

3GPP TS 29.198-11 V2.0.0 (2001-06)

Technical Specification

**3rd Generation Partnership Project;
Technical Specification Group Core Network;
Open Service Access (OSA);
Application Programming Interface (API);
Part 11: Account Management
(Release 4)**



The present document has been developed within the 3rd Generation Partnership Project (3GPP™) and may be further elaborated for the purposes of 3GPP.

The present document has not been subject to any approval process by the 3GPP Organizational Partners and shall not be implemented. This Specification is provided for future development work within 3GPP only. The Organizational Partners accept no liability for any use of this Specification. Specifications and reports for implementation of the 3GPP™ system should be obtained via the 3GPP Organizational Partners' Publications Offices.

Keywords

UMTS, API, OSA

3GPP

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

<http://www.3gpp.org>

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.

© 2001, 3GPP Organizational Partners (ARIB, CWTS, ETSI, T1, TTA, TTC).
All rights reserved.

Contents

Foreword.....	5
Introduction.....	5
1 Scope	6
2 References	6
3 Definitions and abbreviations.....	6
3.1 Definitions.....	6
3.2 Abbreviations	7
4 Account Management SCF.....	7
5 Sequence Diagrams.....	7
5.1 Standard Transaction History Retrieval	7
5.2 Standard Query Handling.....	8
5.3 Standard Notification handling	9
6 Class Diagrams.....	10
7 The Service Interface Specifications	12
7.1 Interface Specification Format	12
7.1.1 Interface Class	12
7.1.2 Method descriptions	12
7.1.3 Parameter descriptions.....	13
7.1.4 State Model.....	13
7.2 Base Interface.....	13
7.2.1 Interface Class IpInterface.....	13
7.3 Service Interfaces	13
7.3.1 Overview	13
7.4 Generic Service Interface.....	13
7.4.1 Interface Class IpService	13
8 Account Management Interface Classes.....	14
8.1 Interface Class IpAccountManager	14
8.2 Interface Class IpAppAccountManager	17
9 State Transition Diagrams	19
9.1 State Transition Diagrams for IpAccountManager.....	19
9.1.1 Active State	20
9.1.2 Notifications created State.....	20
10 Account Management Service Properties.....	20
11 Data Definitions.....	22
11.1 Account Management Data Definitions	22
11.1.1 TpBalanceQueryError	22
11.1.2 TpChargingEventName	22
11.1.3 TpBalanceInfo	22
11.1.4 TpChargingEventInfo.....	23
11.1.5 TpChargingEventCriteria	23
11.1.6 TpChargingEventNameSet.....	24
11.1.7 TpChargingEventCriteriaResult	24
11.1.8 TpBalance.....	24
11.1.9 TpBalanceSet.....	24
11.1.10 TpTransactionHistory.....	24
11.1.11 TpTransactionHistorySet.....	24
11.1.12 TpTransactionHistoryStatus	24

12 Exception Classes 25

Annex A (normative): OMG IDL Description of Account Management SCF 26

Annex B (informative): Change history 27

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
 - 1 presented to TSG for information;
 - 2 presented to TSG for approval;
 - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

Introduction

The present document is part 11 of a multi-part TS covering the 3rd Generation Partnership Project: Technical Specification Group Core Network; Open Service Access (OSA); Application Programming Interface (API), as identified below. The **API specification** (3GPP TS 29.198) is structured in the following Parts:

Part 1:	Overview	
Part 2:	Common Data Definitions	
Part 3:	Framework	
Part 4:	Call Control SCF	
Part 5:	User Interaction SCF	
Part 6:	Mobility SCF	
Part 7:	Terminal Capabilities SCF	
Part 8:	Data Session Control SCF	
Part 9:	Generic Messaging SCF	(not part of 3GPP Release 4)
Part 10:	Connectivity Manager SCF	(not part of 3GPP Release 4)
Part 11:	Account Management SCF	
Part 12:	Charging SCF	

The **Mapping specification of the OSA APIs and network protocols** (3GPP TR 29.998) is also structured as above. A mapping to network protocols is however not applicable for all Parts, but the numbering of Parts is kept. Also in case a Part is not supported in a Release, the numbering of the parts is maintained.

OSA API specifications 29.198-family		OSA API Mapping - 29.998-family	
29.198-1	Part 1: Overview	29.998-1	Part 1: Overview
29.198-2	Part 2: Common Data Definitions	29.998-2	Not Applicable
29.198-3	Part 3: Framework	29.998-3	Not Applicable
29.198-4	Part 4: Call Control SCF	29.998-4-1	Subpart 1: Generic Call Control – CAP mapping
		29.998-4-2	
29.198-5	Part 5: User Interaction SCF	29.998-5-1	Subpart 1: User Interaction – CAP mapping
		29.998-5-2	
		29.998-5-3	
		29.998-5-4	Subpart 4: User Interaction – SMS mapping
29.198-6	Part 6: Mobility SCF	29.998-6	User Status and User Location – MAP mapping
29.198-7	Part 7: Terminal Capabilities SCF	29.998-7	Not Applicable
29.198-8	Part 8: Data Session Control SCF	29.998-8	Data Session Control – CAP mapping
29.198-9	Part 9: Generic Messaging SCF	29.998-9	Not Applicable
29.198-10	Part 10: Connectivity Manager SCF	29.998-10	Not Applicable
29.198-11	Part 11: Account Management SCF	29.998-11	Not Applicable
29.198-12	Part 12: Charging SCF	29.998-12	Not Applicable

1 Scope

The present document is Part 11 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 concepts and the functional architecture for the OSA are contained in 3GPP TS 23.127 [3]. The requirements for OSA are contained in 3GPP TS 22.127 [2].

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

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

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

This specification has been defined jointly between 3GPP TSG CN WG5, ETSI SPAN 12 and the Parlay Consortium, in co-operation with the JAIN consortium.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

- [1] 3GPP TS 29.198-1 "Open Service Access; Application Programming Interface; Part 1: Overview".
- [2] 3GPP TS 22.127: "Stage 1 Service Requirement for the Open Service Access (OSA) (Release 4)".
- [3] 3GPP TS 23.127: "Virtual Home Environment (Release 4)".
- [4] ISO-4217:1995: "".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TS 29.198-1 [1] apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in TS 29.198-1 [1] apply.

4 Account Management SCF

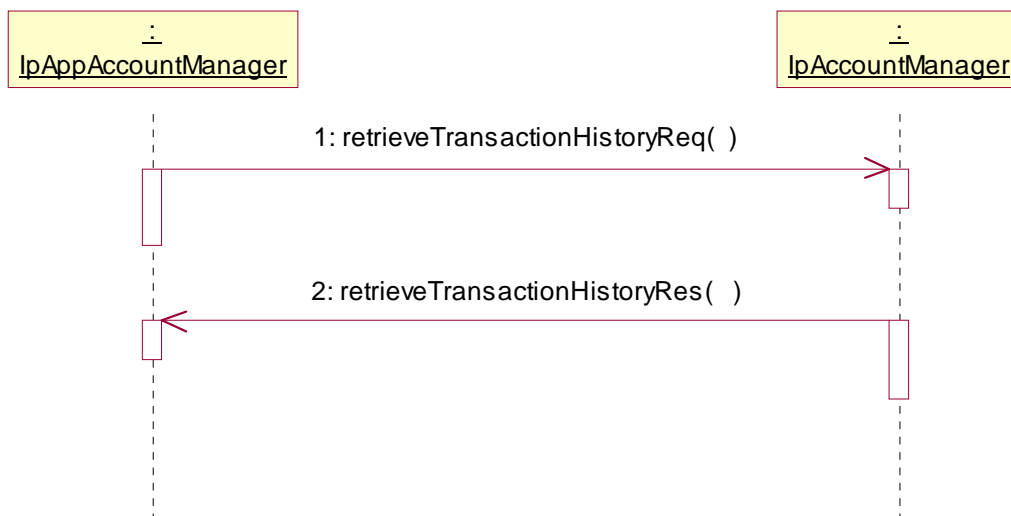
The following clauses describe each aspect of the Account Management Service Capability Feature (SCF).

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 progression of internal processes either in the application, or Gateway.
- The Data definitions section 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 of this specification.

5 Sequence Diagrams

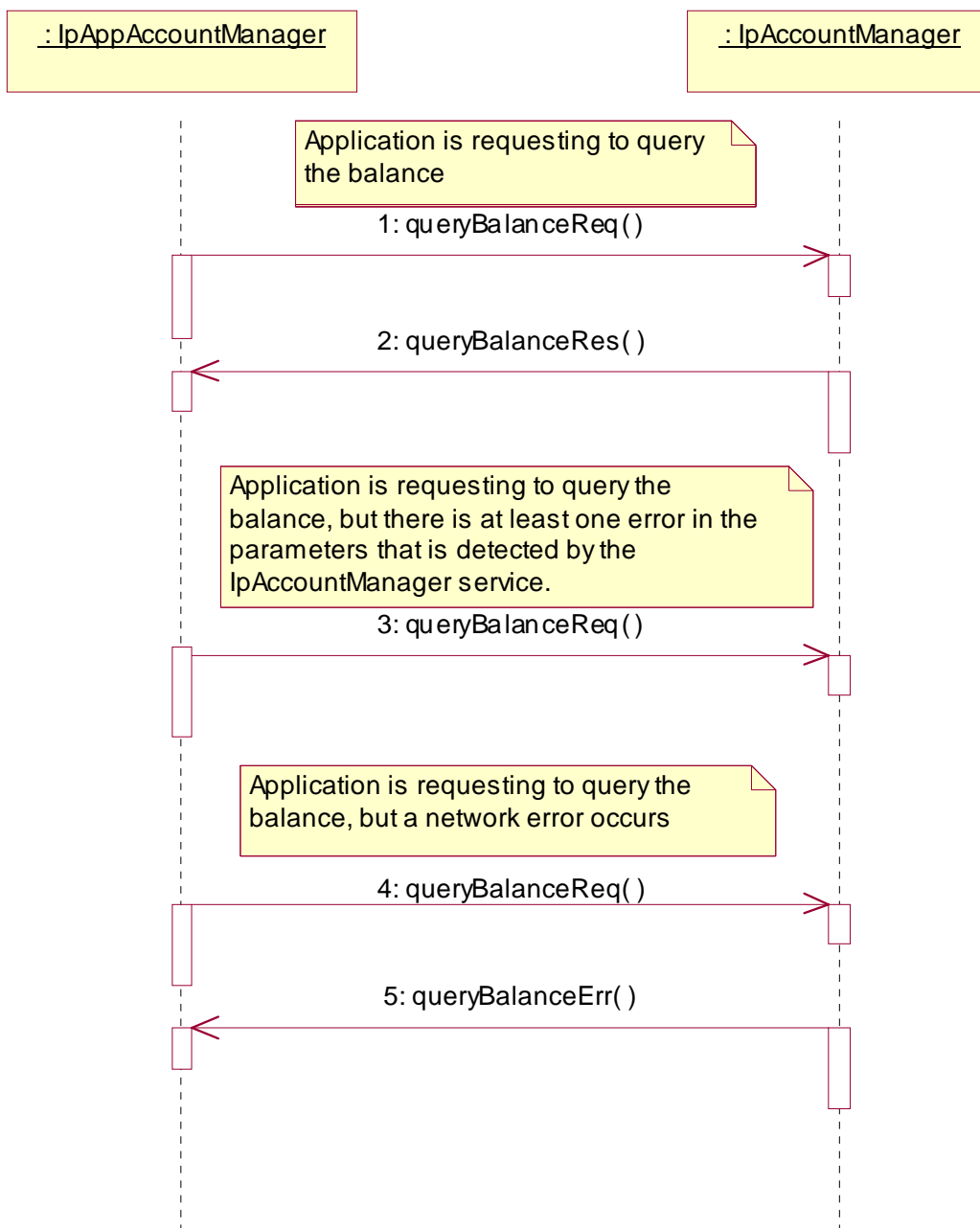
5.1 Standard Transaction History Retrieval



1: This message is used by the application to retrieve a transaction history for a certain subscriber's account.

2: This method passes the result of the transaction history retrieval request for a specific user to its callback object.

5.2 Standard Query Handling



1: This message is used to query the balance of the account of one or several users.

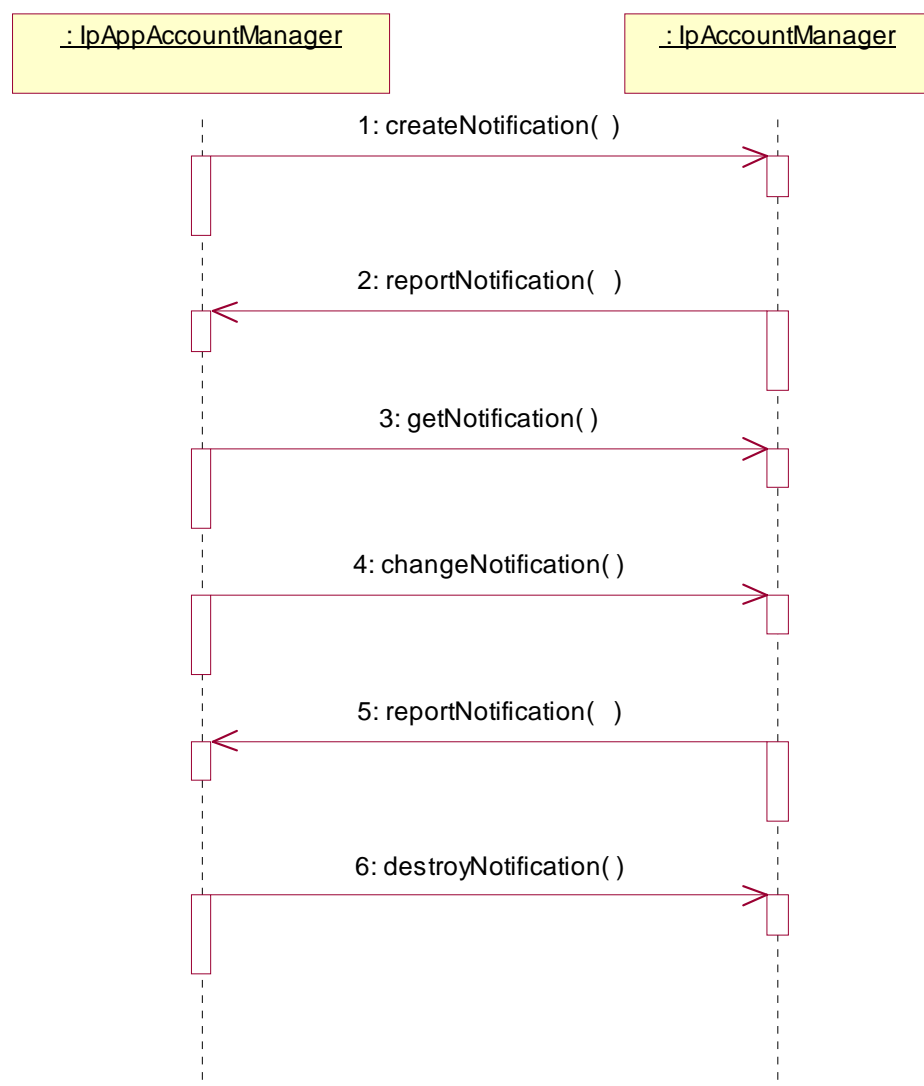
2: This message passes the result of the balance query for one or several users to its callback object.

3: This scenario shows the case where at least one error in the parameters of the message is detected by the IpAccountManager object. An exception will be thrown.

4: This scenario shows the case where a network error occurs.

5: This message passes the error of the balance query. No exception is thrown.

5.3 Standard Notification handling



1: This message is used by the application to request notifications from the IpAccountManager service on certain criteria for one or several users.

2: This message is used by the IpAccountManager service to report a charging event that meets the criteria set in the createNotification message.

3: The application can request the current criteria set in the IpAccountManager service by invoking the getNotification method.

4: This message is used by the application to change the criteria initially created by createNotification, and previously obtained by getNotification.

5: This message is used by the IpAccountManager service to report a charging event that meets the new criteria.

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

6 Class Diagrams

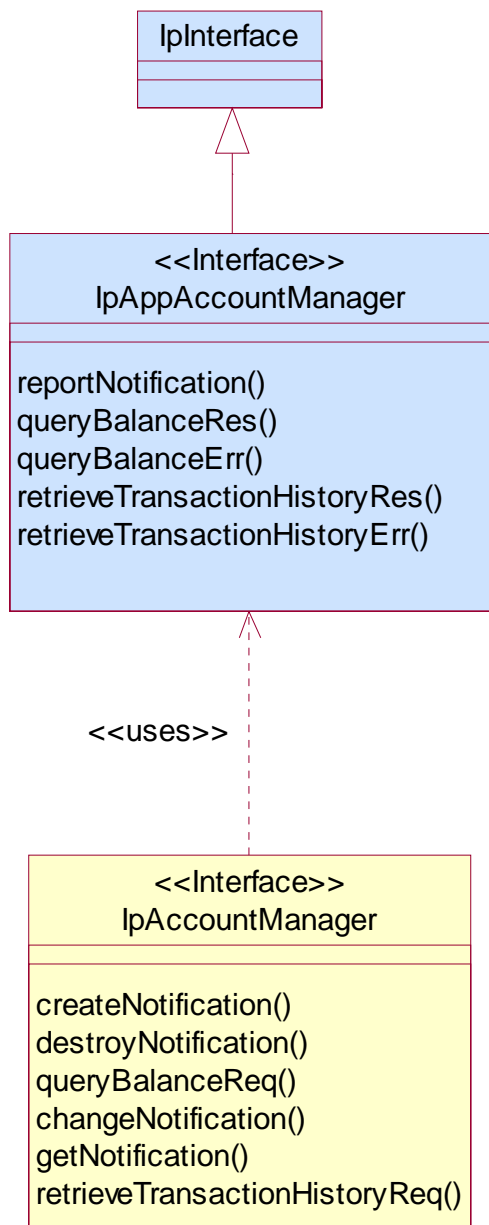


Figure: Application Interfaces

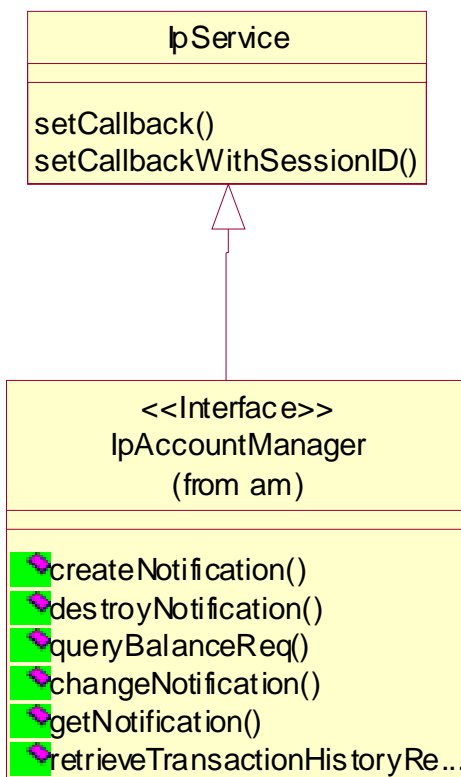


Figure: Service Interfaces

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.

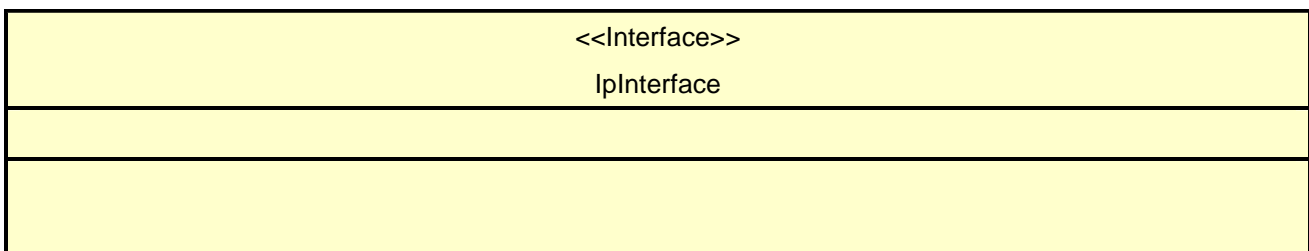
7.1.4 State Model

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.

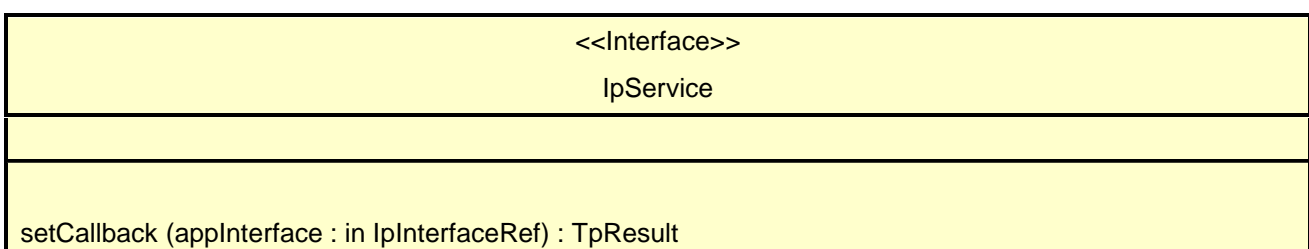
The interfaces that are implemented by the services are denoted as 'Service Interface'. The corresponding interfaces that must be implemented by the application (e.g. for API callbacks) are denoted as 'Application Interface'.

7.4 Generic Service Interface

7.4.1 Interface Class IpService

Inherits from: IpInterface

All service interfaces inherit from the following interface.



setCallbackWithSessionID (appInterface : in IpInterfaceRef, sessionID : in TpSessionID) : TpResult
--

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 use 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

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
createNotification (ChargingEventCriteria : in TpChargingEventCriteria, assignmentId : out TpAssignmentIDRef) : TpResult
destroyNotification (assignmentId : in TpAssignmentID) : TpResult
queryBalanceReq (users : in TpAddressSet, queryId : out TpSessionIDRef) : TpResult
changeNotification (assignmentID : in TpAssignmentID, eventCriteria : in TpChargingEventCriteria) : TpResult
getNotification (eventCriteria : out TpChargingEventCriteriaResultRef) : TpResult
retrieveTransactionHistoryReq (user : in TpAddress, transactionInterval : in TpTimeInterval, retrievalID : out TpSessionIDRef) : TpResult

*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 TpSessionIDRef

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 TpSessionIDRef

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 TpSessionID, balances : in TpBalanceSet) : TpResult queryBalanceErr (queryId : in TpSessionID, cause : in TpBalanceQueryError) : TpResult retrieveTransactionHistoryRes (retrievalID : in TpSessionID, transactionHistory : in TpTransactionHistorySet) : TpResult retrieveTransactionHistoryErr (retrievalID : in TpSessionID, 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 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.

9 State Transition Diagrams

9.1 State Transition Diagrams for IpAccountManager

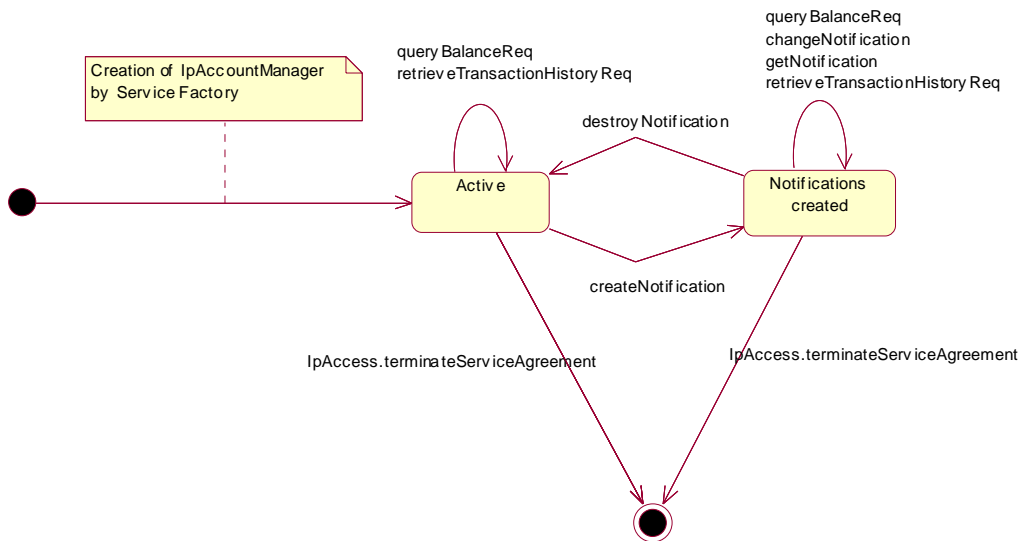


Figure : Application view on the IpAccountManager

9.1.1 Active State

In this state a relation between the Application and the Account Management has been established. The state allows the application to indicate that it is interested in charging related events, by calling createNotification. In case such an event occurs, Account Manager will inform the application by invoking the operation reportNotification() on the IpAppAccountManager interface. The application can also indicate it is no longer interested in certain charging related events by calling destroyNotification().

9.1.2 Notifications created State

When the Account Manager is in the Notifications created state, events requested with createNotification() will be forwarded to the application. In this state the application can request to change the notifications or query the Account Manager for the notifications currently set.

10 Account Management Service Properties

The following table lists properties relevant for the Account Management API.

Property	Type	Description/Interpretation
P_EVENT_TYPES	INTEGER_SET	Indicates the event types supported by the SCS. Static events are the events by which applications are initiated.
P_ADDRESSPLAN	INTEGER_SET	Indicates the supported address plan (defined in TpAddressPlan.) E.g. {P_ADDRESS_PLAN_E164, P_ADDRESS_PLAN_IP}

The previous table lists properties related to the capabilities of the SCS itself. The following table lists properties that are used in the context of the Service Level Agreement, e.g. to restrict the access of applications to the capabilities of the SCS.

Property	Type	Description/Interpretation
P_TRIGGERING_ADDRESSES	ADDRESS_RANGE_SET	Indicates for which numbers the notification may be set. For terminating notifications it applies to the terminating number,

		for originating notifications it applies only to the originating number.
P_CURRENCY_ALLOWED	STRING_SET	Indicates the currencies that can be returned in the queryBalanceRes. The valid values for the string set are according to ISO-4217:1995. E.g. {"EUR", "NLG"}.
P_HISTORY_ALLOWED	STRING_SET	Indicates the length of the transaction history interval that is allowed to be retrieved by the application. The valid values for the string are according to TpDateAndTime. The string-set will be of format {"lower_start_time", "upper_stop_time"}, e.g. {"1998-12-04 10:30", "1999-12-04 10:30"}
P_BULK_QUERY_ALLOWED	BOOLEAN_SET	Indicates whether the application is allowed to issue a queryBalanceReq for more than one user. Value = TRUE : the users parameter of type TpAddressSet may contain more than one user. Value = FALSE : the users parameter of type TpAddressSet may contain only one user.

11 Data Definitions

11.1 Account Management Data Definitions

This clause provides the Account Management specific data definitions necessary to support the OSA interface specification.

The general format of a data definition specification is the following:

- Data type, that shows the name of the data type.
- Description, that describes the data type.
- Tabular specification, that specifies the data types and values of the data type.
- Example, if relevant, shown to illustrate the data type.

11.1.1 TpBalanceQueryError

Defines an error that is reported by the Charging service capability feature as a result of a balance query request.

Name	Value	Description
P_BALANCE_QUERY_OK	0	No error occurred while processing the request
P_BALANCE_QUERY_ERROR_UNDEFINED	1	General error, unspecified
P_BALANCE_QUERY_UNKNOWN_SUBSCRIBER	2	Subscriber for which balance is queried is unknown
P_BALANCE_QUERY_UNAUTHORIZED_APPLICATION	3	Application is not authorized to query balance
P_BALANCE_QUERY_SYSTEM_FAILURE	4	System failure. The request could not be handled

11.1.2 TpChargingEventName

Defines the charging event for which notifications can be requested by the application.

Name	Value	Description
P_AM_CHARGING	0	End user's account has been charged by an application
P_AM_RECHARGING	1	End user has recharged the account
P_AM_ACCOUNT_LOW	2	Account balance is below the balance threshold
P_AM_ACCOUNT_ZERO	3	Account balance is at zero
P_AM_ACCOUNT_DISABLED	4	Account has been disabled

11.1.3 TpBalanceInfo

Defines the structure of data elements that specifies detailed balance info.

Structured Member Name	Structured Member Type	Description
Currency	TpString	Currency unit according to ISO-4217:1995 [4]
ValuePartA	TpInt32	This data type is identical to a TpInt32 and specifies the most significant part of the composed value. A currency amount is composed as follows: (ValuePartA*2 ³² + ValuePartB) * 0.0001)
ValuePartB	TpInt32	This data type is identical to a TpInt32 and specifies the least significant part of the composed value.
Exponent	TpInt32	Specifies the position of the decimal point in the currency amount made up of the unitPart and the fractionPart, as described above. E.g. an exponent of 4 means a pure integer value, whereas an exponent of 2 means an accuracy of 0.01.
AdditionalInfo	TpString	Descriptive string, containing additional information, which is sent to the application without prior evaluation.

As an example, the currency amount composed of a Currency of EUR, a ValuePartA of 0, a ValuePartB of 10000, and an exponent of 2 yields a currency amount of € 100.00.

Valid Currencies are:

ADP, AED, AFA, ALL, AMD, ANG, AON, AOR, ARS, ATS, AUD, AWG, AZM, BAM, BBD, BDT, BEF, BGL, BGN, BHD, BIF, BMD, BND, BOB, BOV, BRL, BSD, BTN, BWP, BYB, BZD, CAD, CDF, CHF, CLF, CLP, CNY, COP, CRC, CUP, CVE, CYP, CZK, DEM, DJF, DKK, DOP, DZD, ECS, ECV, EEK, EGP, ERN, ESP, ETB, EUR, FIM, FJD, FKP, FRF, GBP, GEL, GHC, GIP, GMD, GNF, GRD, GTQ, GWP, GYD, HKD, HNL, HRK, HTG, HUF, IDR, IEP, ILS, INR, IQD, IRR, ISK, ITL, JMD, JOD, JPY, KES, KGS, KHR, KMF, KPW, KRW, KWD, KYD, KZT, LAK, LBP, LKR, LRD, LSL, LTL, LUF, LVL, LYD, MAD, MDL, MGF, MKD, MMK, MNT, MOP, MRO, MTL, MUR, MVR, MWK, MXN, MXV, MYR, MZM, NAD, NGN, NIO, NLG, NOK, NPR, NZD, OMR, PAB, PEN, PGK, PHP, PKR, PLN, PTE, PYG, QAR, ROL, RUB, RUR, RWF, SAR, SBD, SCR, SDD, SEK, SGD, SHP, SIT, SKK, SLL, SOS, SRG, STD, SVC, SYP, SZL, THB, TJR, TMM, TND, TOP, TPE, TRL, TTD, TWD, TZS, UAH, UGX, USD, USN, USS, UYU, UZS, VEB, VND, VUV, WST, XAF, XAG, XAU, XBA, XBB, XBC, XBD, XCD, XDR, XFO, XFU, XOF, XPD, XPF, XPT, XTS, XXX, YER, YUM, ZAL, ZAR, ZMK, ZRN, ZWD.

XXX is used for transactions where no currency is involved.

11.1.4 TpChargingEventInfo

Defines the structure of data elements that specifies charging event information.

Structured Member Name	Structured Member Type	Description
ChargingEventName	TpChargingEventName	The charging event for which notifications can be requested by the application
CurrentBalanceInfo	TpBalanceInfo	The current balance of the user's account
ChargingEventTime	TpTime	The time at which the charging event occurred.

11.1.5 TpChargingEventCriteria

Defines the structure of data elements that specifies charging event criteria.

Structured Member Name	Structured Member Type	Description
Users	TpAddressSet	Specifies the user(s) for which the charging events are requested to be reported.
ChargingEvents	TpChargingEventNameSet	Specifies the specific charging event criteria used by the application to define the event required.

11.1.6 TpChargingEventNameSet

Defines a collection of TpChargingEventName elements.

11.1.7 TpChargingEventCriteriaResult

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

Sequence Element Name	Sequence Element Type
ChargingEventCriteria	TpChargingEventCriteria
AssignmentID	TpAssignmentID

11.1.8 TpBalance

Defines the structure of data elements that specifies a balance.

Structured Member Name	Structured Member Type	Description
UserID	TpAddress	Specifies the user to whom the account belongs to.
StatusCode	TpBalanceQueryError	Specifies the status code for the balance query request.
BalanceInfo	TpBalanceInfo	Specifies the balance information for the user.

11.1.9 TpBalanceSet

Defines a collection of TpBalance elements.

11.1.10 TpTransactionHistory

This data type is a sequence of data elements that describes the transaction history.

Sequence Element Name	Sequence Element Type	Description
TransactionID	TpAssignmentID	Specifies the ID of the specific transaction
TimeStamp	TpDateAndTime	Specifies the date and time when the specific transaction was processed.
AdditionalInfo	TpString	Specifies a free format string providing additional information on the specific transaction. This could be the applicationDescription provided with the actual transaction.

11.1.11 TpTransactionHistorySet

Defines a collection of TpTransactionHistory elements.

11.1.12 TpTransactionHistoryStatus

Defines a status code that is reported by the Account Manager service capability feature as a result of a transaction history retrieval request.

Name	Value	Description
P_AM_TRANSACTION_ERROR_UNSPECIFIED	0	General error, unspecified
P_AM_TRANSACTION_INVALID_INTERVAL	1	An invalid interval for the transaction history was specified.
P_AM_TRANSACTION_UNKNOWN_ACCOUNT	2	No account for the specified user is known.
P_AM_TRANSACTION_UNAUTHORIZED_APPLICATION	3	Application is not authorized to query balance.
P_AM_TRANSACTION_PROCESSING_ERROR	4	A processing error occurred while compiling the transaction history.
P_AM_TRANSACTION_SYSTEM_FAILURE	5	System failure. The request could not be handled

12 Exception Classes

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

Name	Description
P_UNAUTHORIZED_APPLICATION	Application is not authorized to perform charging operations

Each exception class contains the following structure:

Structure Element Name	Structure Element Type	Structure Element Description
<code>extraInformation</code>	<code>TpString</code>	Carries extra information to help identify the source of the exception, e.g. a parameter name

Annex A (normative): OMG IDL Description of Account Management SCF

The OMG IDL representation of this interface specification is contained in a text file (am.idl contained in archive 2919811IDL.ZIP) which accompanies the present document.

Annex B (informative): Change history

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
Mar 2001	CN_11	NP-010134	047	--	CR 29.198: for moving TS 29.198 from R99 to Rel 4 (N5-010158)	3.2.0	1.0.0
Jun 2001	CN_12	NP-010327	--	--	Approved at TSG CN#12 and placed under Change Control	2.0.0	4.0.0

```
//Source file: am.idl
//Date: 10 June 2001
```

```
#ifndef __AM_DEFINED
#define __AM_DEFINED
```

```
#include "osa.idl"
```

```
module org {
```

```
    module csapi {
```

```
        module am {
```

```
            enum TpChargingEventName {
                P_AM_CHARGING,
                P_AM_RECHARGING,
                P_AM_ACCOUNT_LOW,
                P_AM_ACCOUNT_ZERO,
                P_AM_ACCOUNT_DISABLED
            };
```

```
            enum TpBalanceQueryError {
                P_BALANCE_QUERY_OK,
                P_BALANCE_QUERY_ERROR_UNDEFINED,
                P_BALANCE_QUERY_UNKNOWN_SUBSCRIBER,
                P_BALANCE_QUERY_UNAUTHORIZED_APPLICATION,

                P_BALANCE_QUERY_SYSTEM_FAILURE
            };
```

```
            struct TpBalanceInfo {
                TpString Currency;
                TpInt32 ValuePartA;
                TpInt32 ValuePartB;
                TpInt32 Exponent;
                TpString AdditionalInfo;
            };
```

```
            struct TpChargingEventInfo {
                TpChargingEventName ChargingEventName;
                TpBalanceInfo CurrentBalanceInfo;
                TpTime ChargingEventTime;
            };
```

```
            struct TpBalance {
                TpAddress UserID;
                TpBalanceQueryError StatusCode;
            };
```

```

        TpBalanceInfo BalanceInfo;
};

typedef sequence <TpBalance> TpBalanceSet;

struct TpTransactionHistory {
    TpAssignmentID TransactionID;
    TpDateAndTime TimeStamp;
    TpString AdditionalInfo;
};

typedef sequence <TpTransactionHistory>
TpTransactionHistorySet;

enum TpTransactionHistoryStatus {
    P_AM_TRANSACTION_ERROR_UNSPECIFIED,
    P_AM_TRANSACTION_INVALID_INTERVAL,
    P_AM_TRANSACTION_UNKNOWN_ACCOUNT,
    P_AM_TRANSACTION_UNAUTHORIZED_APPLICATION,
    P_AM_TRANSACTION_PROCESSING_ERROR,
    P_AM_TRANSACTION_SYSTEM_FAILURE
};

typedef sequence <TpChargingEventName> TpChargingEventNameSet;

struct TpChargingEventCriteria {
    TpAddressSet Users;
    TpChargingEventNameSet ChargingEvents;
};

struct TpChargingEventCriteriaResult {
    TpChargingEventCriteria ChargingEventCriteria;
    TpAssignmentID AssignmentID;
};

exception P_UNAUTHORIZED_APPLICATION {
    TpString extraInformation;
};

interface IpAppAccountManager : IpInterface {

    void reportNotification (
        in TpChargingEventInfo chargingEventInfo,
        in TpAssignmentID assignmentId
    );
};

```

```

void queryBalanceRes (
    in TpSessionID queryId,
    in TpBalanceSet balances
);

void queryBalanceErr (
    in TpSessionID queryId,
    in TpBalanceQueryError cause
);

void retrieveTransactionHistoryRes (
    in TpSessionID retrievalID,
    in TpTransactionHistorySet transactionHistory
);

void retrieveTransactionHistoryErr (
    in TpSessionID retrievalID,
    in TpTransactionHistoryStatus
transactionHistoryError
);

};

interface IpAccountManager : IpService {

    void createNotification (
        in TpChargingEventCriteria ChargingEventCriteria,

        out TpAssignmentID assignmentId
    )
    raises (TpCommonExceptions, P_INVALID_CRITERIA,
P_INVALID_EVENT_TYPE, P_UNKNOWN_SUBSCRIBER);

    void destroyNotification (
        in TpAssignmentID assignmentId
    )
    raises (TpCommonExceptions,
P_INVALID_ASSIGNMENT_ID);

    void queryBalanceReq (
        in TpAddressSet users,
        out TpSessionID queryId
    )
    raises (TpCommonExceptions, P_UNKNOWN_SUBSCRIBER,
P_UNAUTHORIZED_APPLICATION);

    void changeNotification (
        in TpAssignmentID assignmentID,
        in TpChargingEventCriteria eventCriteria
    )
    raises (TpCommonExceptions,
P_INVALID_ASSIGNMENT_ID, P_INVALID_CRITERIA, P_INVALID_EVENT_TYPE);

```

```
void getNotification (
    out TpChargingEventCriteriaResult eventCriteria
)
    raises (TpCommonExceptions);

void retrieveTransactionHistoryReq (
    in TpAddress user,
    in TpTimeInterval transactionInterval,
    out TpSessionID retrievalID
)
    raises (TpCommonExceptions, P_UNKNOWN_SUBSCRIBER,
P_UNAUTHORIZED_APPLICATION, P_INVALID_TIME_AND_DATE_FORMAT);

};

};

};

#endif
```