3GPP TSG CN Plenary Meeting #13 **NP-010473**
Beijing, China, 19th –21st September 2001

Source: **CN5 (OSA)**

Title: **CRs 29.198-12 Rel-4**

Agenda item: **8.5**

Document for: **Approval**

Doc-1st-Level	Doc-2nd-Level	Spec	CR	Rev	Phase	Subject	Cat	Version-Current	Version-New	Meeting-2nd-Level	Workitem
NP-010473	N5-010683	29.198-12	001		Rel-4	Changing references to JAIN	F	4.0.0	4.1.0	N5-12	OSA1
NP-010473	N5-010557	29.198-12	002		Rel-4	Error corrections charging	F	4.0.0	4.1.0	N5-12	OSA1
NP-010473	N5-010558	29.198-12	003		Rel-4	Changed semantics of closeReservation parameter	F	4.0.0	4.1.0	N5-12	OSA1
NP-010473	N5-010559	29.198-12	004		Rel-4	Missing errors in definition of (credit/debit)(Amoun/Unit)Err	F	4.0.0	4.1.0	N5-12	OSA1
NP-010473	N5-010560	29.198-12	005		Rel-4	Clarification of Unit Reservation	F	4.0.0	4.1.0	N5-12	OSA1
NP-010473	N5-010667	29.198-12	006		Rel-4	Improving correlation request and response for applications	F	4.0.0	4.1.0	N5-12	OSA1
NP-010473	N5-010668	29.198-12	007		Rel-4	Remove the P_CHS_PARAM_RESULT value from the TpChargingParameterID type	F	4.0.0	4.1.0	N5-12	OSA1
NP-010473	N5-010711	29.198-12	008		Rel-4	Align the order of parameters for similar methods	F	4.0.0	4.1.0	N5-12	OSA1

3GPP TSG_CN5 (Open Service Access – OSA) **N5-010683**
Meeting #12, Sophia Antipolis, FRANCE, 16 – 19 July 2001

CR-Form-v4

CHANGE REQUEST

⌘ **29.198-12** **CR** **001** ⌘ ev **-** ⌘ Current version: **4.0.0** ⌘

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

Proposed change affects: ⌘ (U)SIM ☐ ME/UE ☐ Radio Access Network ☐ Core Network **X**

Title:	⌘	Changing references to JAIN
Source:	⌘	CN5
Work item code:	⌘	OSA1
Date:	⌘	30/08/2001
Category:	⌘	**F**
Release:	⌘	REL-4

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

Use one of the following releases:

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

Reason for change:	⌘	Incorrect references to JAIN.
Summary of change:	⌘	Correct references to the JAIN.
Consequences if not approved:	⌘	Potential legal ramifications

Clauses affected:	⌘	1		
Other specs affected:	⌘	**X** Other core specifications ☐ Test specifications ☐ O&M Specifications	⌘	All other parts of TS 29.198 Rel-4
Other comments:	⌘			

How to create CRs using this form:

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

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

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

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

1 Scope

The present document is Part 12 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 Charging Service Capability Feature (SCF) aspects of the interface. All aspects of the Charging 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 a number of JAIN™ Community member companies~~the JAIN consortium~~.

CR-Form-v3

CHANGE REQUEST

⌘ **29.198-12** **CR** **002** ⌘ rev **-** ⌘ Current version: **4.0.0** ⌘

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

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

Title:	⌘	Error corrections charging
Source:	⌘	CN5
Work item code:	⌘	OSA1
Date:	⌘	30/08/2001
Category:	⌘	**F**
Release:	⌘	REL-4

Use one of the following categories:
- ***F*** *(essential correction)*
- ***A*** *(corresponds to a correction in an earlier release)*
- ***B*** *(Addition of feature),*
- ***C*** *(Functional modification of feature)*
- ***D*** *(Editorial modification)*

Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

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

Reason for change:	⌘	Erroneous specification. Sequence diagram immediate charge incorrect and name of applicationDescription parameter in createChargingSession() is confusing.
Summary of change:	⌘	Use correct methods in sequence diagram and change name of applicationDescription parameter in createChargingSession().
Consequences if not approved:	⌘	Erroneous and unclear specification which may potentially lead to incorrect implementations of applications and/or the Charging Service.

Clauses affected:	⌘	4.2 and 7.1
Other specs affected:	⌘	☐ Other core specifications ⌘ ☐ Test specifications ☐ O&M Specifications
Other comments:	⌘	

Introduction

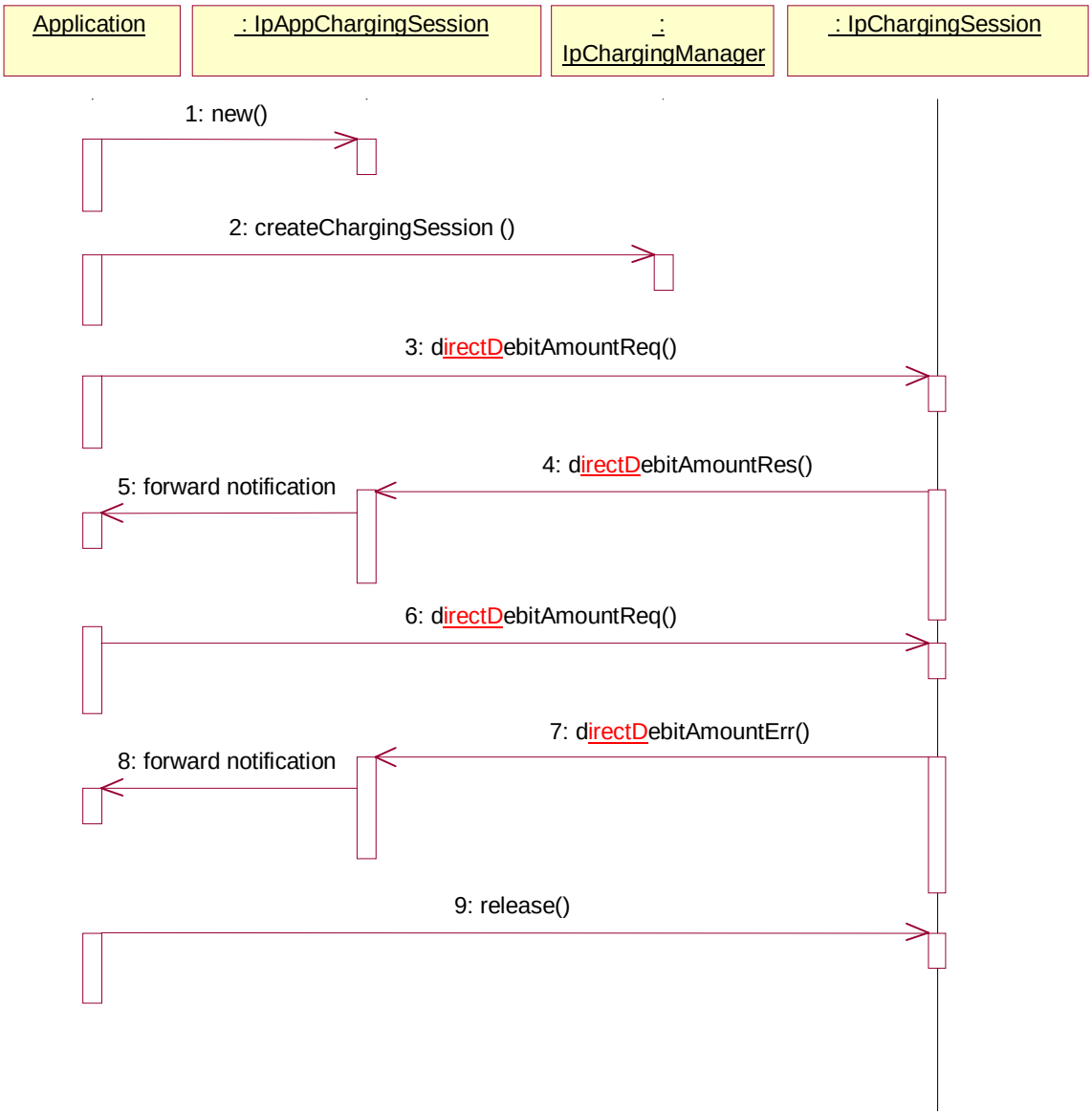
While reviewing v4.0.0 of the Charging Specification, Ericsson came accross the following issues:

- In the Charging API specification, section 4.2, the sequence diagram lists methods for debitAmountReq() and debitAmountRes(). These should however be replaced by directDebitAmoutReq() and directDebitAmountRes().
- During the meeting in San Diego, the type for the application description was changed from TpString to TpApplicationDescription (contribution N5-010441 from Siemens A.G.). This type can hold a set of descriptive items (such as the actual text and a timestamp). This type has been changed in the methods of the IpChargingSession class but not in the methods of the IpChargingManager class. Method `createChargingSession()` also has an `applicationDescription` parameter of type of `TpString`. Having two parameters with the same name and different type is misleading. In createChargingSession(), the parameter actually describes the session, and not the application as iis suggested.

Ericsson kindly requests the meeting to consider this change and approve it for inclusion into the Charging API.

Proposed Changes

See the proposed changes below (with revision marks).



<<Interface>> IpChargingManager
CreateChargingSession (appChargingSession : in IpAppChargingSession, ~~applicationDescription~~ sessionDescription : in TpString, merchantAccount : in TpMerchantAccountID, user : in TpAddress, correlationID : in TpCorrelationID, chargingSession : out TpChargingSessionID) : TpResult

Method

createChargingSession()

This method creates an instance of the IpChargingSession interface to handle the charging events related to the specified user and to the application invoking this method.

Parameters

appChargingSession : in IpAppChargingSession

Callback interface for the session in the application

~~applicationDescription~~ sessionDescription : in TpString

Descriptive text for informational purposes ~~(e.g. text presented on the bill and used in communication towards the user)~~.

merchantAccount : in TpMerchantAccountID

Identifies the account of the party providing the application to be used.

user : in TpAddress

Specifies the user that is using the application. This may or may not be the user that will be charged. The Charging service will determine the charged user. When this method is invoked the Charging service shall determine if charging is allowed for this application for this subscriber. An exception shall be thrown if this type of charging is not allowed.

correlationID : in TpCorrelationID

This value can be used to correlate the charging to network activity.

chargingSession : out TpChargingSessionID

Defines the session.

Raises

TpCommonExceptions,P_CHS_INVALID_USER,P_CHS_INVALID_ACCOUNT

CR-Form-v3

CHANGE REQUEST

⌘ **29.198-12** **CR** **003** ⌘ rev **-** ⌘ Current version: **4.0.0** ⌘

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

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

Title:	⌘	Changed semantics of closeReservation parameter
Source:	⌘	CN5
Work item code:	⌘	OSA1
Date:	⌘	30/08/2001
Category:	⌘	**F**
Release:	⌘	REL-4

Use one of the following categories:
- ***F*** *(essential correction)*
- ***A*** *(corresponds to a correction in an earlier release)*
- ***B*** *(Addition of feature),*
- ***C*** *(Functional modification of feature)*
- ***D*** *(Editorial modification)*

Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

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

Reason for change:	⌘	The semantics of the “closeReservation” parameter are incorrect an may lead to unnecessary abortion of a charging session.
Summary of change:	⌘	Change semantics of closeReservation parameter from (credit/debit)(Amoun/Unit)Req so that session is not released.
Consequences if not approved:	⌘	A service may not be offered to a user when there there is sufficient money on the account (and thus revenue loss for the operator).

Clauses affected:	⌘	8.3 and 9.1.
Other specs affected:	⌘	☐ Other core specifications ☐ Test specifications ☐ O&M Specifications
	⌘	
Other comments:	⌘	

Introduction

While reviewing v4.0.0 of the Charging Specification, Ericsson came accross the following issue:

- The methods creditAmountReq(), debitAmountReq(), creditUnitReq(), and debitUnitReq() from class IpChargingSession have a parameter closeReservation that indicates whether the reservation can be closed. According to the current specification, the reservation is freed AND the session is closed (released). This however causes a problem when the Charging Session was successfully released, but the result message was lost. The application will retry the operation, but the object has already been destructed. Ericsson proposes to change the semantics of this parameter in such a way that the reservation is freed, but the session is not released.

Ericsson kindly requests the meeting to consider this change and approve it for inclusion into the Charging API.

Proposed Changes

See the proposed changes below (with revision marks).

Method

creditAmountReq()

This method credits an amount towards the reservation associated with the session.

The amount left in the reservation will be increased by this amount.

Each request to debit / credit an amount towards a reservation is handled separately. For example, two requests for a payment of EUR 1,- will give a total payment of EUR 2,-.

A credit of EUR 1,- and a debit of EUR 1 will give a total payment of EUR 0,-.

Parameters

sessionID : in TpSessionID

The ID of the session.

applicationDescription : in TpApplicationDescription

Descriptive text for informational purposes (e.g. text presented on the bill and used in communication towards the user)

amount : in TpChargingPrice

The amount of specified currency to be credited towards the user.

closeReservation : in TpBoolean

If set to true, this parameter indicates that the remaining part of the reservation can be freed ~~and the session can be closed~~. This may also mean addition of currency to the subscriber's account if more credits than debits have been made. The session is not released, this has to be done explicitly by calling the release() method.

requestNumber : in TpInt32

Specifies the number given in the result of the previous operation on this session, or when creating the session. When no answer is received the same operation with the same parameters must be retried with the same requestNumber.

Raises

TpCommonExceptions,P_INVALID_SESSION_ID,P_INVALID_AMOUNT,P_INVALID_CURRENCY,P_INVALID_REQUEST_NUMBER

Method

debitAmountReq()

This method debits an amount from the reservation.

The amount left in the reservation will be decreased by this amount.

Each request to debit / credit an amount towards a reservation is handled separately. For example, two requests for a payment of EUR 1,- will give a total payment of EUR 2,-.

A credit of EUR 1,- and a debit of EUR 1 will give a total payment of EUR 0,-.

When a debit operation would exceed the limit of the reservation, the debit operation fails.

Parameters

sessionID : in TpSessionID

The ID of the session.

applicationDescription : in TpApplicationDescription

Descriptive text for informational purposes (e.g. text presented on the bill and used in communication towards the user)

amount : in TpChargingPrice

The amount of specified currency to be debited from the user.

closeReservation : in TpBoolean

If set to true, this parameter indicates that the reservation can be freed ~~and the session can be closed~~. The session is not released, this has to be done explicitly by calling the release() method.

requestNumber : in TpInt32

Specifies the number given in the result of the previous operation on this session, or when creating the session. When no answer is received the same operation with the same parameters must be retried with the same requestNumber.

Raises

TpCommonExceptions,P_INVALID_SESSION_ID,P_INVALID_AMOUNT,P_INVALID_CURRENCY,P_INVALID_REQUEST_NUMBER

Method

creditUnitReq()

This method credits a volume of application usage towards the reservation.

The volumes left in the reservation of this will be increased by this amount.

Each request to debit / credit a volume towards a reservation is handled separately. For example, two requests for a payment for 10 kilobytes will give a total payment for 20 kilobytes

Parameters

sessionID : in TpSessionID

The ID of the session.

applicationDescription : in TpApplicationDescription

Descriptive text for informational purposes (e.g. text presented on the bill and used in communication towards the user)

volumes : in TpVolumeSet

Specifies the credited volumes in different units, more specifically a sequence of data elements each containing the amount and applied unit.

closeReservation : in TpBoolean

If set to true, this parameter indicates that the reservation can be freed ~~and the session can be closed~~. The session is not released, this has to be done explicitly by calling the release() method.

requestNumber : in TpInt32

Specifies the number given in the result of the previous operation on this session, or when creating the session. When no answer is received the same operation with the same parameters must be retried with the same requestNumber.

Raises

TpCommonExceptions,P_INVALID_SESSION_ID,P_INVALID_VOLUME,P_INVALID_REQUEST_NUMBER

Method

debitUnitReq()

This method debits a volume of application usage from the reservation.

The volumes left in the reservation will be decreased by this amount.

Each request to debit / credit a volume towards a reservation is handled separately. For example, two requests for a payment for 10 kilobytes will give a total payment for 20 kilobytes.

When a debit operation would exceed the limit of the reservation, the debit operation succeeds, and the debited volumes will be the rest of the volumes in the reservation.

Parameters

sessionID : in TpSessionID

The ID of the session.

applicationDescription : in TpApplicationDescription

Descriptive text for informational purposes (e.g. text presented on the bill and used in communication towards the user)

volumes : in TpVolumeSet

Specifies the charged volumes in different units, more specifically a sequence of data elements each containing the amount and applied unit.

closeReservation : in TpBoolean

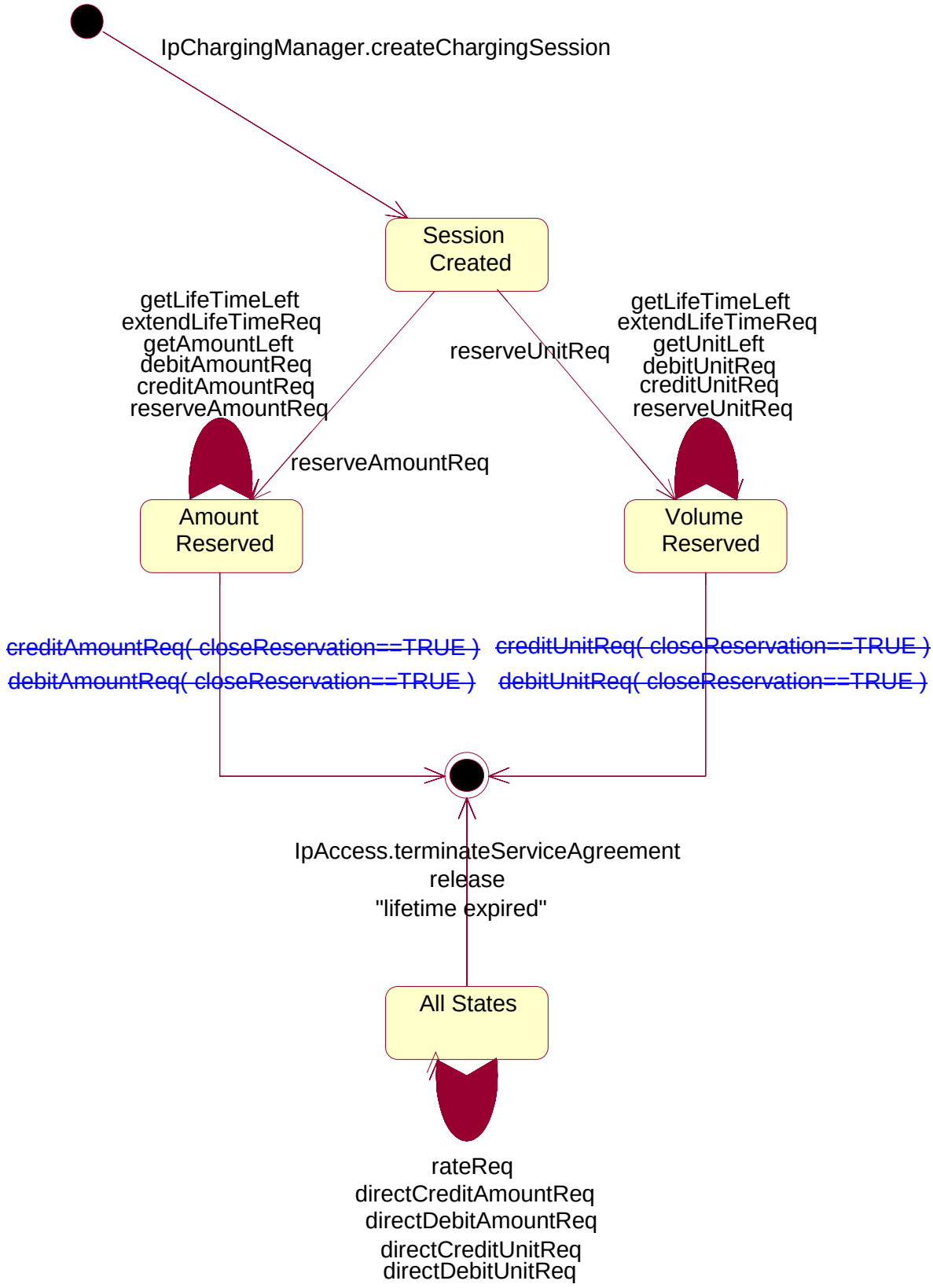
If set to true, this parameter indicates that the reservation can be freed ~~and the session can be closed~~. The session is not released, this has to be done explicitly by calling the release() method.

requestNumber : in TpInt32

Specifies the number given in the result of the previous operation on this session, or when creating the session. When no answer is received the same operation with the same parameters must be retried with the same requestNumber.

Raises

TpCommonExceptions,P_INVALID_SESSION_ID,P_INVALID_VOLUME,P_INVALID_REQUEST_NUMBER


IpChargingManager.createChargingSession
Session Created
getLifeTimeLeft
extendLifeTimeReq
getAmountLeft
debitAmountReq
creditAmountReq
reserveAmountReq
reserveUnitReq
getLifeTimeLeft
extendLifeTimeReq
getUnitLeft
debitUnitReq
creditUnitReq
reserveUnitReq
reserveAmountReq
Amount Reserved
Volume Reserved
creditAmountReq(closeReservation==TRUE)
debitAmountReq(closeReservation==TRUE)
creditUnitReq(closeReservation==TRUE)
debitUnitReq(closeReservation==TRUE)
IpAccess.terminateServiceAgreement
release
"lifetime expired"
All States
rateReq
directCreditAmountReq
directDebitAmountReq
directCreditUnitReq
directDebitUnitReq

CR-Form-v3

CHANGE REQUEST

⌘ **29.198-12** **CR** **004** ⌘ rev **-** ⌘ Current version: **4.0.0** ⌘

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

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

Title:	⌘	Missing errors in definition of (credit/debit)(Amoun/Unit)Err			
Source:	⌘	CN5			
Work item code:	⌘	OSA1	***Date:***	⌘	30/08/2001
Category:	⌘	**F**	***Release:***	⌘	REL-4

Use one of the following categories:
- ***F*** *(essential correction)*
- ***A*** *(corresponds to a correction in an earlier release)*
- ***B*** *(Addition of feature),*
- ***C*** *(Functional modification of feature)*
- ***D*** *(Editorial modification)*

Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

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

Reason for change:	⌘	Missing errors in definition of (credit/debit)(Amoun/Unit)Err.
Summary of change:	⌘	Additional errors to be returned by (credit/debit)(Amoun/Unit)Err.
Consequences if not approved:	⌘	Interoperatability problems when application is not aware of possible errors that can be returned.

Clauses affected:	⌘	8.2.		
Other specs affected:	⌘	☐ Other core specifications ☐ Test specifications ☐ O&M Specifications	⌘	
Other comments:	⌘			

.

Introduction

While reviewing v4.0.0 of the Charging Specification, Ericsson came accross the following issue:

- The methods debitAmountErr() and debitUnitErr() can also return P_CHS_ERR_RESERVATION_LIMIT if the bounds of the reservation are exceeded. The methods creditAmountErr() and creditUnitErr() can also return P_CHS_ERR_NO_CREDIT if creditting for a specific user is not allowed.

Ericsson kindly requests the meeting to consider this change and approve it for inclusion into the Charging API.

Proposed Changes

See the proposed changes below (with revision marks).

Method

creditAmountErr()

This method indicates that the corresponding request failed completely and that no money has been credited.

Parameters

sessionID : in TpSessionID

This is the ID of the session for which the operation was called.

error : in TpChargingError

Indicates the reason for failure. Possible errors are: P_CHS_ERR_CURRENCY, P_CHS_ERR_NO_CREDIT.

requestNumberNextRequest : in TpInt32

This request number must be used in the next request (requiring a Request Number) for this session.

Method

debitAmountErr()

This method indicates that the corresponding request failed completely and that no money has been debited.

Parameters

sessionID : in TpSessionID

This is the ID of the session for which the operation was called.

error : in TpChargingError

Indicates the reason for failure. Possible errors are: P_CHS_ERR_CURRENCY, P_CHS_ERR_RESERVATION_LIMIT

requestNumberNextRequest : in TpInt32

This request number must be used in the next request (requiring a Request Number) for this session.

Method

creditUnitErr()

This method indicates that the corresponding request failed completely and that no units have been credited.

Parameters

sessionID : in TpSessionID

This is the ID of the session for which the operation was called.

error : in TpChargingError

Indicates the reason for failure. Possible errors are: P_CHS_ERR_VOLUMES, P_CHS_ERR_NO_CREDIT

requestNumberNextRequest : in TpInt32

This request number must be used in the next request (requiring a Request Number) for this session.

Method

debitUnitErr()

This method indicates that the corresponding request failed completely and that no units have been debited.

Parameters

sessionID : in TpSessionID

This is the ID of the session for which the operation was called.

error : in TpChargingError

Indicates the reason for failure. Possible errors are: P_CHS_ERR_VOLUMES, P_CHS_ERR_RESERVATION_LIMIT

requestNumberNextRequest : in TpInt32

This request number must be used in the next request (requiring a Request Number) for this session.

CR-Form-v3

CHANGE REQUEST

⌘ **29.198-12** **CR** **005** ⌘ rev **-** ⌘ Current version: **4.0.0** ⌘

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

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

Title:	⌘	Clarification of Unit Reservation
Source:	⌘	CN5
Work item code:	⌘	OSA1
Date:	⌘	30/08/2001
Category:	⌘	**F**
Release:	⌘	REL-4

Use one of the following categories:
- ***F*** *(essential correction)*
- ***A*** *(corresponds to a correction in an earlier release)*
- ***B*** *(Addition of feature),*
- ***C*** *(Functional modification of feature)*
- ***D*** *(Editorial modification)*

Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

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

Reason for change:	⌘	It is unclear if units of different types are consolidated.
Summary of change:	⌘	State explicitly in description of reserveUnitReq() method that units of different types are NOT consolidated.
Consequences if not approved:	⌘	Interoperatibility problems when application expects consolidation of units and the Charging Service does not.

Clauses affected:	⌘	8.
Other specs Affected:	⌘	☐ Other core specifications ⌘ ☐ Test specifications ☐ O&M Specifications
Other comments:	⌘	

Introduction

While reviewing v4.0.0 of the Charging Specification, Ericsson came accross the following issue:

- The method reserveUnitReq() can be called multiple times to extend the reservation. The documentation of the method states that units of the same kind (for instance kbyte) are consolidated. This means that reserving 10 kbytes and another 10 kbytes results in a reservation of 20 kbytes. To prevent that the Charging Service has to know all the rules for consolidating units of different types, it is proposed to add to the specification that this is not the case. This means that if a reservation of 2 minutes is made and later on a reservation of 100 seconds, the total reservation holds 2 minutes and 100 seconds (and not 3 minutes and 40 seconds). The application in this case is responsible for using the same type of units in reservation and debit/credit methods.

Ericsson kindly requests the meeting to consider this change and approve it for inclusion into the Charging API.

Proposed Changes

See the proposed changes below (with revision marks).

8 Charging Interface classes

The Charging SCF is used by applications to charge for the usage of the applications. The charged user can be the same user as that uses the application. It is also possible that another user will pay the charge.

In the interfaces of the Charging SCF a "Request Number" is used when invoking operations that operate on the user's account (directly or indirectly via reservations) in order to make retries possible after application, service, or communication errors. A retry of these operations can be done by invoking the same operation with the same Request Number.

In the callback to the application, the Request Number to be used for the next request operation is returned. This is the only Request Number besides the one in the last request operation that can be used. This mechanism ensures that an application retries an operation when it does not receive an answer.

The units (of different types) that are used in a TpVolumeSet are NOT consolidated by the charging SCF. The application must use the same units when making the reservation and when debitting the amount. For example, when after a reservation of 10 minutes a debit request for 5 seconds is done, an error will be returned.

3GPP TSG_CN5 (Open Service Access – OSA) **N5-010667**
Meeting #12, Sophia Antipolis, FRANCE, 16 – 19 July 2001

CR-Form-v4

CHANGE REQUEST

⌘ **29.198-12** **CR** **006** ⌘ ev **-** ⌘ Current version: **4.0.0** ⌘

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

Proposed change affects: ⌘ (U)SIM ☐ ME/UE ☐ Radio Access Network ☐ Core Network **X**

Title:	⌘	Improving correlation request and response for applications
Source:	⌘	CN5
Work item code:	⌘	OSA1
Date:	⌘	30/08/2001
Category:	⌘	**F**
Release:	⌘	REL-4

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

Use one of the following releases:
2 (GSM Phase 2)
R96 (Release 1996)
R97 (Release 1997)
R98 (Release 1998)
R99 (Release 1999)
REL-4 (Release 4)
REL-5 (Release 5)

Reason for change:	⌘	The current standard makes it very difficult for an application to correlate the response to a specific request, especially if a response is delayed, and turns up several request later.
Summary of change:	⌘	Add the requestNumber to the result of all Asynchronous requests where there is a risk for over/under charging of the end-user.
Consequences if not approved:	⌘	It will be nearly impossible for applications to recognise what response belongs to what request of the same kind and session if responses get delayed. This causes interoperatibility problems between application and Charging service.

Clauses affected:	⌘	8, 8.1, 8.2
Other specs affected:	⌘	☐ Other core specifications ⌘ ☐ Test specifications ☐ O&M Specifications
Other comments:	⌘	

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

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

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

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

Introduction

The current standard makes it very difficult for an application to correlate the response to a specific request, especially if a response is delayed, and turns op several request later. (See Figure 1)

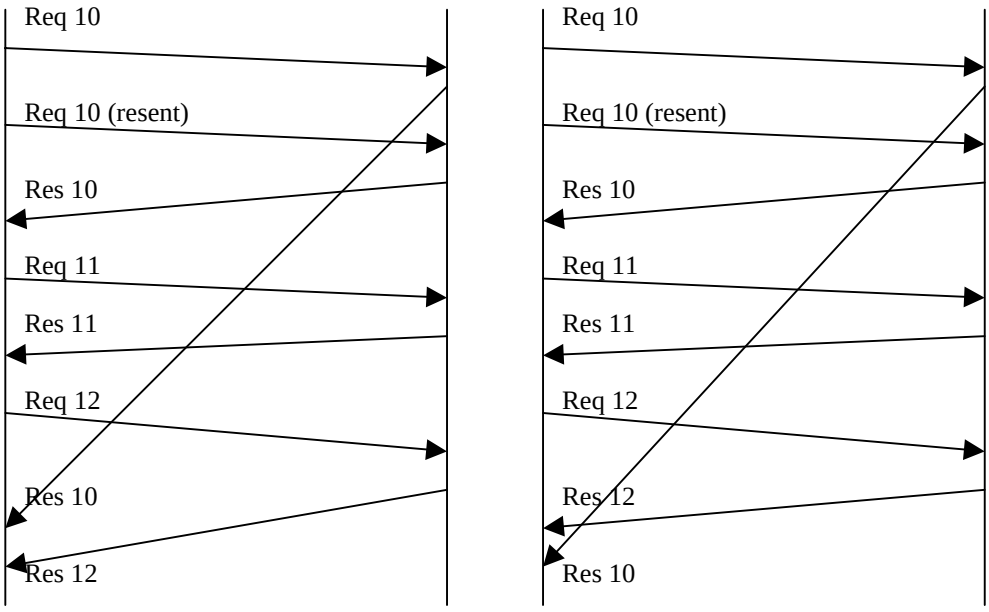


Figure 1

During a session a response may be lost, but as the responses have no indicator as to which request they belong it is not possible to correlate a response with a request, therefore it must be assumed that a response received belongs to the corresponding previous request.

When in the example of Figure 1 Request 10 and Request 12 are of the same type, e.g. both directDebitAmountReq, the application will probably assume it receives the response for request 12 in one of the two cases in figure 1, while in reality the application received the response for request 10.

Proposal

The service shall, in the response for each asynchronous request also provide the requestNumber in the response, this mechanism shall also be used for rateReq.

A more eleborate use of the request number shall be given.

At meeting #11 in San Diego Lucent proposed a different way of matching requests with responses as part of contribution N5-010330 (Proposal for a More Flexible Mechanism for Request Numbers). This proposal was rejected by the meeting.

8 Charging Interface Classes

The Charging SCF is used by applications to charge for the usage of the applications. The charged user can be the same user as that uses the application. It is also possible that another user will pay the charge.

In the interfaces of the Charging SCF a "Request Number" is used when invoking operations that operate on the user's account (directly or indirectly via reservations) in order to make retries possible after application, service, or communication errors. A retry of these operations can be done by invoking the same operation with the same Request Number.

In the callback to the application, the Request Number to be used for the next request operation is returned. This is the only Request Number besides the one in the last request operation that can be used. This mechanism ensures that an application retries an operation when it does not receive an answer.

The use of the Request Number causes that there can only be one outstanding request per Charging Session. Only after an answer is received (result or error), the next request can be made. Note however that only asynchronous operations that could lead to over or under charging of the user require a request number.

Because responses from the Charging SCF can be delayed in the network the Charging SCF shall guarantee that Request Numbers are unique in a timespan where delayed responsed can arrive. Suppose, for example, that the response from a retried request is received indicating the next request number to use is 1000. During the period that the response to the original request (which also carries the next request number to use equal to 1000) can arrive, this request number may not be used again.

8.1 Interface Class IpAppChargingSession

Inherits from: IpInterface.

This application interface must be implemented by the client application to handle callbacks from the IpChargingSession.

<<Interface>> IpAppChargingSession
ExtendLifeTimeRes (sessionID : in TpSessionID, SessionTimeLeft : in TpInt32) : TpResult ExtendLifeTimeErr (sessionID : in TpSessionID, error : in TpChargingError) : TpResult creditAmountRes (sessionID : in TpSessionID, requestNumber : In TpInt32, creditedAmount : in TpChargingPrice, reservedAmountLeft : in TpChargingPrice, requestNumberNextRequest : in TpInt32) : TpResult creditAmountErr (sessionID : in TpSessionID, requestNumber : In TpInt32, error : in TpChargingError, requestNumberNextRequest : in TpInt32) : TpResult debitAmountRes (sessionID : in TpSessionID, requestNumber : In TpInt32, debitedAmount : in TpChargingPrice, reservedAmountLeft : in TpChargingPrice, requestNumberNextRequest : in TpInt32) : TpResult debitAmountErr (sessionID : in TpSessionID, requestNumber : In TpInt32, error : in TpChargingError, requestNumberNextRequest : in TpInt32) : TpResult reserveAmountRes (sessionID : in TpSessionID, requestNumber : In TpInt32, reservedAmount : in TpChargingPrice, sessionTimeLeft : in TpInt32, requestNumberNextRequest : in TpInt32) : TpResult reserveAmountErr (sessionID : in TpSessionID, requestNumber : In TpInt32, error : in TpChargingError, requestNumberNextRequest : in TpInt32) : TpResult directCreditAmountRes (sessionID : in TpSessionID, requestNumber : In TpInt32, creditedAmount : in

TpChargingPrice, requestNumberNextRequest : in TpInt32) : TpResult

directCreditAmountErr (sessionID : in TpSessionID, requestNumber : In TpInt32, error : in TpChargingError, requestNumberNextRequest : in TpInt32) : TpResult

directDebitAmountRes (sessionID : in TpSessionID, requestNumber : In TpInt32, debitedAmount : in TpChargingPrice, requestNumberNextRequest : in TpInt32) : TpResult

directDebitAmountErr (sessionID : in TpSessionID, requestNumber : In TpInt32, error : in TpChargingError, requestNumberNextRequest : in TpInt32) : TpResult

creditUnitRes (sessionID : in TpSessionID, requestNumber : In TpInt32, creditedVolumes : in TpVolumeSet, reservedUnitsLeft : in TpVolumeSet, requestNumberNextRequest : in TpInt32) : TpResult

creditUnitErr (sessionID : in TpSessionID, requestNumber : In TpInt32, error : in TpChargingError, requestNumberNextRequest : in TpInt32) : TpResult

debitUnitRes (sessionID : in TpSessionID, requestNumber : In TpInt32, debitedVolumes : in TpVolumeSet, reservedUnitsLeft : in TpVolumeSet, requestNumberNextRequest : in TpInt32) : TpResult

debitUnitErr (sessionID : in TpSessionID, requestNumber : In TpInt32, error : in TpChargingError, requestNumberNextRequest : in TpInt32) : TpResult

reserveUnitRes (sessionID : in TpSessionID, requestNumber : In TpInt32, reservedUnits : in TpVolumeSet, sessionTimeLeft : in TpInt32, requestNumberNextRequest : in TpInt32) : TpResult

sessionEnded (sessionID : in TpSessionID, report : in TpSessionEndedCause) : TpResult

reserveUnitErr (sessionID : in TpSessionID, requestNumber : In TpInt32, error : in TpChargingError, requestNumberNextRequest : in TpInt32) : TpResult

rateRes (sessionID : in TpSessionID, rates : in TpPriceVolumeSet, validityTimeLeft : in TpDuration) : TpResult

rateErr (sessionID : in TpSessionID, error : in TpChargingError) : TpResult

directCreditUnitRes (sessionID : in TpSessionID, requestNumber : In TpInt32, creditedVolumes : in TpVolumeSet, requestNumberNextRequest : in TpInt32) : TpResult

directCreditUnitErr (sessionID : in TpSessionID, requestNumber : In TpInt32, error : in TpChargingError, requestNumberNextRequest : in TpInt32) : TpResult

directDebitUnitRes (sessionID : in TpSessionID, requestNumber : In TpInt32, debitedVolumes : in TpVolumeSet, requestNumberNextRequest : in TpInt32) : TpResult

directDebitUnitErr (sessionID : in TpSessionID, requestNumber : In TpInt32, error : in TpChargingError, requestNumberNextRequest : in TpInt32) : TpResult

Method

`extendLifeTimeRes()`

This method indicates that the corresponding request was successful.

Parameters

`sessionID : in TpSessionID`

This is the ID of the session for which the operation was called.

`SessionTimeLeft : in TpInt32`

Indicates the number of seconds that the session remains valid.

Method
extendLifeTimeErr()

This method indicates that the corresponding request failed.

Parameters

sessionID : in TpSessionID

This is the ID of the session for which the operation was called.

error : in TpChargingError

Indicates the reason for failure. Possible errors are: P_CHS_ERR_NO_EXTEND

Method
creditAmountRes()

This method indicates that the corresponding request was successful.

Parameters

sessionID : in TpSessionID

This is the ID of the session for which the operation was called.

requestNumber: In TpInt32

This is the request number for this request.

creditedAmount : in TpChargingPrice

Indicates the credited amount.

reservedAmountLeft : in TpChargingPrice

The amount left of the reservation.

requestNumberNextRequest : in TpInt32

This request number must be used in the next request (requiring a Request Number) for this session.

Method
creditAmountErr()

This method indicates that the corresponding request failed completely and that no money has been credited.

Parameters

sessionID : in TpSessionID

This is the ID of the session for which the operation was called.

requestNumber: In TpInt32

This is the request number for this request.

error : in TpChargingError

Indicates the reason for failure. Possible errors are: P_CHS_ERR_CURRENCY.

requestNumberNextRequest : in TpInt32

This request number must be used in the next request (requiring a Request Number) for this session.

Method

debitAmountRes()

This method indicates that the corresponding request was successful.

Parameters

sessionID : in TpSessionID

This is the ID of the session for which the operation was called.

requestNumber: In TpInt32

This is the request number for this request.

debitedAmount : in TpChargingPrice

Indicates the debited amount.

reservedAmountLeft : in TpChargingPrice

The amount left of the reservation.

requestNumberNextRequest : in TpInt32

This request number must be used in the next request (requiring a Request Number) for this session.

Method

debitAmountErr()

This method indicates that the corresponding request failed completely and that no money has been debited.

Parameters

sessionID : in TpSessionID

This is the ID of the session for which the operation was called.

requestNumber: In TpInt32

This is the request number for this request.

error : in TpChargingError

Indicates the reason for failure. Possible errors are: P_CHS_ERR_CURRENCY

requestNumberNextRequest : in TpInt32

This request number must be used in the next request (requiring a Request Number) for this session.

Method
reserveAmountRes()

This method indicates that the corresponding request was successful.

Parameters

sessionID : in TpSessionID

This is the same as the session ID returned in the request.

requestNumber: In TpInt32

This is the request number for this request.

reservedAmount : in TpChargingPrice

The amount reserved. If there was already a pending reservation, the sum of that and the new reservation is given.

sessionTimeLeft : in TpInt32

Indicates the number of seconds that the session and the reservation therein remains valid.

requestNumberNextRequest : in TpInt32

This request number must be used in the next request (requiring a Request Number) for this session.

Method
reserveAmountErr()

This method indicates that the corresponding request failed. The reservation cannot be used.

Parameters

sessionID : in TpSessionID

This is the same as the session ID returned in the request.

requestNumber: In TpInt32

This is the request number for this request.

error : in TpChargingError

Indicates the reason for failure. Possible errors are: P_CHS_ERR_PARAMETER, P_CHS_ERR_RESERVATION_LIMIT, P_CHS_ERR_CURRENCY, P_CHS_ERR_NO_EXTEND

requestNumberNextRequest : in TpInt32

This request number must be used in the next request (requiring a Request Number) for this session.

Method
directCreditAmountRes()

This method indicates that the corresponding request was successful.

Parameters

sessionID : in TpSessionID

This is the ID of the session for which the operation was called.

requestNumber: In TpInt32

This is the request number for this request.

creditedAmount : in TpChargingPrice

Indicates the credited amount.

requestNumberNextRequest : in TpInt32

This request number must be used in the next request (requiring a Request Number) for this session.

Method

directCreditAmountErr()

This method indicates that the corresponding request failed completely and that no money has been credited.

Parameters

sessionID : in TpSessionID

This is the ID of the session for which the operation was called.

requestNumber: In TpInt32

This is the request number for this request.

error : in TpChargingError

Indicates the reason for failure. Possible errors are: P_CHS_ERR_PARAMETER, P_CHS_ERR_NO_CREDIT, P_CHS_ERR_CURRENCY

requestNumberNextRequest : in TpInt32

This request number must be used in the next request (requiring a Request Number) for this session.

Method

directDebitAmountRes()

This method indicates that the corresponding request was successful.

Parameters

sessionID : in TpSessionID

This is the ID of the session for which the operation was called.

requestNumber: In TpInt32

This is the request number for this request.

debitedAmount : in TpChargingPrice

Indicates the debited amount.

requestNumberNextRequest : in TpInt32

This request number must be used in the next request (requiring a Request Number) for this session.

Method

directDebitAmountErr()

This method indicates that the corresponding request failed completely and that no money has been debited.

Parameters

sessionID : in TpSessionID

This is the ID of the session for which the operation was called.

requestNumber: In TpInt32

This is the request number for this request.

error : in TpChargingError

Indicates the reason for failure. Possible errors are: P_CHS_ERR_PARAMETER, P_CHS_ERR_NO_DEBIT, P_CHS_ERR_CURRENCY

requestNumberNextRequest : in TpInt32

This request number must be used in the next request (requiring a Request Number) for this session.

Method

creditUnitRes()

This method indicates that the corresponding request was successful.

Parameters

sessionID : in TpSessionID

This is the ID of the session for which the operation was called.

requestNumber: In TpInt32

This is the request number for this request.

creditedVolumes : in TpVolumeSet

Indicates the credited volumes of application usage.

reservedUnitsLeft : in TpVolumeSet

The volume of application usage left in the reservation.

requestNumberNextRequest : in TpInt32

This request number must be used in the next request (requiring a Request Number) for this session.

Method
creditUnitErr()

This method indicates that the corresponding request failed completely and that no units have been credited.

Parameters

sessionID : in TpSessionID

This is the ID of the session for which the operation was called.

requestNumber: In TpInt32

This is the request number for this request.

error : in TpChargingError

Indicates the reason for failure. Possible errors are: P_CHS_ERR_VOLUMES

requestNumberNextRequest : in TpInt32

This request number must be used in the next request (requiring a Request Number) for this session.

Method
debitUnitRes()

This method indicates that the corresponding request was successful.

Parameters

sessionID : in TpSessionID

This is the ID of the session for which the operation was called.

requestNumber: In TpInt32

This is the request number for this request.

debitedVolumes : in TpVolumeSet

Indicates the debited volumes of application usage.

reservedUnitsLeft : in TpVolumeSet

The volume of application usage left in the reservation.

requestNumberNextRequest : in TpInt32

This request number must be used in the next request (requiring a Request Number) for this session.

Method
debitUnitErr()

This method indicates that the corresponding request failed completely and that no units have been debited.

Parameters

sessionID : in TpSessionID

This is the ID of the session for which the operation was called.

requestNumber: In TpInt32

This is the request number for this request.

error : in TpChargingError

Indicates the reason for failure. Possible errors are: P_CHS_ERR_VOLUMES

requestNumberNextRequest : in TpInt32

This request number must be used in the next request (requiring a Request Number) for this session.

Method

reserveUnitRes()

This method indicates that the corresponding request was successful.

Parameters

sessionID : in TpSessionID

This is the same as the session ID returned in the request.

requestNumber: In TpInt32

This is the request number for this request.

reservedUnits : in TpVolumeSet

The volume of application usage reserved. If there was already a pending reservation, the sum of that and the new reservation is returned. E.g. a pending reservation of 25 charging units and a new reservation of 1000 octets and 10 charging units will result in two TpVolume elements for this parameter: 1000 octets and 35 charging units.

sessionTimeLeft : in TpInt32

Indicates the number of seconds that the session and the reservation therein remains valid.

requestNumberNextRequest : in TpInt32

This request number must be used in the next request (requiring a Request Number) for this session.

Method

sessionEnded()

This method indicates to the application that the charging session has terminated in the charging server. The application is expected to de-assign the charging session object after having received the sessionEnded.

Parameters

sessionID : in TpSessionID

Specifies the charging sessionID.

report : in TpSessionEndedCause

Specifies the cause the charging session is terminated.

Method

reserveUnitErr()

This method indicates that the corresponding request failed. The reservation cannot be used.

Parameters

sessionID : in TpSessionID

This is the same as the session ID returned in the request.

requestNumber: In TpInt32

This is the request number for this request.

error : in TpChargingError

Indicates the reason for failure. Possible errors are: P_CHS_ERR_PARAMETER, P_CHS_ERR_VOLUMES, P_CHS_ERR_RESERVATION_LIMIT, P_CHS_ERR_NO_EXTEND

requestNumberNextRequest : in TpInt32

This request number must be used in the next request (requiring a Request Number) for this session.

Method

rateRes()

This method indicates that the corresponding request was successful.

Parameters

sessionID : in TpSessionID

This is the ID of the session for which the operation was called.

rates : in TpPriceVolumeSet

The applicable rates.

validityTimeLeft : in TpDuration

Indicates the number of milli-seconds that this information remains valid.

Method

rateErr()

This method indicates that the corresponding request failed.

Parameters

sessionID : in TpSessionID

This is the ID of the session for which the operation was called.

error : in TpChargingError

Indicates the reason for failure. Possible errors are: P_CHS_ERR_PARAMETER

Method

directCreditUnitRes()

This method indicates that the corresponding request was successful.

Parameters

sessionID : in TpSessionID

This is the ID of the session for which the operation was called.

requestNumber: In TpInt32

This is the request number for this request.

creditedVolumes : in TpVolumeSet

Indicates the credited volumes of application usage.

requestNumberNextRequest : in TpInt32

This request number must be used in the next request (requiring a Request Number) for this session.

Method

directCreditUnitErr()

This method indicates that the corresponding request failed completely and that no units have been credited.

Parameters

sessionID : in TpSessionID

This is the ID of the session for which the operation was called.

requestNumber: In TpInt32

This is the request number for this request.

error : in TpChargingError

Indicates the reason for failure. Possible errors are: P_CHS_ERR_PARAMETER, P_CHS_ERR_NO_CREDIT, P_CHS_ERR_VOLUMES

requestNumberNextRequest : in TpInt32

This request number must be used in the next request (requiring a Request Number) for this session.

Method
directDebitUnitRes()

This method indicates that the corresponding request was successful.

Parameters

sessionID : in TpSessionID

This is the ID of the session for which the operation was called.

requestNumber: In TpInt32

This is the request number for this request.

debitedVolumes : in TpVolumeSet

Indicates the debited volumes of application usage.

requestNumberNextRequest : in TpInt32

This request number must be used in the next request (requiring a Request Number) for this session.

Method
directDebitUnitErr()

This method indicates that the corresponding request failed completely and that no units have been debited.

Parameters

sessionID : in TpSessionID

This is the ID of the session for which the operation was called.

requestNumber: In TpInt32

This is the request number for this request.

error : in TpChargingError

Indicates the reason for failure. Possible errors are: P_CHS_ERR_PARAMETER, P_CHS_ERR_NO_DEBIT, P_CHS_ERR_VOLUMES

requestNumberNextRequest : in TpInt32

This request number must be used in the next request (requiring a Request Number) for this session.

3GPP TSG_CN5 (Open Service Access – OSA) **N5-010668**
Meeting #12, Sophia Antipolis, FRANCE, 16 – 19 July 2001

CR-Form-v4

CHANGE REQUEST

29.198-12	CR	007	⌘	ev	-	⌘	Current version:	4.0.0	⌘

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

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

Title:	⌘	Remove the P_CHS_PARAM_RESULT value from the TpChargingParameterID type
Source:	⌘	CN5
Work item code:	⌘	OSA1
Date:	⌘	30/08/2001
Category:	⌘	**F**
Release:	⌘	REL-4

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

Use one of the following releases:

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

Reason for change:	⌘	The Value P_CHS_PARAM_RESULT cannot be used, because the service result cannot be known at times when the parent data type (TpChargingParameterSet) is used.
Summary of change:	⌘	The value should be removed from the enumerated type.
Consequences if not approved:	⌘	- Definition of parameters that cannot be used is confusing. - Divergence between ETSI and Parlay specification on one hand and 3GPP 29.198-12 on the other hand.

Clauses affected:	⌘	10.1		
Other specs affected:	⌘	☐ Other core specifications ☐ Test specifications ☐ O&M Specifications	⌘	
Other comments:	⌘			

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

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

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

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

8.3 Interface Class IpChargingSession

TpChargingParameterSet

Defines a Numbered Set of Data Elements of TpChargingParameter

TpChargingParameter

Defines a Sequence of Data Elements that defines the used service.

Sequence Element Name	Sequence Element Type
ParameterID	TpChargingParameterID
ParameterValue	TpChargingParameterValue

TpChargingParameterID

Defines the type of charging parameter. This type can be extended with operator specific items.

Name	Value	Description
P_CHS_PARAM_UNDEFINED	0	Unknown parameter
P_CHS_PARAM_ITEM	1	Parameter represents kind of service delivered to the end user
P_CHS_PARAM_SUBTYPE	2	Parameter represents subtype / operation of service delivered to the end user
~~P_CHS_PARAM_RESULT~~	~~3~~	~~Parameter represents the result of the service~~

TpChargingParameterValue

Defines the Tagged Choice of Data Elements that identify a charging parameter.

	Tag Element Type	
	TpChargingParameterValueType	

Tag Element Value	Choice Element Type	Choice Element Name
P_CHS_PARAMETER_INT32	TpInt32	IntValue
P_CHS_PARAMETER_FLOAT	TpFloat	FloatValue
P_CHS_PARAMETER_STRING	TpString	StringValue
P_CHS_PARAMETER_BOOLEAN	TpBoolean	BooleanValue

TpChargingParameterValueType

Defines the type of charging parameter.

Name	Value	Description
P_CHS_PARAMETER_INT32	0	Parameter represented by a TpInt32
P_CHS_PARAMETER_FLOAT	1	Parameter represented by a TpFloat
P_CHS_PARAMETER_STRING	2	Parameter represented by a TpString
P_CHS_PARAMETER_BOOLEAN	3	Parameter represented by a TpBoolean

3GPP TSG_CN5 (Open Service Access – OSA) **N5-010711**
Meeting #12, Sophia Antipolis, FRANCE, 16 – 19 July 2001

CR-Form-v4

CHANGE REQUEST

29.198-12	CR	008	⌘	ev	-	⌘	Current version:	4.0.0	⌘

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

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

Title:	⌘	Align the order of parameters for similar methods
Source:	⌘	CN5
Work item code:	⌘	OSA1
Date:	⌘	30/08/2001
Category:	⌘	**F**
Release:	⌘	REL-4

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

Use one of the following releases:
2 (GSM Phase 2)
R96 (Release 1996)
R97 (Release 1997)
R98 (Release 1998)
R99 (Release 1999)
REL-4 (Release 4)
REL-5 (Release 5)

Reason for change:	⌘	- Inconsistent order of parameters in similar methods - Similar changes already agreed by ETSI & Parlay
Summary of change:	⌘	Order of parameters of IpChargingSession.reserveUnitReq() and IpChargingSession.reserveAmountReq() shall be changed.
Consequences if not approved:	⌘	- Difficult use of the API - Divergence between ETSI and Parlay specification on one hand and 3GPP 29.198-12 on the other hand.

Clauses affected:	⌘	8.3
Other specs affected:	⌘	☐ Other core specifications ⌘ ☐ Test specifications ☐ O&M Specifications
Other comments:	⌘	

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

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

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

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

8.3 Interface Class IpChargingSession

Inherits from: IpService.
The Charging Session interface provides operations to facilitate transactions between a merchant and a user. The application programmer can use this interface to debit or credit amounts and/or units towards a user, to create and extend the lifetime of a reservation and to get information about what is left of the reservation.

<<Interface>>

IpChargingSession

~~ReserveAmountReq (sessionID : in TpSessionID, preferredAmount : in TpChargingPrice, minimumAmount : in TpChargingPrice, applicationDescription : in TpApplicationDescription, requestNumber : in TpInt32, chargingParameters : in TpChargingParameterSet) : TpResult~~

ReserveAmountReq (sessionID : in TpSessionID, applicationDescription : in TpApplicationDescription, chargingParameters : in TpChargingParameterSet, preferredAmount : in TpChargingPrice, minimumAmount : in TpChargingPrice, requestNumber : in TpInt32) : TpResult

creditAmountReq (sessionID : in TpSessionID, applicationDescription : in TpApplicationDescription, amount : in TpChargingPrice, closeReservation : in TpBoolean, requestNumber : in TpInt32) : TpResult

debitAmountReq (sessionID : in TpSessionID, applicationDescription : in TpApplicationDescription, amount : in TpChargingPrice, closeReservation : in TpBoolean, requestNumber : in TpInt32) : TpResult

getAmountLeft (sessionID : in TpSessionID, amountLeft : out TpChargingPriceRef) : TpResult

release (sessionID : in TpSessionID, requestNumber : in TpInt32) : TpResult

extendLifeTimeReq (sessionID : in TpSessionID) : TpResult

getLifeTimeLeft (sessionID : in TpSessionID, reservationTimeLeft : out TpInt32Ref) : TpResult

directCreditAmountReq (sessionID : in TpSessionID, applicationDescription : in TpApplicationDescription, chargingParameters : in TpChargingParameterSet, amount : in TpChargingPrice, requestNumber : in TpInt32) : TpResult

directDebitAmountReq (sessionID : in TpSessionID, applicationDescription : in TpApplicationDescription, chargingParameters : in TpChargingParameterSet, amount : in TpChargingPrice, requestNumber : in TpInt32) : TpResult

~~reserveUnitReq (sessionID : in TpSessionID, chargingParameters : in TpChargingParameterSet, volumes : in TpVolumeSet, applicationDescription : in TpApplicationDescription, requestNumber : in TpInt32) : TpResult~~

reserveUnitReq (sessionID : in TpSessionID, applicationDescription : in TpApplicationDescription, chargingParameters : in TpChargingParameterSet, volumes : in TpVolumeSet, requestNumber : in TpInt32) : TpResult

creditUnitReq (sessionID : in TpSessionID, applicationDescription : in TpApplicationDescription, volumes : in TpVolumeSet, closeReservation : in TpBoolean, requestNumber : in TpInt32) : TpResult

debitUnitReq (sessionID : in TpSessionID, applicationDescription : in TpApplicationDescription, volumes : in TpVolumeSet, closeReservation : in TpBoolean, requestNumber : in TpInt32) : TpResult

getUnitLeft (sessionID : in TpSessionID, volumesLeft : out TpVolumeSetRef) : TpResult

rateReq (sessionID : in TpSessionID, chargingParameters : in TpChargingParameterSet) : TpResult

directCreditUnitReq (sessionID : in TpSessionID, applicationDescription : in TpApplicationDescription, chargingParameters : in TpChargingParameterSet, volumes : in TpVolumeSet, requestNumber : in TpInt32) : TpResult

directDebitUnitReq (sessionID : in TpSessionID, applicationDescription : in TpApplicationDescription, chargingParameters : in TpChargingParameterSet, volumes : in TpVolumeSet, requestNumber : in TpInt32) : TpResult

Method

reserveAmountReq()

This method is used when an application wants to reserve an amount of money for services to be delivered to a user. It is also possible to enlarge the existing amount reservation by invoking this method. If a reservation is extended, the lifetime of the reservation is re-initialized.

Parameters

sessionID : in TpSessionID

The ID of the session.

applicationDescription : in TpApplicationDescription

Descriptive text for informational purposes (e.g. text presented on the bill and used in communication towards the user)

chargingParameters : in TpChargingParameterSet

These parameters and their values specify to the charging service what was provided to the end user so that the charging service can determine the applicable tariff.

preferredAmount : in TpChargingPrice

The amount of specified currency that the application wants to be reserved.

minimumAmount : in TpChargingPrice

The minimum amount that can be used by the application if the preferred amount cannot be granted.

~~**applicationDescription : in TpApplicationDescription**~~

~~Descriptive text for informational purposes (e.g. text presented on the bill and used in communication towards the user)~~

requestNumber : in TpInt32

Specifies the number given in the result of the previous operation on this session, or when creating the session. When no answer is received the same operation with the same parameters must be retried with the same requestNumber.

~~**chargingParameters : in TpChargingParameterSet**~~

~~These parameters and their values specify to the charging service what was provided to the end user so that the charging service can determine the applicable tariff.~~

Raises

TpCommonExceptions,P_INVALID_SESSION_ID,P_INVALID_AMOUNT,P_INVALID_CURRENCY,P_INVALID_REQUEST_NUMBER

Method

creditAmountReq()

This method credits an amount towards the reservation associated with the session.
The amount left in the reservation will be increased by this amount.
Each request to debit / credit an amount towards a reservation is handled separately. For example, two requests for a payment of EUR 1,- will give a total payment of EUR 2,-.
A credit of EUR 1,- and a debit of EUR 1 will give a total payment of EUR 0,-.

Parameters

sessionID : in TpSessionID

The ID of the session.

applicationDescription : in TpApplicationDescription

Descriptive text for informational purposes (e.g. text presented on the bill and used in communication towards the user)

amount : in TpChargingPrice

The amount of specified currency to be credited towards the user.

closeReservation : in TpBoolean

If set to true, this parameter indicates that the remaining part of the reservation can be freed and the session can be closed. This may also mean addition of currency to the subscriber's account if more credits than debits have been made.

requestNumber : in TpInt32

Specifies the number given in the result of the previous operation on this session, or when creating the session. When no answer is received the same operation with the same parameters must be retried with the same requestNumber.

Raises

TpCommonExceptions,P_INVALID_SESSION_ID,P_INVALID_AMOUNT,P_INVALID_CURRENCY,P_INVALID_REQUEST_NUMBER

Method

debitAmountReq()

This method debits an amount from the reservation.
The amount left in the reservation will be decreased by this amount.
Each request to debit / credit an amount towards a reservation is handled separately. For example, two requests for a payment of EUR 1,- will give a total payment of EUR 2,-.
A credit of EUR 1,- and a debit of EUR 1 will give a total payment of EUR 0,-.
When a debit operation would exceed the limit of the reservation, the debit operation fails.

Parameters

sessionID : in TpSessionID

The ID of the session.

applicationDescription : in TpApplicationDescription

Descriptive text for informational purposes (e.g. text presented on the bill and used in communication towards the user)

amount : in TpChargingPrice

The amount of specified currency to be debited from the user.

closeReservation : in TpBoolean

If set to true, this parameter indicates that the reservation can be freed and the session can be closed.

requestNumber : in TpInt32

Specifies the number given in the result of the previous operation on this session, or when creating the session. When no answer is received the same operation with the same parameters must be retried with the same requestNumber.

Raises

TpCommonExceptions,P_INVALID_SESSION_ID,P_INVALID_AMOUNT,P_INVALID_CURRENCY,P_INVALID_REQUEST_NUMBER

Method

getAmountLeft()

With this method an application can request the remaining amount of the reservation.

Parameters

sessionID : in TpSessionID

The ID of the session.

amountLeft : out TpChargingPriceRef

Gives the amount left in the reservation.

Raises

TpCommonExceptions,P_INVALID_SESSION_ID

Method

release()

This method releases the session, no operations can be done towards this session anymore (not even retries). Unused parts of a reservation are freed.

Parameters

sessionID : in TpSessionID

The ID of the session.

requestNumber : in TpInt32

Specifies the number given in the result of the previous operation on this session, or when creating the session.

Raises

TpCommonExceptions,P_INVALID_SESSION_ID,P_INVALID_REQUEST_NUMBER

Method

extendLifeTimeReq()

With this method an application can request the lifetime of the reservation to be extended. If no reservation has been made on the charging session, this method raises an exception (P_TASK_REFUSED).

Parameters

sessionID : in TpSessionID

The ID of the session.

Raises

TpCommonExceptions,P_INVALID_SESSION_ID

Method

getLifeTimeLeft()

With this method an application can request the remaining lifetime of the reservation. If no reservation has been made on the charging session, this method raises an exception (P_TASK_REFUSED).

Parameters

sessionID : in TpSessionID

The ID of the session.

reservationTimeLeft : out TpInt32Ref

Indicates the number of seconds that the session remains valid.

Raises

TpCommonExceptions,P_INVALID_SESSION_ID

Method

directCreditAmountReq()

This method directly credits an amount towards the user.
A possible reservation associated with this session is not influenced.

Parameters

sessionID : in TpSessionID

The ID of the session.

applicationDescription : in TpApplicationDescription

Descriptive text for informational purposes (e.g. text presented on the bill and used in communication towards the user)

chargingParameters : in TpChargingParameterSet

These parameters and their values specify to the charging service what was provided to the end user so that the charging service can determine the applicable tariff..

amount : in TpChargingPrice

The amount of specified currency to be credited towards the user.

requestNumber : in TpInt32

Specifies the number given in the result of the previous operation on this session, or when creating the session. When no answer is received the same operation with the same parameters must be retried with the same requestNumber.

Raises

TpCommonExceptions,P_INVALID_SESSION_ID,P_INVALID_AMOUNT,P_INVALID_CURRENCY,P_INVALID_REQUEST_NUMBER

Method

directDebitAmountReq()

This method directly debits an amount towards the user.
A possible reservation associated with this session is not influenced.

Parameters

sessionID : in TpSessionID

The ID of the session.

applicationDescription : in TpApplicationDescription

Descriptive text for informational purposes (e.g. text presented on the bill and used in communication towards the user)

chargingParameters : in TpChargingParameterSet

These parameters and their values specify to the charging service what was provided to the end user so that the charging service can determine the applicable tariff..

amount : in TpChargingPrice

The amount of specified currency to be debited from the user.

requestNumber : in TpInt32

Specifies the number given in the result of the previous operation on this session, or when creating the session. When no answer is received the same operation with the same parameters must be retried with the same requestNumber.

Raises

TpCommonExceptions,P_INVALID_SESSION_ID,P_INVALID_AMOUNT,P_INVALID_CURRENCY,P_INVALID_REQUEST_NUMBER

Method

reserveUnitReq()

This method is used when an application wants to reserve volumes of application usage to be delivered to a user in the session. When using units it is assumed that the price setting for the units is handled by the network side services. It is also possible to enlarge the existing unit reservation by invoking this method.

Parameters

sessionID : in TpSessionID

The ID of the session.

applicationDescription : in TpApplicationDescription

Descriptive text for informational purposes (e.g. text presented on the bill and used in communication towards the user)

chargingParameters : in TpChargingParameterSet

These parameters and their values specify to the charging service what was provided to the end user so that the charging service can determine the applicable tariff..

volumes : in TpVolumeSet

Specifies the reserved volumes in different units, more specifically a sequence of data elements each containing the amount and applied unit. It is e.g. possible to make a reservation for 10000 octets and 5 charging units.

~~**applicationDescription : in TpApplicationDescription**~~

~~Descriptive text for informational purposes (e.g. text presented on the bill and used in communication towards the user)~~

requestNumber : in TpInt32

Specifies the number given in the result of the previous operation on this session, or when creating the session. When no answer is received the same operation with the same parameters must be retried with the same requestNumber.

Raises

TpCommonExceptions,P_INVALID_SESSION_ID,P_INVALID_VOLUME,P_INVALID_REQUEST_NUMBER

Method

creditUnitReq()

This method credits a volume of application usage towards the reservation.
The volumes left in the reservation of this will be increased by this amount.
Each request to debit / credit a volume towards a reservation is handled separately. For example, two requests for a payment for 10 kilobytes will give a total payment for 20 kilobytes

Parameters

sessionID : in TpSessionID

The ID of the session.

applicationDescription : in TpApplicationDescription

Descriptive text for informational purposes (e.g. text presented on the bill and used in communication towards the user)

volumes : in TpVolumeSet

Specifies the credited volumes in different units, more specifically a sequence of data elements each containing the amount and applied unit.

closeReservation : in TpBoolean

If set to true, this parameter indicates that the reservation can be freed and the session can be closed.

requestNumber : in TpInt32

Specifies the number given in the result of the previous operation on this session, or when creating the session. When no answer is received the same operation with the same parameters must be retried with the same requestNumber.

Raises

TpCommonExceptions,P_INVALID_SESSION_ID,P_INVALID_VOLUME,P_INVALID_REQUEST_NUMBER

Method

debitUnitReq()

This method debits a volume of application usage from the reservation.
The volumes left in the reservation will be decreased by this amount.
Each request to debit / credit a volume towards a reservation is handled separately. For example, two requests for a payment for 10 kilobytes will give a total payment for 20 kilobytes.
When a debit operation would exceed the limit of the reservation, the debit operation succeeds, and the debited volumes will be the rest of the volumes in the reservation.

Parameters

sessionID : in TpSessionID

The ID of the session.

applicationDescription : in TpApplicationDescription

Descriptive text for informational purposes (e.g. text presented on the bill and used in communication towards the user)

volumes : in TpVolumeSet

Specifies the charged volumes in different units, more specifically a sequence of data elements each containing the amount and applied unit.

closeReservation : in TpBoolean

If set to true, this parameter indicates that the reservation can be freed and the session can be closed.

requestNumber : in TpInt32

Specifies the number given in the result of the previous operation on this session, or when creating the session. When no answer is received the same operation with the same parameters must be retried with the same requestNumber.

Raises

TpCommonExceptions,P_INVALID_SESSION_ID,P_INVALID_VOLUME,P_INVALID_REQUEST_NUMBER

Method

getUnitLeft()

With this method an application can request the remaining amount of the reservation.

Parameters

sessionID : in TpSessionID

The ID of the session.

volumesLeft : out TpVolumeSetRef

Specifies the remaining volumes in different units, more specifically a sequence of data elements each containing the amount and applied unit.

Raises

TpCommonExceptions, P_INVALID_SESSION_ID

Method

rateReq()

This method is used when the application wants to have an item rated by the charging service. The result can be used to present pricing information to the end user before the end user actually want to start using the service.

Parameters

sessionID : in TpSessionID

The ID of the session.

chargingParameters : in TpChargingParameterSet

These parameters and their values specify to the charging service what was provided to the end user so that the charging service can determine the applicable tariff..

Raises

TpCommonExceptions,P_INVALID_SESSION_ID

Method

directCreditUnitReq()

This method directly credits a volume of application usage towards the user.
The volumes in a possible reservation associated with this session are not influenced.

Parameters

sessionID : in TpSessionID

The ID of the reservation.

applicationDescription : in TpApplicationDescription

Descriptive text for informational purposes (e.g. text presented on the bill and used in communication towards the user)

chargingParameters : in TpChargingParameterSet

These parameters and their values specify to the charging service what was provided to the end user so that the charging service can determine the applicable tariff..

volumes : in TpVolumeSet

Specifies the credited volumes in different units, more specifically a sequence of data elements each containing the amount and applied unit.

requestNumber : in TpInt32

Specifies the number given in the result of the previous operation on this session, or when creating the session. When no answer is received the same operation with the same parameters must be retried with the same requestNumber.

Raises

TpCommonExceptions,P_INVALID_SESSION_ID,P_INVALID_VOLUME,P_INVALID_REQUEST_NUMBER

Method

directDebitUnitReq()

This method directly credits a volume of application usage towards the user.
The volumes in a possible reservation associated with this session are not influence.

Parameters

sessionID : in TpSessionID

The ID of the reservation.

applicationDescription : in TpApplicationDescription

Descriptive text for informational purposes (e.g. text presented on the bill and used in communication towards the user)

chargingParameters : in TpChargingParameterSet

These parameters and their values specify to the charging service what was provided to the end user so that the charging service can determine the applicable tariff..

volumes : in TpVolumeSet

Specifies the charged volumes in different units, more specifically a sequence of data elements each containing the amount and applied unit.

requestNumber : in TpInt32

Specifies the number given in the result of the previous operation on this session, or when creating the session. When no answer is received the same operation with the same parameters must be retried with the same requestNumber.

Raises

TpCommonExceptions,P_INVALID_SESSION_ID,P_INVALID_VOLUME,P_INVALID_REQUEST_NUMBER