

**3GPP TSG CN Plenary Meeting #14
Kyoto, Japan, 12-14 December 2001**

NP-010602

Source: CN5 (OSA)

Title: Rel-4 CRs 29.198-11

Agenda item: 8.5

Document for: Decision

Doc-1st-Level	Spec	CR	Pha	Subject	Cat	Ver Cur	Ver -New	Doc-2nd-Level	Workitem
NP-010602	29.198-11	003	Rel-4	Replace Out Parameters with Return Types	F	4.1.0	4.2.0	N5-010569	OSA1
NP-010602	29.198-11	004	Rel-4	Replace erroneous use of incorrect data type TpSessionID by TpAssignmentID in Account Management interface	F	4.1.0	4.2.0	N5-010802	OSA1

CHANGE REQUEST

⌘ **29.198-11 CR 003** ⌘ ev **-** ⌘ Current version: **4.1.0** ⌘

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:	⌘ Replacing Out Parameters with Return Types in OSA R4		
Source:	⌘ CN5		
Work item code:	⌘ OSA1	Date:	⌘ 19/072001
Category:	⌘ F	Release:	⌘ REL-4
	Use <u>one</u> 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 <u>one</u> 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:	⌘ At CN5 and CN it was agreed that Out-parameters should be removed from methods as a means of returning information, to be replaced by Return Types, in line with commonly used programming practice
Summary of change:	⌘ For each method, replace the return parameter TpResult with: 'void' if the method has no out-parameter; or the type of the out-parameter if the method has an out-parameter, and delete the out-parameter from the method.
Consequences if not approved:	⌘ If this particular CR is not agreed, TS 29.198-11 is out of sync. with the other parts of TS 29.198. If the related batch of CRs is not agreed, OSA will have a limited acceptance among the application development community, since it will be more difficult to implement. This presents a risk to the return on investment in development of OSA.

Clauses affected:	⌘ 7, 8		
Other specs affected:	⌘ <input checked="" type="checkbox"/> Other core specifications <input type="checkbox"/> Test specifications <input type="checkbox"/> O&M Specifications	⌘	All other parts of TS 29.198 Rel-4
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.

7 The Service Interface Specifications

7.1 Interface Specification Format

This section 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. All methods in the API return a value of type `TpResult`, indicating, amongst other things, if the method invocation was successfully executed or not.

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.

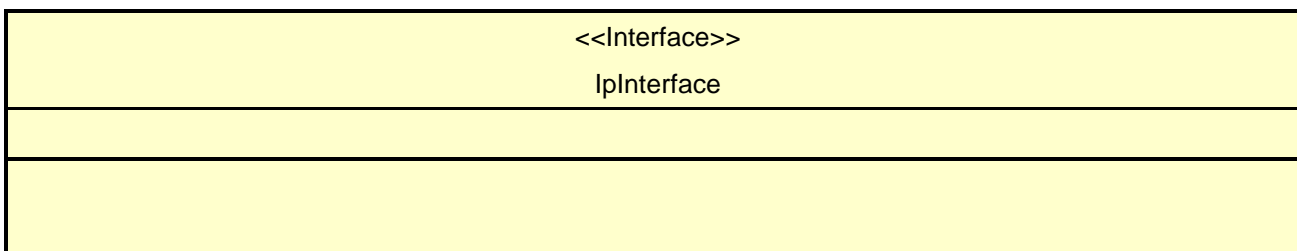
7.1.4 State Model

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.

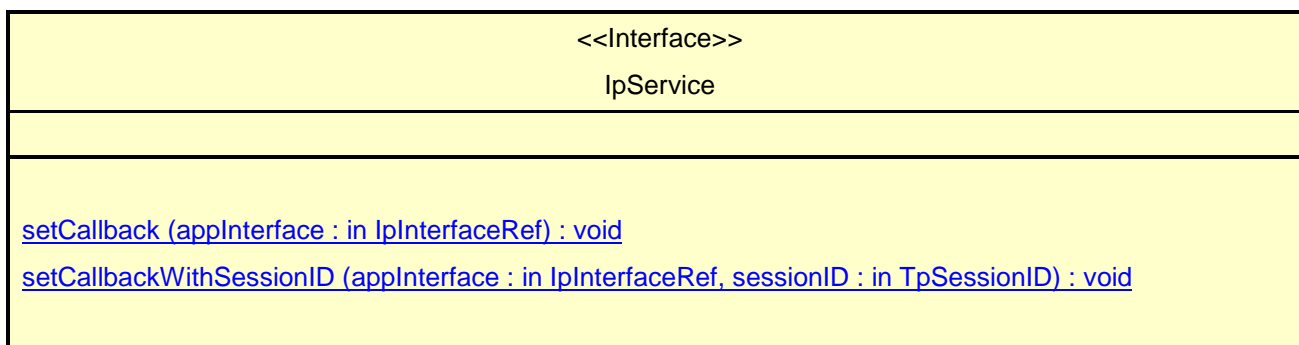
The interfaces that are implemented by the services are denoted as 'Service Interface'. The corresponding interfaces that must be implemented by the application (e.g. for API callbacks) are denoted as 'Application Interface'.

7.4 Generic Service Interface

7.4.1 Interface Class IpService

Inherits from: IpInterface

All service interfaces inherit from the following interface.



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 SessionID's.

Parameters

appInterface : in IpInterfaceRef

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

Raises

TpCommonExceptions

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 uses SessionID's.

*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.

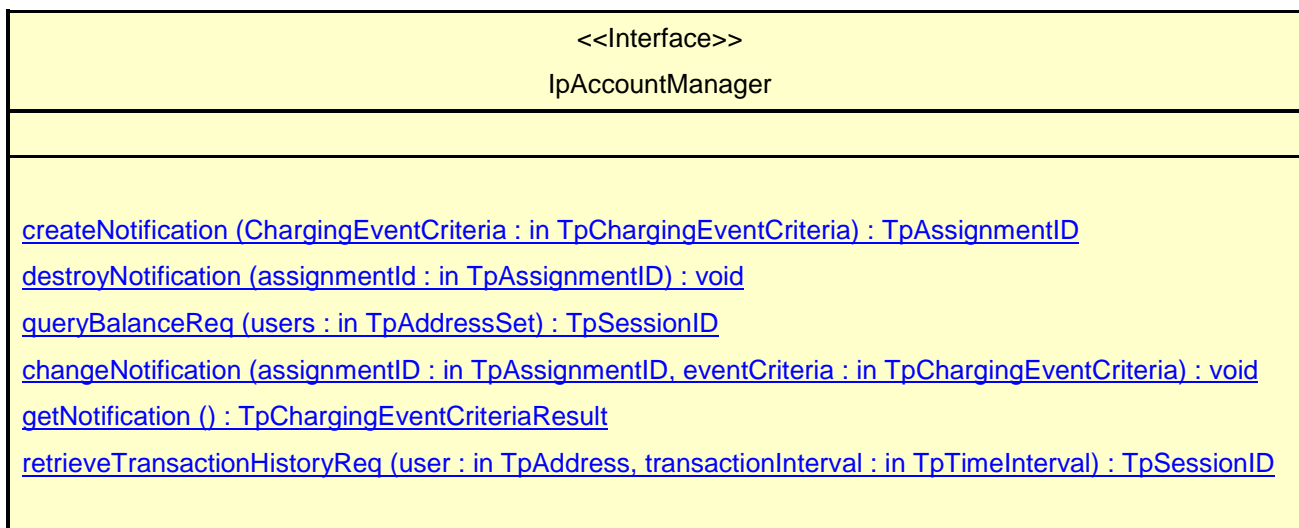
*Raises***TpCommonExceptions, P_INVALID_SESSION_ID**

8 Account Management Interface Classes

8.1 Interface Class IpAccountManager

Inherits from: IpService.

The account manager interface provides methods for monitoring accounts. Applications can use this interface to enable or disable charging-related event notifications and to query account balances.

*Method***createNotification()**[This method is used by the application to enable charging event notifications to be sent to the application.](#)[Returns assignmentId : Specifies the ID assigned by the account management object for this newly enabled event notification.](#)*Parameters***ChargingEventCriteria : in TpChargingEventCriteria**

Specifies the event specific criteria used by the application to define the charging event required. Individual addresses or address ranges may be specified for subscriber accounts. Example of events are "charging" and "recharging".

Returns

[TpAssignmentID](#)

Raises

TpCommonExceptions, P_INVALID_CRITERIA, P_INVALID_EVENT_TYPE, P_UNKNOWN_SUBSCRIBER

*Method***destroyNotification()**

This method is used by the application to disable charging notifications.

Parameters

assignmentId : in TpAssignmentID

Specifies the assignment ID that was given by the account management object when the application enabled the charging notification.

Raises

TpCommonExceptions, P_INVALID_ASSIGNMENT_ID

*Method***queryBalanceReq()**

[This method is used by the application to query the balance of an account for one or several users.](#)

[Returns queryId : Specifies the ID of the balance query request.](#)

Parameters

users : in TpAddressSet

Specifies the user(s) for which the balance is queried.

Returns

[TpSessionID](#)

Raises

TpCommonExceptions, P_UNKNOWN_SUBSCRIBER, P_UNAUTHORIZED_APPLICATION

*Method***changeNotification()**

This method is used by the application to change the event criteria introduced with createNotification. Any stored criteria associated with the specified assignmentID will be replaced with the specified criteria.

*Parameters***assignmentID : in TpAssignmentID**

Specifies the ID assigned by the manager interface for the event notification.

eventCriteria : in TpChargingEventCriteria

Specifies the new set of event criteria used by the application to define the event required. Only events that meet these criteria are reported

*Raises***TpCommonExceptions, P_INVALID_ASSIGNMENT_ID, P_INVALID_CRITERIA,
P_INVALID_EVENT_TYPE***Method***getNotification()**[This method is used by the application to query the event criteria set with createNotification or changeNotification.](#)[Returns eventCriteria : Specifies the event criteria used by the application to define the event required. Only events that meet these criteria are reported.](#)*Parameters*

No Parameters were identified for this method

Returns[TpChargingEventCriteriaResult](#)*Raises***TpCommonExceptions***Method***retrieveTransactionHistoryReq()**[This asynchronous method is used by the application to retrieve a transaction history of a subscriber's account. The history is a set of Detailed Records.](#)[Returns retrievalID : Specifies the retrieval ID of the transaction history retrieval request.](#)*Parameters***user : in TpAddress**

Specifies the subscriber for whose account the transaction history is to be retrieved.

transactionInterval : in TpTimeInterval

Specifies the time interval for which the application history is to be retrieved.

Returns

[TpSessionID](#)

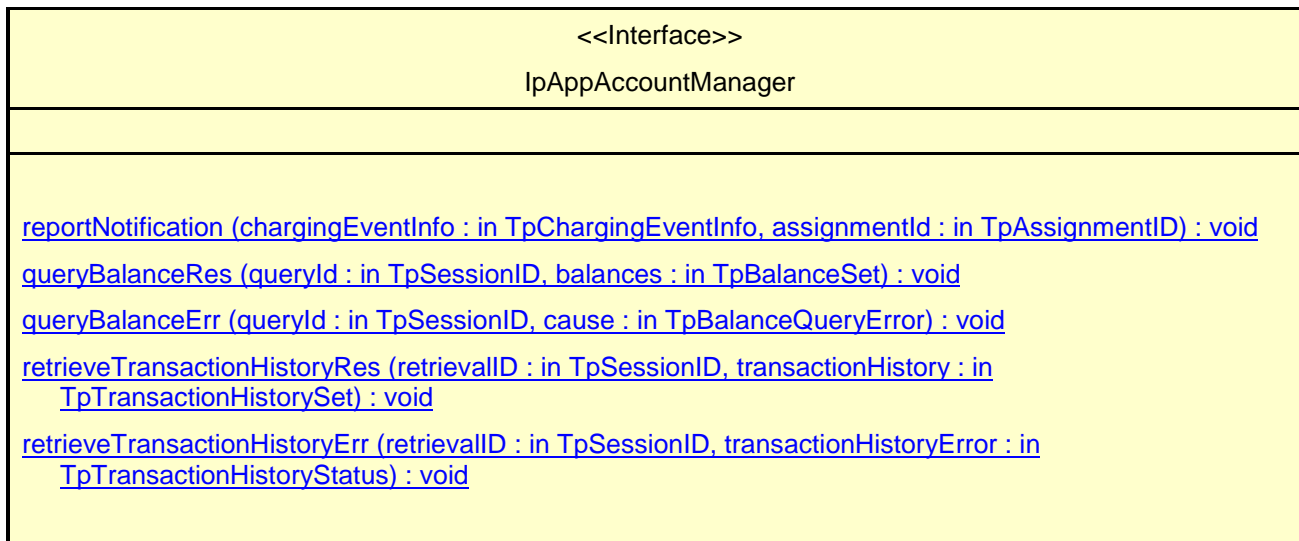
Raises

TpCommonExceptions, P_UNKNOWN_SUBSCRIBER, P_UNAUTHORIZED_APPLICATION, P_INVALID_TIME_AND_DATE_FORMAT

8.2 Interface Class IpAppAccountManager

Inherits from: IpInterface.

The account manager application interface is implemented by the client application developer and is used to handle charging event notifications and query balance responses.



Method

reportNotification()

This method is used to notify the application of a charging event.

Parameters

chargingEventInfo : in TpChargingEventInfo

Specifies data associated with this charging event. These data include the charging event being notified, the current value of the balance after the notified event occurred, and the time at which the charging event occurred.

assignmentId : in TpAssignmentID

Specifies the assignment ID that was returned by the createNotification() method. The application can use the assignment ID to associate events with event-specific criteria and to act accordingly.

Method

queryBalanceRes()

This method indicates that the request to query the balance was successful and it reports the requested balance of an account to the application.

Parameters

queryId : in TpSessionID

Specifies the ID of the balance query request.

balances : in TpBalanceSet

Specifies the balance for one or more user accounts.

Method

queryBalanceErr()

This method indicates that the request to query the balance failed and it reports the cause of failure to the application.

Parameters

queryId : in TpSessionID

Specifies the ID of the balance query request.

cause : in TpBalanceQueryError

Specifies the error that led to the failure.

Method

retrieveTransactionHistoryRes()

This method indicates that the request to retrieve the transaction history was successful and it returns the requested transaction history.

Parameters

retrievalID : in TpSessionID

Specifies the retrievalID of the transaction history retrieval request.

transactionHistory : in TpTransactionHistorySet

Specifies the requested transaction history.

Method

retrieveTransactionHistoryErr()

This method indicates that the request to retrieve the transaction history failed and it reports the cause of failure to the application.

Parameters

retrievalID : in TpSessionID

Specifies the retrievalID of the transaction history retrieval request.

transactionHistoryError : in TpTransactionHistoryStatus

Specifies the error that occurred while retrieving the transaction history.

Error! No text of specified style in document.

10

Error! No text of specified style in document.

CR-Form-v4

CHANGE REQUEST

⌘ **29.198-11 CR 004** ⌘ ev **-** ⌘ Current version: **4.1.0** ⌘

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:	⌘	Replace erroneous use of incorrect data type TpSessionID by TpAssignmentID in Account Management interface	
Source:	⌘	CN5 ¹	
Work item code:	⌘	OSA1	Date: ⌘ 13/09/2001
Category:	⌘	F	Release: ⌘ REL-4
		Use <u>one</u> 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 <u>one</u> 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:	⌘	The session ID parameter defined in the Account Management interface is used to relate responses to their corresponding requests. Currently the data type TpSessionID is being used for these parameters. However, according to section 5.1.15 in 3G TS 29.198-02 v4.1.0, this data type is used by the API to identify sessions within an object implementing an interface capable of handling multiple sessions. So the current use of the TpSessionID data type for the session ID parameters used in the Account Management interface is incorrect. Rather the data type TpAssignmentID should be used.
Summary of change:	⌘	This CR corrects the data type for session ID parameters in the following methods: <ul style="list-style-type: none"> – queryBalanceReq – queryBalanceRes – queryBalanceErr – retrieveTransactionHistoryReq – retrieveTransactionHistoryRes – retrieveTransactionHistoryErr
Consequences if not approved:	⌘	Incorrect specification. Failure to adopt this CR would result in divergence between the 3GPP R4 specification and the ETSI/Parlay specifications.

Clauses affected:	⌘	8.1, 8.2, 9.1
Other specs affected:	⌘	<input type="checkbox"/> Other core specifications ⌘ <input type="checkbox"/> Test specifications

¹ Contact: Musa Unmehopa (Lucent Technologies) unmehopa@lucent.com

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.

8 Account Management Interface Classes

8.1 Interface Class IpAccountManager

Inherits from: IpService.

The account manager interface provides methods for monitoring accounts. Applications can use this interface to enable or disable charging-related event notifications and to query account balances.

<<Interface>> IpAccountManager
<pre> createNotification (ChargingEventCriteria : in TpChargingEventCriteria, assignmentId : out TpAssignmentIDRef) : TpResult destroyNotification (assignmentId : in TpAssignmentID) : TpResult queryBalanceReq (users : in TpAddressSet, queryId : out TpSessionIDRefTpAssignmentID) : TpResult changeNotification (assignmentId : in TpAssignmentID, eventCriteria : in TpChargingEventCriteria) : TpResult getNotification (eventCriteria : out TpChargingEventCriteriaResultRef) : TpResult retrieveTransactionHistoryReq (user : in TpAddress, transactionInterval : in TpTimeInterval, retrievalID : out TpSessionIDRefTpAssignmentID) : TpResult </pre>

Method

createNotification()

This method is used by the application to enable charging event notifications to be sent to the application.

Parameters

ChargingEventCriteria : in TpChargingEventCriteria

Specifies the event specific criteria used by the application to define the charging event required. Individual addresses or address ranges may be specified for subscriber accounts. Example of events are "charging" and "recharging".

assignmentId : out TpAssignmentIDRef

Specifies the ID assigned by the account management object for this newly enabled event notification.

Raises

TpCommonExceptions, P_INVALID_CRITERIA, P_INVALID_EVENT_TYPE, P_UNKNOWN_SUBSCRIBER

*Method***destroyNotification()**

This method is used by the application to disable charging notifications.

Parameters

assignmentId : in TpAssignmentID

Specifies the assignment ID that was given by the account management object when the application enabled the charging notification.

Raises

TpCommonExceptions, P_INVALID_ASSIGNMENT_ID

*Method***queryBalanceReq()**

This method is used by the application to query the balance of an account for one or several users.

Parameters

users : in TpAddressSet

Specifies the user(s) for which the balance is queried.

queryId : out TpSessionIDRefTpAssignmentID

Specifies the ID of the balance query request.

Raises

TpCommonExceptions, P_UNKNOWN_SUBSCRIBER, P_UNAUTHORIZED_APPLICATION

*Method***changeNotification()**

This method is used by the application to change the event criteria introduced with createNotification. Any stored criteria associated with the specified assignmentID will be replaced with the specified criteria.

Parameters

assignmentID : in TpAssignmentID

Specifies the ID assigned by the manager interface for the event notification.

eventCriteria : in TpChargingEventCriteria

Specifies the new set of event criteria used by the application to define the event required. Only events that meet these criteria are reported

Raises

TpCommonExceptions, P_INVALID_ASSIGNMENT_ID, P_INVALID_CRITERIA, P_INVALID_EVENT_TYPE

*Method***getNotification()**

This method is used by the application to query the event criteria set with createNotification or changeNotification.

Parameters

eventCriteria : out TpChargingEventCriteriaResultRef

Specifies the event criteria used by the application to define the event required. Only events that meet these criteria are reported.

Raises

TpCommonExceptions

*Method***retrieveTransactionHistoryReq()**

This asynchronous method is used by the application to retrieve a transaction history of a subscriber's account. The history is a set of Detailed Records.

Parameters

user : in TpAddress

Specifies the subscriber for whose account the transaction history is to be retrieved.

transactionInterval : in TpTimeInterval

Specifies the time interval for which the application history is to be retrieved.

retrievalID : out TpSessionIDRefTpAssignmentID

Specifies the retrieval ID of the transaction history retrieval request.

Raises

TpCommonExceptions, P_UNKNOWN_SUBSCRIBER, P_UNAUTHORIZED_APPLICATION, P_INVALID_TIME_AND_DATE_FORMAT

8.2 Interface Class IpAppAccountManager

Inherits from: IpInterface.

The account manager application interface is implemented by the client application developer and is used to handle charging event notifications and query balance responses.

<<Interface>> IpAppAccountManager
reportNotification (chargingEventInfo : in TpChargingEventInfo, assignmentId : in TpAssignmentID) :

TpResult

```

queryBalanceRes (queryId : in TpSessionIDTpAssignmentID, balances : in TpBalanceSet) : TpResult
queryBalanceErr (queryId : in TpSessionIDTpAssignmentID, cause : in TpBalanceQueryError) : TpResult
retrieveTransactionHistoryRes (retrievalID : in TpSessionIDTpAssignmentID, transactionHistory : in
    TpTransactionHistorySet) : TpResult
retrieveTransactionHistoryErr (retrievalID : in TpSessionIDTpAssignmentID, transactionHistoryError : in
    TpTransactionHistoryStatus) : TpResult

```

*Method***reportNotification()**

This method is used to notify the application of a charging event.

*Parameters***chargingEventInfo : in TpChargingEventInfo**

Specifies data associated with this charging event. These data include the charging event being notified, the current value of the balance after the notified event occurred, and the time at which the charging event occurred.

assignmentId : in TpAssignmentID

Specifies the assignment ID that was returned by the createNotification() method. The application can use the assignment ID to associate events with event-specific criteria and to act accordingly.

*Method***queryBalanceRes()**

This method indicates that the request to query the balance was successful and it reports the requested balance of an account to the application.

*Parameters***queryId : in TpSessionIDTpAssignmentID**

Specifies the ID of the balance query request.

balances : in TpBalanceSet

Specifies the balance for one or more user accounts.

*Method***queryBalanceErr()**

This method indicates that the request to query the balance failed and it reports the cause of failure to the application.

*Parameters***queryId : in ~~TpSessionID~~TpAssignmentID**

Specifies the ID of the balance query request.

cause : in TpBalanceQueryError

Specifies the error that led to the failure.

*Method***retrieveTransactionHistoryRes()**

This method indicates that the request to retrieve the transaction history was successful and it returns the requested transaction history.

*Parameters***retrievalID : in ~~TpSessionID~~TpAssignmentID**

Specifies the retrievalID of the transaction history retrieval request.

transactionHistory : in TpTransactionHistorySet

Specifies the requested transaction history.

*Method***retrieveTransactionHistoryErr()**

This method indicates that the request to retrieve the transaction history failed and it reports the cause of failure to the application.

*Parameters***retrievalID : in ~~TpSessionID~~TpAssignmentID**

Specifies the retrievalID of the transaction history retrieval request.

transactionHistoryError : in TpTransactionHistoryStatus

Specifies the error that occurred while retrieving the transaction history.

9 State Transition Diagrams

9.1 State Transition Diagrams for IpAccountManager

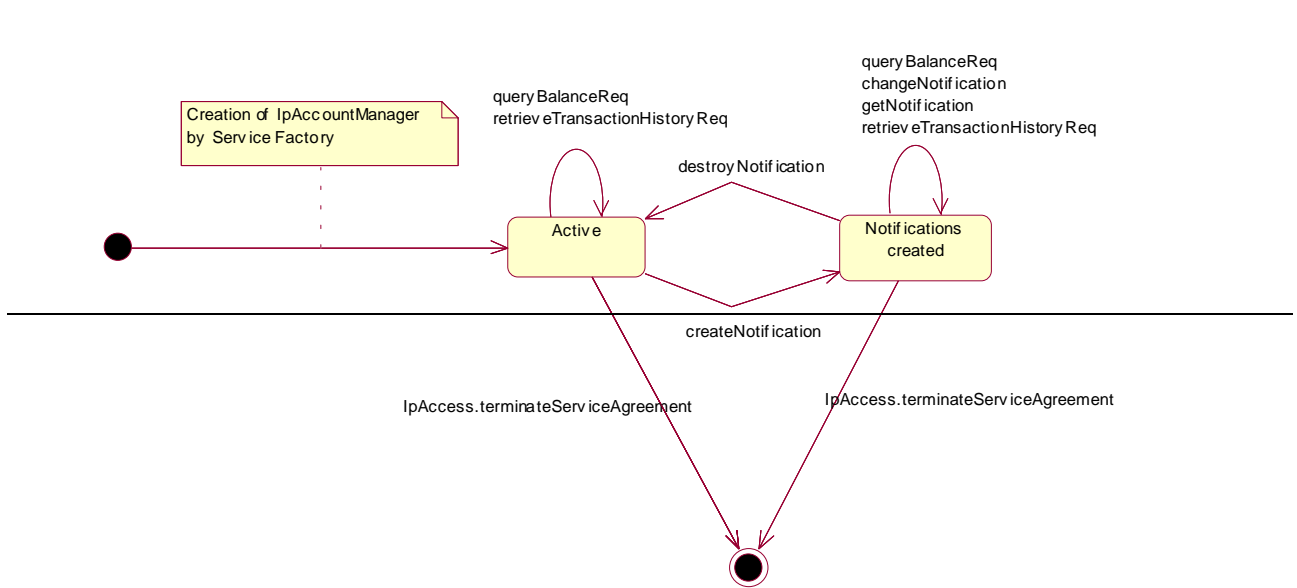


Figure : Application view on the IpAccountManager

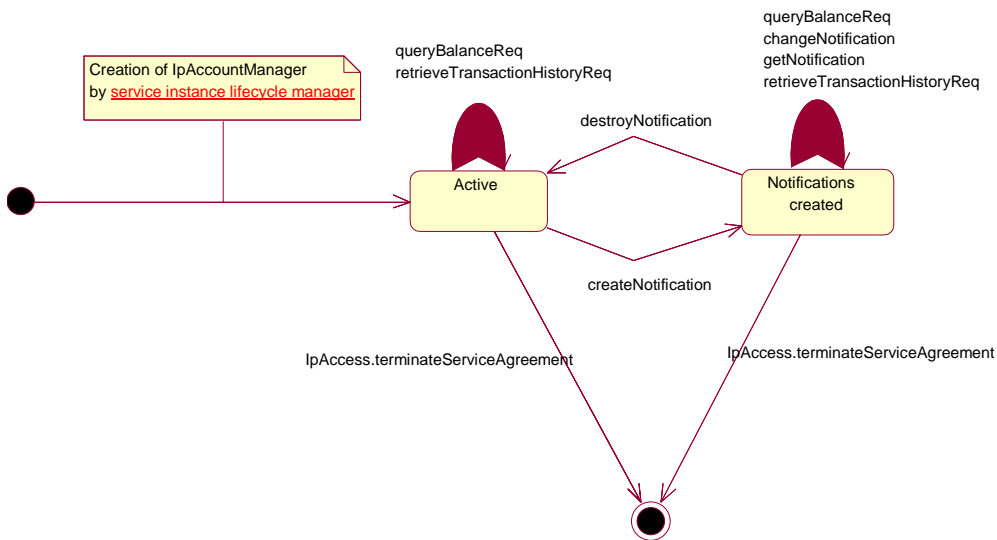


Figure : Application view on the IpAccountManager