the Parlay gateway. Eventually this would backfire to the application when system or service level agreement limits are reached. To prevent this scenario, an application must currently be persistent with regards to all requests it has started. This applies not only to triggered user status requests, but also to triggered and periodic user location requests. This CR proposes to modify the specification in order to avoid this need for persistency.
Summary of change:	⌘	Provide a means to applications to obtain information from the Parlay gateway regarding all mobility requests they have started
Consequences if not approved:	⌘	High persistency requirements on applications, thus increasing the threshold to create Parlay applications.

Clauses affected:	⌘		
Other specs affected:	⌘	□ Other core specifications □ Test specifications □ O&M Specifications	⌘
Other comments:	⌘		

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

Proposed Changes

8.1.1 Interface Class IpUserLocation

Inherits from: IpService.

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

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

<<Interface>>

IpUserLocation

locationReportReq (appLocation : in IpAppUserLocationRef, users : in TpAddressSet) : 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

getNextPeriodicLocationRequest(reset: in TpBoolean): TpPeriodicLocationRequestSetEntry

8.1.1.5 Method 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

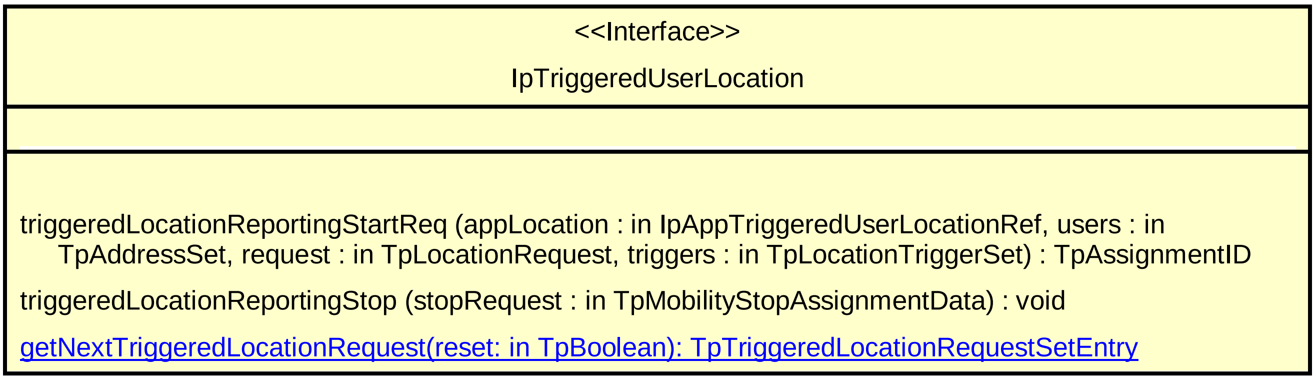
TpCommonExceptions

8.1.3 Interface Class IpTriggeredUserLocation

Inherits from: IpUserLocation.

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

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



<<Interface>>

IpTriggeredUserLocation

triggeredLocationReportingStartReq (appLocation : in IpAppTriggeredUserLocationRef, users : in TpAddressSet, request : in TpLocationRequest, triggers : in TpLocationTriggerSet) : TpAssignmentID

triggeredLocationReportingStop (stopRequest : in TpMobilityStopAssignmentData) : void

getNextTriggeredLocationRequest(reset: in TpBoolean): TpTriggeredLocationRequestSetEntry

8.1.3.3 Method getNextTriggeredLocationRequest()

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 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`

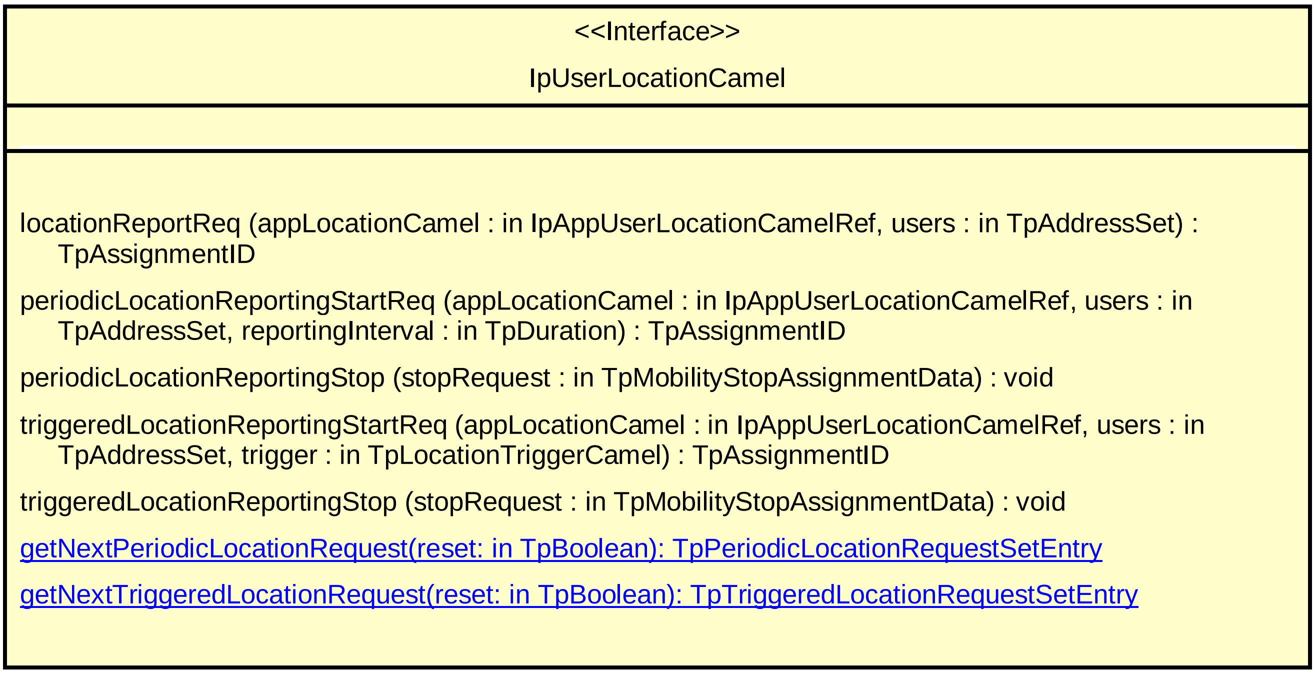
Raises

`TpCommonExceptions`

8.2.1 Interface Class IpUserLocationCamel

Inherits from: IpService.

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



<<Interface>>

IpUserLocationCamel

locationReportReq (appLocationCamel : in IpAppUserLocationCamelRef, users : in TpAddressSet) : TpAssignmentID

periodicLocationReportingStartReq (appLocationCamel : in IpAppUserLocationCamelRef, users : in TpAddressSet, reportingInterval : in TpDuration) : TpAssignmentID

periodicLocationReportingStop (stopRequest : in TpMobilityStopAssignmentData) : void

triggeredLocationReportingStartReq (appLocationCamel : in IpAppUserLocationCamelRef, users : in TpAddressSet, trigger : in TpLocationTriggerCamel) : TpAssignmentID

triggeredLocationReportingStop (stopRequest : in TpMobilityStopAssignmentData) : void

getNextPeriodicLocationRequest(reset: in TpBoolean): TpPeriodicLocationRequestSetEntry

getNextTriggeredLocationRequest(reset: in TpBoolean): TpTriggeredLocationRequestSetEntry

8.2.1.6 Method 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 getNextTriggeredLocationRequest()

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 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`

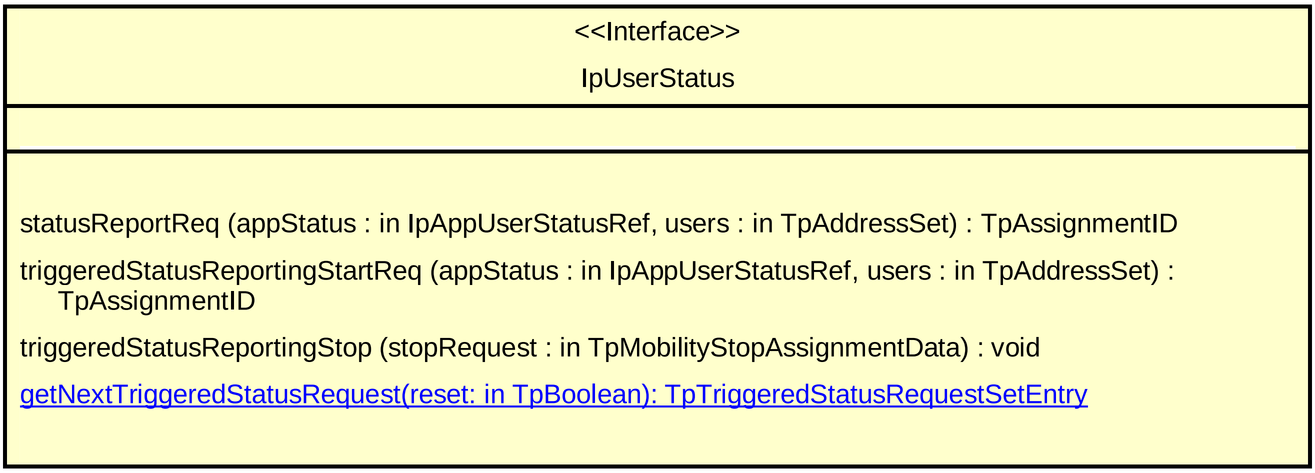
Raises

`TpCommonExceptions`

8.3.2 Interface Class IpUserStatus

Inherits from: IpService.

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



<<Interface>>

IpUserStatus

statusReportReq (appStatus : in IpAppUserStatusRef, users : in TpAddressSet) : TpAssignmentID

triggeredStatusReportingStartReq (appStatus : in IpAppUserStatusRef, users : in TpAddressSet) : TpAssignmentID

triggeredStatusReportingStop (stopRequest : in TpMobilityStopAssignmentData) : void

getNextTriggeredStatusRequest(reset: in TpBoolean): TpTriggeredStatusRequestSetEntry

8.2.1.4 Method 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

`TpTriggeredRequestSetEntry`

Raises

`TpCommonExceptions`

11.5.8 TpTriggeredRequestSetEntry

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	TpInt32

11.5.11 TpTriggeredLocationSetEntry

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).

11.5.12 TpTriggeredLocationSet

Defines a numbered Set of Data Elements of TpTriggeredLocationRequest.

11.5.13 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	TpInt32

11.5.14 TpTPeriodicLocationSetEntry

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).

11.5.15 TpPeriodicLocationSet

Defines a numbered Set of Data Elements of TpPeriodicLocationRequest.

11.5.16 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	TpInt32

joint-API-group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) N5-030278
Meeting #23, San Diego, CA, USA, 19 - 23 May 2003

CR-Form-v7

CHANGE REQUEST

⌘	**29.198-06**	**CR**	**022**	⌘ **rev**	**-**	⌘	Current version:	**5.2.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network ☒

Title: ⌘	Extension to User Status		
Source: ⌘	Ericsson (Erwin van Rijssen, Erwin.van.Rijssen@etm.ericsson.se)		
Work item code: ⌘	OSA3	***Date:*** ⌘	05/05/2003
Category: ⌘	**B**	***Release:*** ⌘	*REL-6*
	Use one of the following categories: ***F*** *(correction)* ***A*** *(corresponds to a correction in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)* Detailed explanations of the above categories can be found in 3GPP TR 21.900.		*Use one of the following releases:* *2 (GSM Phase 2)* *R96 (Release 1996)* *R97 (Release 1997)* *R98 (Release 1998)* *R99 (Release 1999)* *Rel-4 (Release 4)* *Rel-5 (Release 5)* *Rel-6 (Release 6)*

Reason for change: ⌘	Make the User Status service applicable not only in a telephony network but also in a data network
Summary of change: ⌘	Expand the User Status service to enable applications to determine the user's availability also by checking with data network
Consequences if not approved: ⌘	User Status service will be limited for use to telephony networks

Clauses affected: ⌘	6.3, 8.3, 11.5				
	Y	**N**			
Other specs Affected: ⌘		**X**	Other core specifications	⌘	
		X	Test specifications		
		X	O&M Specifications		
Other comments: ⌘					

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

Introduction

Specification 29.198-06 defines a User Status service that enables applications to obtain the user's status in the telephony network (e.g. busy, not reachable etc.). One way to determine the user's status (that is not covered by the User Status service as currently defined) is to verify if the user got an IP-address allocated. In that case there is a data session (e.g. PDP context or CSD connection) established between the user's terminal and a data network. This information is an indication to the application about the user's availability/status.

Ericsson therefore proposes to extend the User Status service, to enable applications to find out the user's status by checking if the user has currently an ongoing data session (i.e. whether the user's terminal has been assigned an IP-address).

In the proposal we extend the User Status data definitions with User Status Extended data definitions, in the same way User Location was extended with User Location Extended. In this way we are consistent with the principles followed so far in the OSA specificationa and we ensure backwards compatibility of the User Status service.

This contribution only shows the proposed extensions to the existing OSA Mobility (29.198-6 v5.2.0) specification.

Proposal

Include the extensions proposed in this contribution in TS29.198-6 (Mobility) for 3GPP R6

Proposed Changes

6.3 User Status Class Diagrams

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

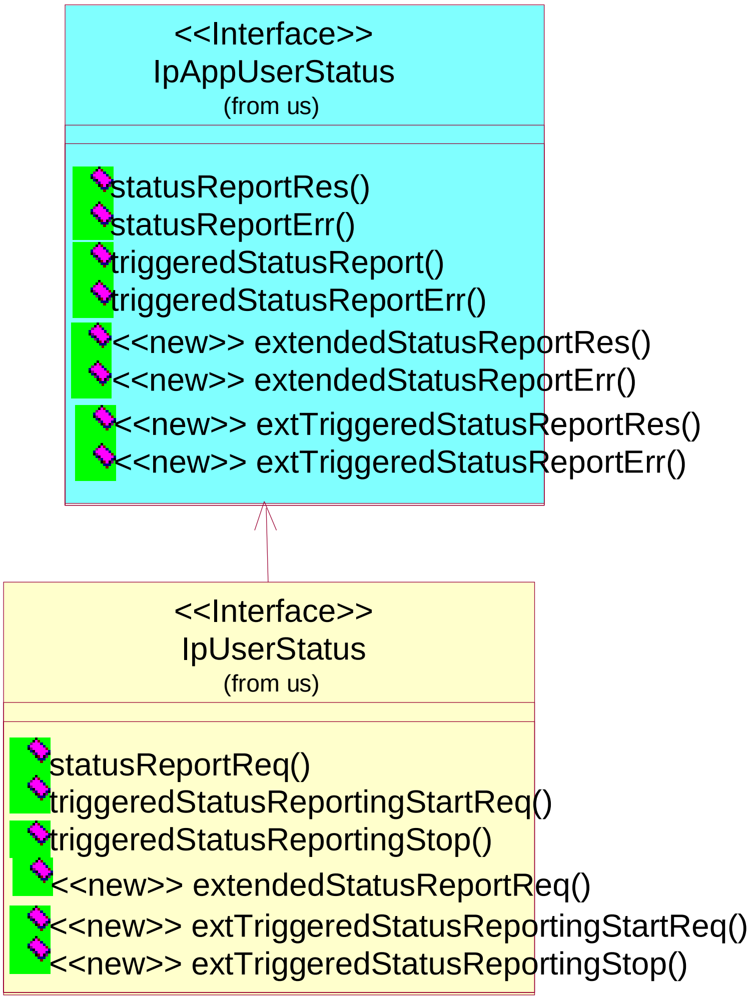

<<Interface>>
IpAppUserStatus
(from us)
statusReportRes()
statusReportErr()
triggeredStatusReport()
triggeredStatusReportErr()
<<new>> extendedStatusReportRes()
<<new>> extendedStatusReportErr()
<<new>> extTriggeredStatusReportRes()
<<new>> extTriggeredStatusReportErr()
<<Interface>>
IpUserStatus
(from us)
statusReportReq()
triggeredStatusReportingStartReq()
triggeredStatusReportingStop()
<<new>> extendedStatusReportReq()
<<new>> extTriggeredStatusReportingStartReq()
<<new>> extTriggeredStatusReportingStop()

Figure: User Status Class Diagram

8.3.1 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.

<<Interface>>

IpAppUserStatus

statusReportRes (assignmentId : in TpAssignmentID, status : in TpUserStatusSet) : void

statusReportErr (assignmentId : in TpAssignmentID, cause : in TpMobilityError, diagnostic : in TpMobilityDiagnostic) : void

triggeredStatusReport (assignmentId : in TpAssignmentID, status : in TpUserStatus) : void

triggeredStatusReportErr (assignmentId : in TpAssignmentID, cause : in TpMobilityError, diagnostic : in TpMobilityDiagnostic) : void

<<new>> extendedStatusReportRes (assignmentId : in TpAssignmentID, status : in TpUserStatusExtendedSet) : void

<<new>> extendedStatusReportErr (assignmentId : in TpAssignmentID, cause : in TpMobilityError, diagnostic : in TpMobilityDiagnostic) : void

<<new>> extTriggeredStatusReport (assignmentId : in TpAssignmentID, status : in TpUserStatusExtended) : void

<<new>> extTriggeredStatusReportErr (assignmentId : in TpAssignmentID, cause : in TpMobilityError, diagnostic : in TpMobilityDiagnostic) : void

8.3.1.5 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.

8.3.1.6 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.

8.3.1.7 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.

8.3.1.8 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.

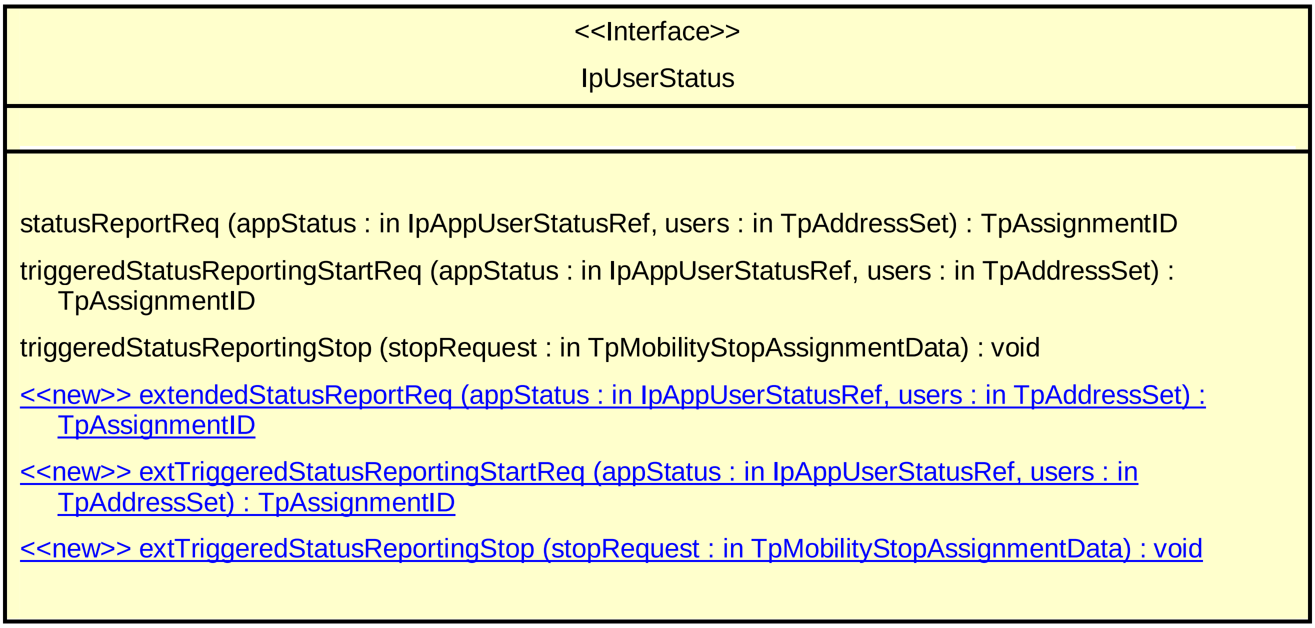
8.3.2 Interface Class IpUserStatus

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 trigggeredStatusReportingStop() methods shall be implemented as a minimum requirement.



<<Interface>>

IpUserStatus

statusReportReq (appStatus : in IpAppUserStatusRef, users : in TpAddressSet) : TpAssignmentID

triggeredStatusReportingStartReq (appStatus : in IpAppUserStatusRef, users : in TpAddressSet) : TpAssignmentID

triggeredStatusReportingStop (stopRequest : in TpMobilityStopAssignmentData) : void

<<new>> extendedStatusReportReq (appStatus : in IpAppUserStatusRef, users : in TpAddressSet) : TpAssignmentID

<<new>> extTriggeredStatusReportingStartReq (appStatus : in IpAppUserStatusRef, users : in TpAddressSet) : TpAssignmentID

<<new>> extTriggeredStatusReportingStop (stopRequest : in TpMobilityStopAssignmentData) : void

8.3.2.4 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

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

8.3.2.5 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`

8.3.2.6 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

11.5.8 <<new>> 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 IPAddress allocated then this is represented with the empty string.
UserMSISDN	TpString	The user MDISDN. 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

11.5.9 <<new>> TpUserStatusExtendedSet

Defines a Numbered Set of Data Elements of TpUserStatusExtended.

11.5.10 <<new>> TpUserStatusIndicatorExtended

Defines the status of a user.

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

11.5.11 <<new>> 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

11.5.12 <<new>> TpUserInfo

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

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

11.5.13 <<new>> TpNetworkStatusIndicator

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

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

11.5.14 <<new>> 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_CSD	1	Access via a circuit switched connection in mobile network
P_FIXED_ACCESS	2	Access via a fixed network

11.5.15 <<new>> TpExtendedRoamingStatus

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