

**3GPP TSG CN Plenary Meeting #19**  
**12- 14 March 2003, Birmingham, UK**

**NP-030025**

**Source:** CN5 (OSA)  
**Title:** Rel-4 CR 29.198-11 OSA API Part 11: Account management  
**Agenda item:** 7.10  
**Document for:** APPROVAL

---

Doc-1st-Level	Spec	CR	Rev	Phase	Subject	Cat	Version-Current	Doc-2nd-Level	Workitem
NP-030025	29.198-11	015	-	Rel-4	Correction to TpChargingEventCriteria in Account Management	F	4.3.0	N5-021127	OSA1
NP-030025	29.198-11	016	-	Rel-5	Correction to TpChargingEventCriteria in Account Management	A	5.1.0	N5-021128	OSA2
NP-030025	29.198-11	017	-	Rel-4	Correction of status of methods to Account Management interfaces	F	4.3.0	N5-021146	OSA1
NP-030025	29.198-11	018	-	Rel-5	Addition of status of methods to Account Management interfaces	A	5.1.0	N5-021148	OSA2

## CHANGE REQUEST

⌘ **29.198-11 CR 015** ⌘ rev **-** ⌘ Current version: **4.3.0** ⌘

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

<b>Title:</b>	⌘ Correction to TpChargingEventCriteria in Account Management		
<b>Source:</b>	⌘ N5		
<b>Work item code:</b>	⌘ OSA1	<b>Date:</b>	⌘ 31/10/2002
<b>Category:</b>	⌘ <b>F</b>	<b>Release:</b>	⌘ REL-4
	Use <u>one</u> of the following categories: <b>F</b> (correction) <b>A</b> (corresponds to a correction in an earlier release) <b>B</b> (addition of feature), <b>C</b> (functional modification of feature) <b>D</b> (editorial modification) Detailed explanations of the above categories can be found in 3GPP <a href="#">TR 21.900</a> .		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)

<b>Reason for change:</b>	⌘ The text definition of TpChargingEventCriteria in TS 29.198-11 contains an error: The order of the two elements in this sequence is swapped compared with the definition in the IDL file.
<b>Summary of change:</b>	⌘ Change specification to match correct IDL description of TpChargingEventCriteria
<b>Consequences if not approved:</b>	⌘ A contradiction will exist between the IDL and the Word document. If no alignment is made, some developers will chose one version of the type, others the other, and interworking problems will arise.

<b>Clauses affected:</b>	⌘ 11.1.9						
<b>Other specs affected:</b>	<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 checked="" type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> </table> Other core specifications	Y	N	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	⌘	
Y	N						
<input checked="" 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 checked="" type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> </table> Test specifications	Y	N	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	⌘	
Y	N						
<input checked="" 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 checked="" type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> </table> O&M Specifications	Y	N	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	⌘	
Y	N						
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						
<b>Other comments:</b>	⌘						

### 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.

←===== MODIFIED SECTION =====>

### 11.1.9 TpChargingEventCriteria

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

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

←===== END MODIFIED SECTION =====>

---

## 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.

```
struct TpChargingEventCriteria {
    TpChargingEventNameSet ChargingEvents;
    TpAddressSet Users;
};
```

## CHANGE REQUEST

⌘ **29.198-11 CR 018** ⌘ rev **-** ⌘ Current version: **5.1.0** ⌘

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

**Proposed change affects:** UICC apps  ME  Radio Access Network  Core Network

<b>Title:</b>	⌘ Addition of status of methods to Account Management interfaces		
<b>Source:</b>	⌘ N5		
<b>Work item code:</b>	⌘ OSA2	<b>Date:</b>	⌘ 31/10/2002
<b>Category:</b>	⌘ <b>A</b>	<b>Release:</b>	⌘ REL-5
	Use <u>one</u> of the following categories: <b>F</b> (correction) <b>A</b> (corresponds to a correction in an earlier release) <b>B</b> (addition of feature), <b>C</b> (functional modification of feature) <b>D</b> (editorial modification) Detailed explanations of the above categories can be found in 3GPP <a href="#">TR 21.900</a> .		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)

<b>Reason for change:</b>	⌘ There is no requirement in the standard about the necessity to implement all or only some of the methods defined for an interface.
<b>Summary of change:</b>	⌘ Add a statement that clarifies which methods are mandatory and which are optional.
<b>Consequences if not approved:</b>	⌘ Application developers will not know which methods will actually be available.

<b>Clauses affected:</b>	⌘ 4, 8										
<b>Other specs affected:</b>	<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> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> </table>	Y	N	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Other core specifications Test specifications O&M Specifications	⌘
Y	N										
<input type="checkbox"/>	<input checked="" type="checkbox"/>										
<input type="checkbox"/>	<input checked="" type="checkbox"/>										
<input type="checkbox"/>	<input checked="" type="checkbox"/>										
<b>Other comments:</b>	⌘										

### 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.

---

## 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 SCF is implemented.
- The Class relationships clause show how each of the interfaces applicable to the SCF, relate to one another
- The Interface specification clause describes in detail each of the interfaces shown within the Class diagram part.
- The State Transition Diagrams (STD) show the transition between states in the SCF. The states and transitions are well-defined; either methods specified in the Interface specification or events occurring in the underlying networks cause state transitions.
- The Data definitions section show 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.

### 4.1 General requirements on support of methods

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

Where a method is not supported by an implementation of a Service interface, the exception P\_METHOD\_NOT\_SUPPORTED shall be returned to any call of that method.

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

## 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.

[This interface shall be implemented by an Account Management SCF.](#)

[The queryBalanceReq\(\) method, or the retrieveTransactionHistoryReq\(\) method, or both the createNotification\(\) and destroyNotification methods, or both the enableNotifications and disableNotifications methods shall be implemented as a minimum requirement.](#)

<<Interface>> IpAccountManager
<pre> 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 &lt;&lt;new&gt;&gt; enableNotifications (appAccountManager : in IpAppAccountManagerRef) : TpAssignmentID &lt;&lt;new&gt;&gt; disableNotifications () : void           </pre>

#### 8.1.1 Method createNotification()

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

If the same application requests two notifications with exactly the same criteria but different callback references, the second callback will be treated as an additional callback. Both notifications will share the same assignmentID. The gateway will always use the most recent callback. In case this most recent callback fails the second most recent is used. In case the enableCallNotification contains no callback, at the moment the application needs to be informed the gateway will use as callback the callback that has been registered by setCallback().

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

##### Parameters

**appAccountManager : in IpAppAccountManagerRef**

If this parameter is set (i.e. not NULL) it specifies a reference to the application interface, which is used for callbacks. If set to NULL, the application interface defaults to the interface specified via the setCallback() method.

**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\_ADDRESS, P\_INVALID\_CRITERIA,  
P\_INVALID\_EVENT\_TYPE, P\_UNKNOWN\_SUBSCRIBER****8.1.2 Method destroyNotification()**

This method is used by the application to disable charging notifications. This method only applies to notifications created with createNotification().

*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****8.1.3 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***TpAssignmentID***Raises***TpCommonExceptions, P\_UNKNOWN\_SUBSCRIBER, P\_UNAUTHORIZED\_APPLICATION**

### 8.1.4 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, P\_UNKNOWN\_SUBSCRIBER, P\_INVALID\_ADDRESS**

### 8.1.5 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*

**TpChargingEventCriteriaResultSet**

#### *Raises*

**TpCommonExceptions**

### 8.1.6 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***TpAssignmentID***Raises***TpCommonExceptions, P\_UNKNOWN\_SUBSCRIBER, P\_UNAUTHORIZED\_APPLICATION, P\_INVALID\_TIME\_AND\_DATE\_FORMAT**

### 8.1.7 Method <<new>> enableNotifications()

This method is used to indicate that the application is able to receive which are provisioned from within the network (i.e. these notifications are NOT set using createNotification() but via, for instance, a network management system). If notifications provisioned for this application are created or changed, the application is unaware of this until the notification is reported.

If the same application requests to enable notifications for a second time with a different IpAppAccountManager reference (i.e. without first disabling them), the second callback will be treated as an additional callback. This means that the callback will only be used in cases when the first callback specified by the application is unable to handle the reportNotification (e.g. due to overload or failure).

When this method is used, it is still possible to use createNotification() for service provider provisioned notifications on the same interface as long as the criteria in the network and provided by createNotification() do not overlap. However, it is NOT recommended to use both mechanisms on the same service manager.

The methods changeNotification(), getNotification(), and destroyNotification() do not apply to notifications provisioned in the network and enabled using enableNotifications(). These only apply to notifications created using createNotification().

Returns assignmentID: Specifies the ID assigned by the manager interface for this operation. This ID is contained in any reportNotification() that relates to notifications provisioned from within the network Repeated calls to enableNotifications() return the same assignment ID.

*Parameters***appAccountManager : in IpAppAccountManagerRef**

If this parameter is set (i.e. not NULL) it specifies a reference to the application interface, which is used for callbacks. If set to NULL, the application interface defaults to the interface specified via the setCallback() method.

*Returns***TpAssignmentID***Raises***TpCommonExceptions**

### 8.1.8 Method <<new>> disableNotifications()

This method is used to indicate that the application is not able to receive notifications for which the provisioning has been done from within the network. (i.e. these notifications that are NOT set using createNotification() but via, for instance, a network management system). After this method is called, no such notifications are reported anymore.

*Parameters*

No Parameters were identified for this method

*Raises*

**TpCommonExceptions**

## 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>> <b>IpAppAccountManager</b>
<pre> 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         </pre>

### 8.2.1 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.

### 8.2.2 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 TpAssignmentID**

Specifies the ID of the balance query request.

**balances : in TpBalanceSet**

Specifies the balance for one or more user accounts.

### 8.2.3 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 TpAssignmentID**

Specifies the ID of the balance query request.

**cause : in TpBalanceQueryError**

Specifies the error that led to the failure.

### 8.2.4 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 TpAssignmentID**

Specifies the retrievalID of the transaction history retrieval request.

**transactionHistory : in TpTransactionHistorySet**

Specifies the requested transaction history.

### 8.2.5 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 TpAssignmentID**

Specifies the retrievalID of the transaction history retrieval request.

**transactionHistoryError : in TpTransactionHistoryStatus**

Specifies the error that occurred while retrieving the transaction history.

## CHANGE REQUEST

⌘ **29.198-11 CR 017** ⌘ rev **-** ⌘ Current version: **4.3.0** ⌘

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

<b>Title:</b>	⌘ Correction of status of methods to Account Management interfaces		
<b>Source:</b>	⌘ N5		
<b>Work item code:</b>	⌘ OSA1	<b>Date:</b>	⌘ 31/10/2002
<b>Category:</b>	⌘ <b>F</b>	<b>Release:</b>	⌘ REL-4
	Use <u>one</u> of the following categories: <b>F</b> (correction) <b>A</b> (corresponds to a correction in an earlier release) <b>B</b> (addition of feature), <b>C</b> (functional modification of feature) <b>D</b> (editorial modification) Detailed explanations of the above categories can be found in 3GPP <a href="#">TR 21.900</a> .		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)

<b>Reason for change:</b>	⌘ There is no requirement in the standard about the necessity to implement all or only some of the methods defined for an interface.
<b>Summary of change:</b>	⌘ Add a statement that clarifies which methods are mandatory and which are optional.
<b>Consequences if not approved:</b>	⌘ Application developers will not know which methods will actually be available.

<b>Clauses affected:</b>	⌘ 4, 8										
<b>Other specs affected:</b>	<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> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> </table>	Y	N	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Other core specifications Test specifications O&M Specifications	⌘
Y	N										
<input type="checkbox"/>	<input checked="" type="checkbox"/>										
<input type="checkbox"/>	<input checked="" type="checkbox"/>										
<input type="checkbox"/>	<input checked="" type="checkbox"/>										
<b>Other comments:</b>	⌘										

### 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.

---

## 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 SCF is implemented.
- The Class relationships clause show how each of the interfaces applicable to the SCF, relate to one another
- The Interface specification clause describes in detail each of the interfaces shown within the Class diagram part.
- The State Transition Diagrams (STD) show the transition between states in the SCF. The states and transitions are well-defined; either methods specified in the Interface specification or events occurring in the underlying networks cause state transitions.
- The Data definitions section show 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.

### 4.1 General requirements on support of methods

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

Where a method is not supported by an implementation of a Service interface, the exception P\_METHOD\_NOT\_SUPPORTED shall be returned to any call of that method.

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

## 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.

[This interface shall be implemented by an Account Management SCF.](#)

[The queryBalanceReq\(\) method, or the retrieveTransactionHistoryReq\(\) method, or both the createNotification\(\) and destroyNotification methods shall be implemented as a minimum requirement.](#)

<<Interface>> IpAccountManager
<pre> createNotification (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         </pre>

#### *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\_ADDRESS, 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***TpAssignmentID***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, P\_UNKNOWN\_SUBSCRIBER, P\_INVALID\_ADDRESS***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***TpChargingEventCriteriaResultSet***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***TpAssignmentID***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) : 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

*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 TpAssignmentID**

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 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 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 TpAssignmentID**

Specifies the retrievalID of the transaction history retrieval request.

**transactionHistoryError : in TpTransactionHistoryStatus**

Specifies the error that occurred while retrieving the transaction history.

## CHANGE REQUEST

⌘ **29.198-11 CR 016** ⌘ rev **-** ⌘ Current version: **5.1.0** ⌘

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

**Proposed change affects:** UICC apps  ME  Radio Access Network  Core Network

<b>Title:</b>	⌘ Correction to TpChargingEventCriteria in Account Management		
<b>Source:</b>	⌘ N5		
<b>Work item code:</b>	⌘ OSA2	<b>Date:</b>	⌘ 31/10/2002
<b>Category:</b>	⌘ <b>A</b>	<b>Release:</b>	⌘ REL-5
	Use <u>one</u> of the following categories: <b>F</b> (correction) <b>A</b> (corresponds to a correction in an earlier release) <b>B</b> (addition of feature), <b>C</b> (functional modification of feature) <b>D</b> (editorial modification) Detailed explanations of the above categories can be found in 3GPP <a href="#">TR 21.900</a> .		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)

<b>Reason for change:</b>	⌘ The text and WSDL definition of TpChargingEventCriteria in TS 29.198-11 contains an error: The order of the two elements in this sequence is swapped compared with the definition in the IDL file.
<b>Summary of change:</b>	⌘ Change specification and WSDL to match correct IDL description of TpChargingEventCriteria
<b>Consequences if not approved:</b>	⌘ A contradiction will exist between the IDL and the Word document. If no alignment is made, some developers will chose one version of the type, others the other, and interworking problems will arise.

<b>Clauses affected:</b>	⌘ 11.1.9, Annex B						
<b>Other specs affected:</b>	<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"/>						
<b>Other comments:</b>	⌘						

### 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.

←===== MODIFIED SECTION =====→

### 11.1.9 TpChargingEventCriteria

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

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

←===== END MODIFIED SECTION =====→

---

## 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.

```
struct TpChargingEventCriteria {
    TpChargingEventNameSet ChargingEvents;
    TpAddressSet Users;
};
```

---

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

The W3C WSDL representation of this specification is contained in a text file (am.wsdl contained in archive 2919811WSDL.ZIP) which accompanies the present document.

```
<xsd:complexType name="TpChargingEventCriteria">
  <xsd:sequence>
    <xsd:element name="Users" type="osaxsd:TpAddressSet"/>
    <xsd:element name="ChargingEvents" type="amxsd:TpChargingEventNameSet"/>
    <xsd:element name="Users" type="osaxsd:TpAddressSet"/>
  </xsd:sequence>
</xsd:complexType>
```