

**3GPP TSG-CN Meeting #24
02 – 04 June 2004, Seoul, KOREA**

NP-040271

Source: CN5 (OSA)
Title: Rel-6 CR 29.198-11 OSA API Part 11: Account management (Account Management missing needed features)
Agenda item: 9.7 (OSA Enhancements [\[OSA3\]](#))
Document for: APPROVAL

Doc-1st-	Spec	CR	Rev	Phase	Subject	Cat	Version	Doc-2nd-	Workite
NP-040271	29.198-11	024	-	Rel-6	Account Management missing needed features	B	6.0.1	N5-040054	OSA3

CHANGE REQUEST

⌘ **29.198-11 CR 024** ⌘ rev **-** ⌘ Current version: **6.0.1** ⌘

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

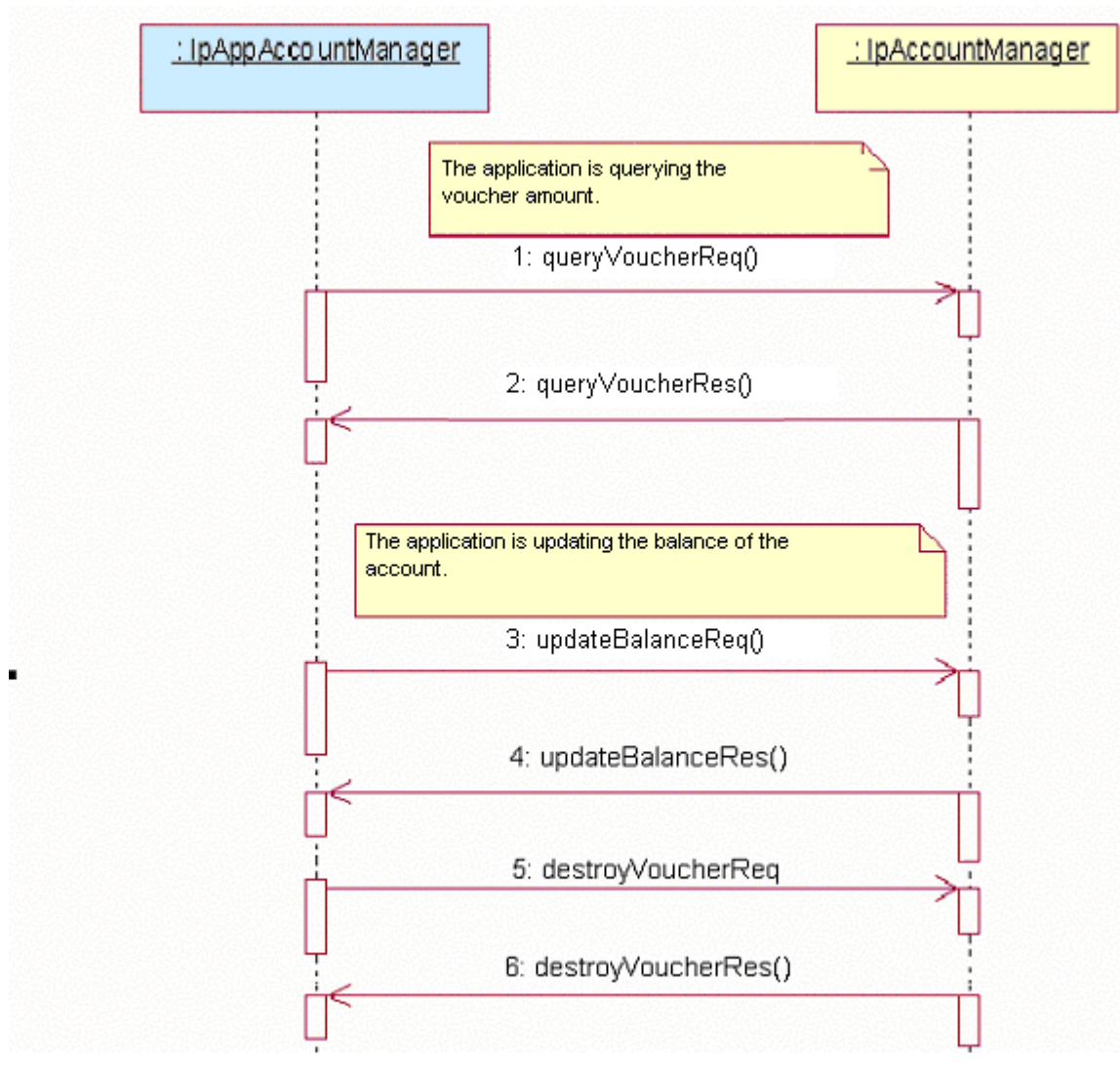
Title:	⌘ Account Management missing needed features		
Source:	⌘ CN5 Scott Broussard (IBM, scottjb@us.ibm.com)		
Work item code:	⌘ OSA3	Date:	⌘ 04/2/2004
Category:	⌘ B	Release:	⌘ REL-6
	Use <u>one</u> of the following categories: F (correction) A (corresponds to a correction in an earlier release) B (addition 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) Rel-6 (Release 6)

Reason for change:	⌘ The recent Parlay X spec contains an account management module that contains methods for balanceUpdate(), voucherUpdate() and getCreditExpiryDate(). These concepts do not exist in the OSA Account Management service. The voucher is an indirect reference to an amount. The referenceCode provided in the Parlay X API for each unique request is not propagated to the OSA API, but maintained and associated with the OSA TpAssignmentID in the usage records of the Parlay X implementation.		
Summary of change:	⌘ Add the necessary methods to OSA Account Management		
Consequences if not approved:	⌘ Parlay X can not be implemented using OSA interfaces. Changes derived from: ftp://ftp.3gpp.org/specs/2003-12/Rel-6/29_series/		

Clauses affected:	⌘ 5, 8, 11, Annex A						
Other specs affected:	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">Y</td> <td style="width: 20px; text-align: center;">N</td> </tr> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> </table> Other core specifications	Y	N	<input type="checkbox"/>	<input checked="" type="checkbox"/>	⌘	
Y	N						
<input type="checkbox"/>	<input checked="" type="checkbox"/>						
	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">Y</td> <td style="width: 20px; text-align: center;">N</td> </tr> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> </table> Test specifications	Y	N	<input type="checkbox"/>	<input checked="" type="checkbox"/>	⌘	
Y	N						
<input type="checkbox"/>	<input checked="" type="checkbox"/>						
	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">Y</td> <td style="width: 20px; text-align: center;">N</td> </tr> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> </table> O&M Specifications	Y	N	<input type="checkbox"/>	<input checked="" type="checkbox"/>	⌘	
Y	N						
<input type="checkbox"/>	<input checked="" type="checkbox"/>						
Other comments:	⌘						

Change in Clause 5.4

5.4 Standard Voucher Handling



[1: This message is used to get a voucher for an amount.](#)

[2: This message used to return the amount requested.](#)

[3: This message is used to update the balance of the account.](#)

[4: This message is used to return the confirmation of the update.](#)

[5: This message is used to remove the voucher.](#)

[6: This message is used to confirm that the voucher is removed.](#)

End of Change in Clause 5.4

Change in Clause 8.1

58 Account Management Interface Classes

5.18.1 Interface Class IpAccountManager

Inherits from: IpService.

The account manager interface provides methods for ~~managing~~~~monitoring~~ accounts. Applications can use this interface to enable or disable charging-related event notifications and to ~~manage~~~~query~~ account balances. [Vouchers allow indirect references to amounts that can be applied to the account.](#)

<<Interface>>

IpAccountManager

```
createNotification (appAccountManager : in IpAppAccountManagerRef, chargingEventCriteria : in
    TpChargingEventCriteria) : TpAssignmentID
destroyNotification (assignmentId : in TpAssignmentID) : void
queryBalanceReq (users : in TpAddressSet) : TpAssignmentID
changeNotification (assignmentID : in TpAssignmentID, eventCriteria : in TpChargingEventCriteria) : void
getNotification () : TpChargingEventCriteriaResultSet
retrieveTransactionHistoryReq (user : in TpAddress, transactionInterval : in TpTimeInterval) :
    TpAssignmentID
<<new>> enableNotifications (appAccountManager : in IpAppAccountManagerRef) : TpAssignmentID
<<new>> disableNotifications () : void
<<new>> queryBalanceExpiryDateReq(users: in TpAddressSet) : TpAssignmentID
<<new>> updateBalanceReq (user: in TpAddress, debit: in TpBoolean, amount: in TpBalanceInfo, period: in
    TpInt32) : TpAssignmentID
<<new>> createVoucherReq (user: in TpAddress, amount: in TpBalanceInfo ) : TpAssignmentID
<<new>> queryVoucherReq (voucherId: in TpAssignmentID) : TpAssignmentID
<<new>> destroyVoucherReq (voucherId: in TpAssignmentID) : TpAssignmentID
<<new>> queryUserVouchersReq (user: in TpAddress) : TpAssignmentID
```

End of Change in Clause 8.1

Change in Clause 8.1.3

8.1.3 Method queryBalanceReq()

This method is used by the application to query the balance of an account for one or several users. [A queryBalanceRes\(\) will be sent for all valid users, and a queryBalanceErr\(\) will be sent for all invalid users or other errors.](#)

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

TpAssignmentID

Raises

TpCommonExceptions, P_UNKNOWN_SUBSCRIBER, P_UNAUTHORIZED_APPLICATION

End of Change in Clause 8.1.3

Change in Clause 8.1.9

8.1.9 Method queryBalanceExpiryDateReq()

This method is used by the application to query the expiration date of the account. The returned date is the is the date the current balance will expire. Nil is returned if the balance does not expire. A queryBalanceExpiryDateRes() will be sent for all valid users, and a queryBalanceExpiryDateErr() will be sent for all invalid users or other errors.

Returns queryId: Specifies the ID of the balance expiry date query request.

Parameters

users : in TpAddressSet

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

Returns

TpAssignmentID

Raises

TpCommonExceptions, P_UNKNOWN_SUBSCRIBER, P_UNAUTHORIZED_APPLICATION

End of Change in Clause 8.1.9

Change in Clause 8.1.10

8.1.10 Method updateBalanceReq()

This method is used by the application to update the balance of the account.

Returns requestId: Specifies the ID of the update balance request.

Parameters

user : in TpAddress

Specifies the user for which the balance update will be done.

debit : in TpBoolean

Specifies that the amount will be subtracted from the balance if the debit parameter is true, or that the amount will be added to the account if the debit parameter is false.

amount : in TpBalanceInfo

Specifies the amount that will be added or subtracted to the user's account. The charge is specified as a currency amount.

period : in TpInt32

Specifies the period at which the balance will expire. The balance is requested to expire in a number of days indicated by the period parameter. The operator's policy may overrule this parameter. If the period parameter is 0, the operator's policy on the balance expiration is always in effect.

Returns

TpAssignmentID

Raises

TpCommonExceptions, P_UNKNOWN_SUBSCRIBER, P_UNAUTHORIZED_APPLICATION

End of Change in Clause 8.1.10

Change in Clause 8.1.11

8.1.11 Method createVoucherReq()

This method is used by the application to create a voucher for an amount.

Returns requestId: Specifies the ID of the voucher create request.

Parameters

user : in TpAddress

Specifies the user for which the voucher create will be done.

amount : in TpBalanceInfo

Specifies the amount of the voucher. The charge is specified as a currency amount.

Returns

TpAssignmentID

Raises

TpCommonExceptions, P_UNKNOWN_SUBSCRIBER, P_UNAUTHORIZED_APPLICATION

End of Change in Clause 8.1.11

Change in Clause 8.1.12

8.1.12 Method destroyVoucherReq()

This method is used by the application to destroy a voucher for an amount.

Returns requestId: Specifies the ID of the destroy voucher request.

Parameters

voucherId : in TpAssignmentID

Specifies the voucher to be destroyed.

Returns

TpAssignmentID

Raises

TpCommonExceptions, P_UNAUTHORIZED_APPLICATION, P_INVALID_ASSIGNMENTID

End of Change in Clause 8.1.12

Change in Clause 8.1.13

8.1.13 Method queryVoucherReq()

This method is used by the application to get the voucher information.

Returns queryId: Specifies the identifier for this request.

Parameters

voucherId : in TpAssignmentID

Specifies the voucher to be queried.

Returns

TpAssignmentID

Raises

TpCommonExceptions, P_UNAUTHORIZED_APPLICATION, P_INVALID_ASSIGNMENTID

End of Change in Clause 8.1.13

Change in Clause 8.1.14

8.1.14 Method queryUserVouchersReq()

This method is used by the application to get the vouchers for a user.

Returns queryId: Specifies the ID for the get vouchers request.

Parameters

user : in TpAddress

Specifies the user for which the vouchers will be returned.

Returns

TpAssignmentID

Raises

TpCommonExceptions, P_UNKNOWN_SUBSCRIBER, P_UNAUTHORIZED_APPLICATION, P_INVALID_ASSIGNMENTID

End of Change in Clause 8.1.14

Change in Clause 8.2

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) : void

queryBalanceRes (queryId : in TpAssignmentID, balances : in TpBalanceSet) : void

queryBalanceErr (queryId : in TpAssignmentID, cause : in TpBalanceQueryError) : void

retrieveTransactionHistoryRes (retrievalId : in TpAssignmentID, transactionHistory : in TpTransactionHistorySet) : void

retrieveTransactionHistoryErr (retrievalId : in TpAssignmentID, transactionHistoryError : in TpTransactionHistoryStatus) : void

<<new>> queryBalanceExpiryDateRes (queryId : in TpAssignmentID, balances : in TpBalanceExpiryDateSet) : void

<<new>> queryBalanceEpiryDateErr (queryId : in TpAssignmentID, cause : in TpBalanceQueryError) : void

<<new>> updateBalanceRes (requestId : in TpAssignmentID, balance : in TpBalance) : void

<<new>> updateBalanceErr (requestId : in TpAssignmentID, cause : in TpBalanceQueryError) : void

<<new>> createVoucherRes (requestId : in TpAssignmentID, voucherId : in TpAssignmentID) : void

<<new>> createVoucherErr (requestId : in TpAssignmentID, cause : in TpVoucherError) : void

<<new>> destroyVoucherRes (requestId : in TpAssignmentID, voucherId : in TpAssignmentID) : void

<<new>> destroyVoucherErr (requestId : in TpAssignmentID, voucherId : in TpAssignmentID, cause : TpVoucherError) : void

<<new>> queryVoucherRes (queryId : in TpAssignmentID, voucher : TpVoucher) : void

<<new>> queryVoucherErr (queryId : in TpAssignmentID, voucherId : in TpAssignmentID, cause : TpVoucherError) : void

<<new>> queryUserVouchersRes (queryId ; in TpAssignmentId, vouchers : TpVoucherSet) : void

<<new>> queryUserVouchersErr (queryId : in TpAssignmentId, cause: TpVoucherError) : void

End of Change in Clause 8.2

Change in Clause 8.2.6

8.2.6 Method queryBalanceExpiryDateRes()

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

Parameters

queryId : in TpAssignmentID

Specifies the ID of the balance query expiry date request.

balances : in TpBalanceExpiryDateSet

Specifies the balance expiry date for one or more user accounts.

End of Change in Clause 8.2.6

Change in Clause 8.2.7

8.2.7 Method queryBalanceExpiryDateErr()

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

Parameters

queryId : in TpAssignmentID

Specifies the ID of the balance query expiry date request.

cause : in TpBalanceQueryError

Specifies the error that led to the failure.

End of Change in Clause 8.2.7

Change in Clause 8.2.8

8.2.8 Method updateBalanceRes()

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

Parameters

requestId : in TpAssignmentID

Specifies the ID of the balance update request.

balance : in TpBalance

Specifies the balance of the account.

End of Change in Clause 8.2.8

Change in Clause 8.2.9

8.2.9 Method updateBalanceErr()

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

Parameters

requestId : in TpAssignmentID

Specifies the ID of the balance update request.

cause : in TpBalanceQueryError

Specifies the error that led to the failure.

End of Change in Clause 8.2.9

Change in Clause 8.2.10

8.2.10 Method createVoucherRes()

This method indicates that the request to create a voucher was successful.

Parameters

requestId : in TpAssignmentID

Specifies the ID of the voucher create request.

voucherId : in TpAssignmentID

Specifies the voucher.

End of Change in Clause 8.2.10

Change in Clause 8.2.11

8.2.11 Method createVoucherErr()

This method indicates that the request to create a voucher failed and it reports the cause of failure to the application.

Parameters

requestId : in TpAssignmentID

Specifies the ID of the voucher create request.

cause : in TpVoucherError

Specifies the error that led to the failure.

End of Change in Clause 8.2.11

Change in Clause 8.2.12

8.2.12 Method destroyVoucherRes()

This method indicates that the request to destroy a voucher was successful.

Parameters

[requestId : in TpAssignmentID](#)

[Specifies the ID of the request.](#)

[voucherId : in TpAssignmentID](#)

[Specifies the voucher.](#)

End of Change in Clause 8.2.12

Change in Clause 8.2.13

[8.2.13 Method destroyVoucherErr\(\)](#)

[This method indicates that the request to destroy a voucher failed and it reports the cause of failure to the application.](#)

Parameters

[requestId : in TpAssignmentID](#)

[Specifies the ID of the request.](#)

[voucherId : in TpAssignmentID](#)

[Specifies the voucher.](#)

[cause : in TpVoucherError](#)

[Specifies the error that led to the failure.](#)

End of Change in Clause 8.2.13

Change in Clause 8.2.14

[8.2.14 Method queryVoucherRes\(\)](#)

[This method indicates that the request to query the voucher information was successful and it reports the amount to the application.](#)

Parameters

[queryId : in TpAssignmentID](#)

[Specifies the ID of the query voucher amount request.](#)

[voucher : in TpVoucher](#)

[Specifies the information of the voucher.](#)

End of Change in Clause 8.2.14

Change in Clause 8.2.15

[8.2.15 Method queryVoucherErr\(\)](#)

[This method indicates that the request to query the voucher failed and it reports the cause of failure to the application.](#)

Parameters

[queryId : in TpAssignmentID](#)

[Specifies the ID of the query voucher amount request.](#)

[voucherId : in TpAssignmentID](#)

Specifies the voucher.

[cause : in TpVoucherError](#)

Specifies the error that led to the failure.

End of Change in Clause 8.2.15

Change in Clause 8.2.16

[8.2.16 Method queryUserVouchersRes\(\)](#)

This method indicates that the request to query the vouchers was successful and it reports the set of vouchers to the application.

Parameters

[queryId : in TpAssignmentID](#)

Specifies the ID of the query vouchers request.

[vouchers : in TpVoucherSet](#)

Specifies the set of vouchers for the user.

End of Change in Clause 8.2.16

Change in Clause 8.2.17

[8.2.17 Method queryUserVouchersErr\(\)](#)

This method indicates that the request to query the vouchers for a user failed and it reports the cause of failure to the application.

Parameters

[queryId : in TpAssignmentID](#)

Specifies the ID of the query vouchers request.

[cause : in TpVoucherError](#)

Specifies the error that led to the failure.

End of Change in Clause 8.2.17

Change in Clause 11.1.13

[11.1.13 TpBalanceExpiryDate](#)

Defines the structure of data elements that specifies a balance expiry date.

<u>Structure Element Name</u>	<u>Structure Element Type</u>	<u>Structure Element Description</u>
UserID	TpAddress	Specifies the user to whom the account belongs.
StatusCode	TpBalanceQueryError	Specifies the status code for the balance query request
ExpiryDate	TpDateAndTime	Defines the expiry date and time

End of Change in Clause 11.1.13

Change in Clause 11.1.14

[11.1.14 TpBalanceExpiryDateSet](#)

[Defines a collection of TpBalanceExpiryDate elements.](#)

End of Change in Clause 11.1.14

Change in Clause 11.1.15

[11.1.15 TpVoucherError](#)

[Defines an error that is reported as a result of a voucher request.](#)

<u>Name</u>	<u>Value</u>	<u>Description</u>
P_VOUCHER_OK	0	No error occurred while processing the request
P_VOUCHER_UNDEFINED	1	General error, unspecified
P_VOUCHER_UNKNOWN_SUBSCRIBER	2	Subscriber is unknown
P_VOUCHER_UNAUTHORIZED_APPLICATION	3	Application is not authorized
P_VOUCHER_SYSTEM_FAILURE	4	System failure. The request could not be handled

End of Change in Clause 11.1.15

Change in Clause 11.1.16

[11.1.16 TpVoucher](#)

[Defines the structure of data elements that specifies a voucher.](#)

<u>Structure Element Name</u>	<u>Structure Element Type</u>	<u>Structure Element Description</u>
VoucherID	TpAssignmentID	Specifies the unique identifier for this voucher.
UserID	TpAddress	Specifies the user to who the voucher belongs.
BalanceInfo	TpBalanceInfo	Specifies the amount of the voucher.

End of Change in Clause 11.1.16

Change in Clause 11.1.17

[11.1.17 TpVoucherSet](#)

[Defines the set of TpVoucher objects.](#)

End of Change in Clause 11.1.17

Change in Clause Annex A.1

Annex A (normative): CORBA IDL, Access Protocol

```
#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 {
```

```
    TpChargingEventNameSet ChargingEvents;
    TpAddressSet Users;
};
```

```
struct TpChargingEventCriteriaResult {
    TpChargingEventCriteria ChargingEventCriteria;
    TpAssignmentID AssignmentID;
};
```

```
exception P_UNAUTHORIZED_APPLICATION {
    TpString ExtraInformation;
};
```

```
typedef sequence <TpChargingEventCriteriaResult>
TpChargingEventCriteriaResultSet;
```

```
struct TpBalanceExpiryDate {
    TpAddress UserID;
    TpBalanceQueryError StatusCode;
    TpDateAndTime ExpiryDate;
};
```

```
typedef sequence <TpBalanceExpiryDate> TpBalanceExpiryDateSet;
```

```
enum TpVoucherError {
    P_VOUCHER_OK,
    P_VOUCHER_UNDEFINED,
    P_VOUCHER_UNKNOWN_SUBSCRIBER,
    P_VOUCHER_UNAUTHORIZED_APPLICATION,
    P_VOUCHER_SYSTEM_FAILURE
};
```

```
struct TpVoucher {
    TpAssignmentID VoucherID;
    TpAddress UserID;
    TpBalanceInfo BalanceInfo;
};
```

```
typedef sequence <TpVoucher> TpVoucherSet;
```

```
interface IpAppAccountManager : IpInterface {
```

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

```
    void queryBalanceRes (
        in TpAssignmentID queryId,
        in TpBalanceSet balances
    );
```

```
    void queryBalanceErr (
        in TpAssignmentID queryId,
        in TpBalanceQueryError cause
    );
```

```
    void retrieveTransactionHistoryRes (
```



```
    in TpAssignmentID retrievalID,  
    in TpTransactionHistorySet transactionHistory  
);
```

```
void retrieveTransactionHistoryErr (  
    in TpAssignmentID retrievalID,  
    in TpTransactionHistoryStatus transactionHistoryError  
);
```

```
void queryBalanceExpiryDateRes (  
    in TpAssignmentID queryId,  
    in TpBalanceExpiryDateSet  
);
```

```
void queryBalanceExpiryDateErr (  
    in TpAssignmentID queryId,  
    in TpBalanceQueryError cause  
);
```

```
void updateBalanceRes (  
    in TpAssignmentID requestId,  
    in TpBalance balance  
);
```

```
void updateBalanceErr (  
    in TpAssignmentID requestId,  
    in TpBalanceQueryError  
);
```

```
void createVoucherRes (  
    in TpAssignmentID requestId,  
    in TpAssignmentID voucherId  
);
```

```
void createVoucherErr (  
    in TpAssignmentID requestId,  
    in TpVoucherError cause  
);
```

```
void destroyVoucherRes (  
    in TpAssignmentID requestId,  
    in TpAssignmentID voucherId  
);
```

```
void destroyVoucherErr (  
    in TpAssignmentID requestId,  
    in TpAssignmentID voucherId,  
    in TpVoucherError cause  
);
```

```
void queryVoucherRes (  
    in TpAssignmentID queryId,  
    in TpVoucher voucher  
);
```

```
void queryVoucherErr (  
    in TpAssignmentID queryId,  
    in TpAssignmentID voucherId,  
    in TpVoucherError cause  
);
```

```
void queryUserVouchersRes (
```

```

        in TpAssignmentID queryId,
        in TpVoucherSet vouchers
    );

    void queryUserVouchersErr (
        in TpAssignmentID queryId,
        in TpVoucherError cause
    );
};

```

```

interface IpAccountManager : IpService {

    TpAssignmentID createNotification (
        in IpAppAccountManager appAccountManager,
        in TpChargingEventCriteria chargingEventCriteria
    )
        raises (TpCommonExceptions, P_INVALID_ADDRESS, P_INVALID_CRITERIA,
P_INVALID_EVENT_TYPE, P_UNKNOWN_SUBSCRIBER);

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

    TpAssignmentID queryBalanceReq (
        in TpAddressSet users
    )
        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, P_UNKNOWN_SUBSCRIBER,
P_INVALID_ADDRESS);

    TpChargingEventCriteriaResultSet getNotification ()
        raises (TpCommonExceptions);

    TpAssignmentID retrieveTransactionHistoryReq (
        in TpAddress user,
        in TpTimeInterval transactionInterval
    )
        raises (TpCommonExceptions, P_UNKNOWN_SUBSCRIBER,
P_UNAUTHORIZED_APPLICATION, P_INVALID_TIME_AND_DATE_FORMAT);

    TpAssignmentID enableNotifications (
        in IpAppAccountManager appAccountManager
    )
        raises (TpCommonExceptions);
};

```

```

void disableNotifications ()
    raises (TpCommonExceptions);

    TpAssignmentID queryBalanceExpiryDateReq (
        in TpAddressSet users
    )
    raises (TpCommonExceptions,
        P_UNKNOWN_SUBSCRIBER, P_UNAUTHORIZED_APPLICATION);

    TpAssignmentID updateBalanceReq (
        in TpAddress user,
        in TpBoolean debit,
        in TpBalanceInfo amount,
        in TpInt32 period
    )
    raises (TpCommonExceptions
        P_UNKNOWN_SUBSCRIBER, P_UNAUTHORIZED_APPLICATION);

    TpAssignmentID createVoucherReq (
        in TpAddress user,
        in TpBalanceInfo amount
    )
    raises (TpCommonExceptions
        P_UNKNOWN_SUBSCRIBER, P_UNAUTHORIZED_APPLICATION);

    TpAssignmentID queryVoucherReq (
        in TpAssignmentID voucherId
    )
    raises (TpCommonExceptions,
        P_UNAUTHORIZED_APPLICATION, P_INVALID_ASSIGNMENTID);

    TpAssignmentID destroyVoucherReq (
        in TpAssignmentID voucherId
    )
    raises (TpCommonExceptions
        P_UNAUTHORIZED_APPLICATION, P_INVALID_ASSIGNMENTID);

    TpAssignmentID queryUserVouchersReq (
        in TpAddress user
    )
    raises (TpCommonExceptions
        P_UNKNOWN_SUBSCRIBER, P_UNAUTHORIZED_APPLICATION,
        P_INVALID_ASSIGNMENTID);

};

};

};

};

#endif

```

<p>End of Change in Annex A.1 End of Document</p>
--