

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

NP-010603

Source: CN5 (OSA)
Title: Rel-4 CRs 29.198-12
Agenda item: 8.5
Document for: Decision

Doc-1st-Level	Spec	CR	Pha	Subject	Cat	Ver Cur	Ver -New	Doc-2nd-Level	Workitem
NP-010603	29.198-12	009	Rel-4	Replace Out Parameters with Return Types	F	4.1.0	4.2.0	N5-010570	OSA1

CHANGE REQUEST

⌘ **29.198-12 CR 009** ⌘ ev **-** ⌘ Current version: **4.1.0** ⌘

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

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

Title:	⌘ Replacing Out Parameters with Return Types in OSA R4		
Source:	⌘ CN5		
Work item code:	⌘ OSA1	Date:	⌘ 19/07/2001
Category:	⌘ F	Release:	⌘ REL-4
	<i>Use one of the following categories:</i> 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 .		<i>Use one of the following releases:</i> 2 (GSM Phase 2) R96 (Release 1996) R97 (Release 1997) R98 (Release 1998) R99 (Release 1999) REL-4 (Release 4) REL-5 (Release 5)

Reason for change:	⌘ At CN5 and CN it was agreed that Out-parameters should be removed from methods as a means of returning information, to be replaced by Return Types, in line with commonly used programming practice
Summary of change:	⌘ For each method, replace the return parameter TpResult with: 'void' if the method has no out-parameter; or the type of the out-parameter if the method has an out-parameter, and delete the out-parameter from the method.
Consequences if not approved:	⌘ If this particular CR is not agreed, TS 29.198-12 is out of sync. with the other parts of TS 29.198. If the related batch of CRs is not agreed, OSA will have a limited acceptance among the application development community, since it will be more difficult to implement. This presents a risk to the return on investment in development of OSA.

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

How to create CRs using this form:

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

- 1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.
- 2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be

downloaded from the 3GPP server under <ftp://ftp.3gpp.org/specs/> For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

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

7 The Service Interface Specifications

7.1 Interface Specification Format

This section defines the interfaces, methods and parameters that form a part of the API specification. The Unified Modelling Language (UML) is used to specify the interface classes. The general format of an interface specification is described below.

7.1.1 Interface Class

This shows a UML interface class description of the methods supported by that interface, and the relevant parameters and types. The Service and Framework interfaces for enterprise-based client applications are denoted by classes with name Ip<name>. The callback interfaces to the applications are denoted by classes with name IpApp<name>. For the interfaces between a Service and the Framework, the Service interfaces are typically denoted by classes with name IpSvc<name>, while the Framework interfaces are denoted by classes with name IpFw<name>

7.1.2 Method descriptions

Each method (API method “call”) is described. All methods in the API return a value of type TResult, indicating, amongst other things, if the method invocation was successfully executed or not.

Both synchronous and asynchronous methods are used in the API. Asynchronous methods are identified by a 'Req' suffix for a method request, and, if applicable, are served by asynchronous methods identified by either a 'Res' or 'Err' suffix for method results and errors, respectively. To handle responses and reports, the application or service developer must implement the relevant IpApp<name> or IpSvc<name> interfaces to provide the callback mechanism.

7.1.3 Parameter descriptions

Each method parameter and its possible values are described. Parameters described as 'in' represent those that must have a value when the method is called. Those described as 'out' are those that contain the return result of the method when the method returns.

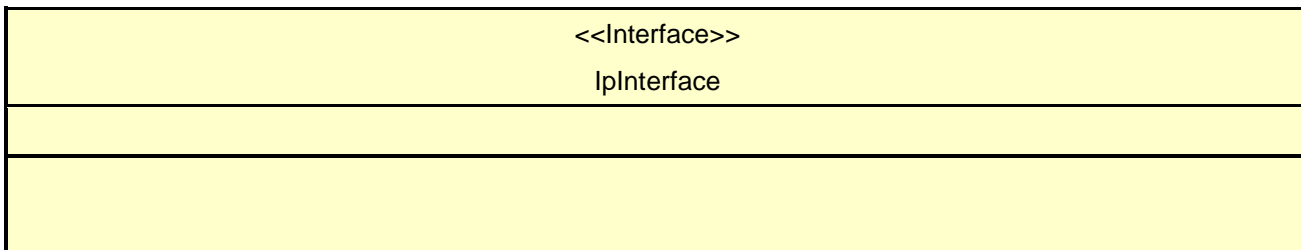
7.1.4 State Model

If relevant, a state model is shown to illustrate the states of the objects that implement the described interface.

7.2 Base Interface

7.2.1 Interface Class IpInterface

All application, framework and service interfaces inherit from the following interface. This API Base Interface does not provide any additional methods.



7.3 Service Interfaces

7.3.1 Overview

The Service Interfaces provide the interfaces into the capabilities of the underlying network - such as call control, user interaction, messaging, mobility and connectivity management.

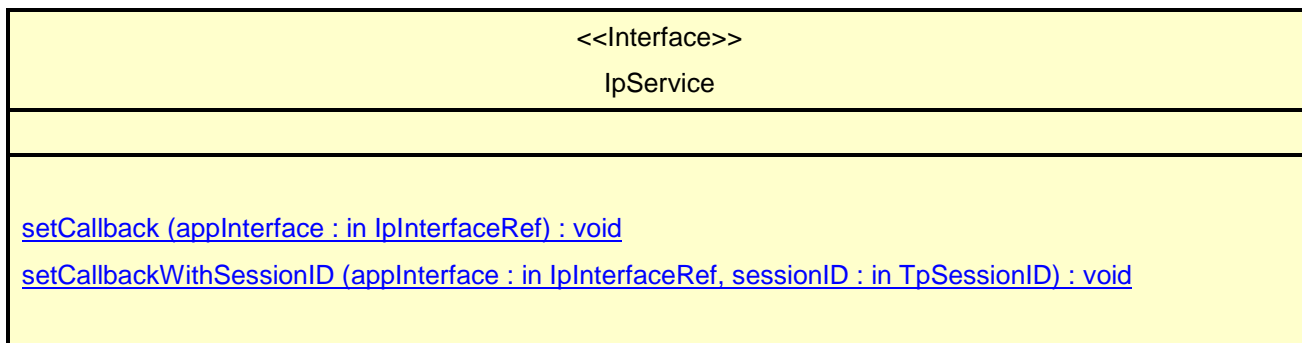
The interfaces that are implemented by the services are denoted as 'Service Interface'. The corresponding interfaces that must be implemented by the application (e.g. for API callbacks) are denoted as 'Application Interface'.

7.4 Generic Service Interface

7.4.1 Interface Class IpService

Inherits from: IpInterface

All service interfaces inherit from the following interface.



Method

setCallback()

This method specifies the reference address of the callback interface that a service uses to invoke methods on the application. It is not allowed to invoke this method on an interface that uses SessionID's.

Parameters

appInterface : in IpInterfaceRef

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

Raises

TpCommonExceptions

Method

setCallbackWithSessionID()

This method specifies the reference address of the application's callback interface that a service uses for interactions associated with a specific session ID: e.g. a specific call, or call leg. It is not allowed to invoke this method on an interface that does not uses SessionID's.

Parameters

appInterface : in IpInterfaceRef

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

sessionId : in TpSessionID

Specifies the session for which the service can invoke the application's callback interface.

Raises

TpCommonExceptions, P_INVALID_SESSION_ID

8 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.

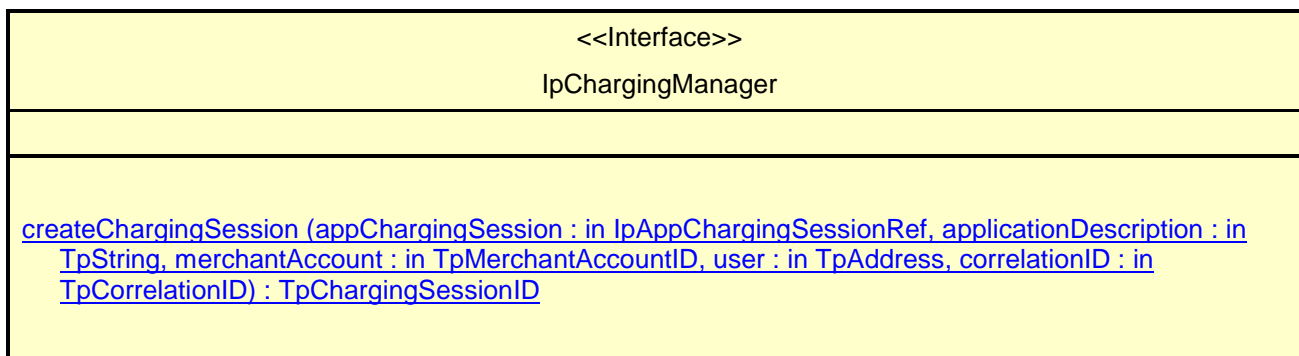
In the interfaces of the Charging SCF a "Request Number" is used when invoking operations that operate on the user's account (directly or indirectly via reservations) in order to make retries possible after application, service, or communication errors. A retry of these operations can be done by invoking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

8.1 Interface Class IpChargingManager

Inherits from: IpService.

This interface is the 'service manager' interface for the Charging Service. The Charging manager interface provides management functions to the charging service. The application programmer can use this interface to start charging sessions.



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.](#)

[Returns chargingSession: Defines the session.](#)

*Parameters***appChargingSession : in IpAppChargingSessionRef**

Callback interface for the session in the application

applicationDescription : 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

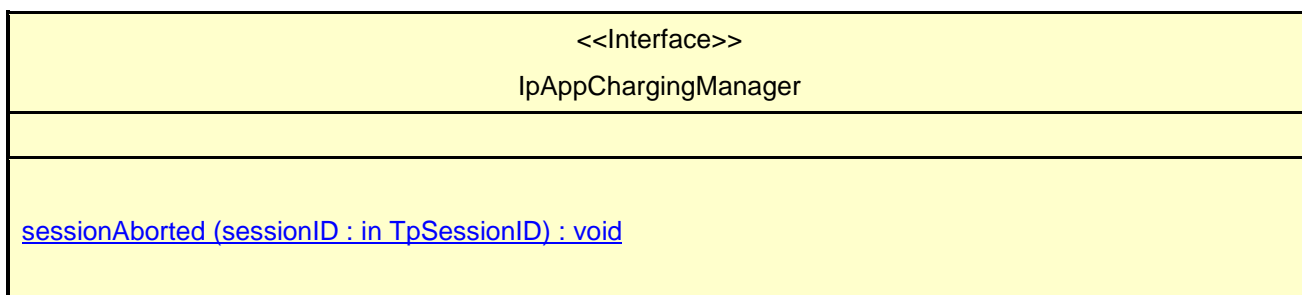
This value can be used to correlate the charging to network activity.

Returns[TpChargingSessionID](#)*Raises***TpCommonExceptions, P_INVALID_USER, P_INVALID_ACCOUNT**

8.2 Interface Class IpAppChargingManager

Inherits from: IpInterface.

This interface is the manager application interface for the Charging Service. The Charging manager interface provides the application Charging Session Management functions to the charging service.

*Method***sessionAborted()**

This method indicates to the application that the charging session object (at the gateway) has aborted or terminated abnormally. No further communication will be possible between the charging session and application.

*Parameters***sessionID : in TpSessionID**

Specifies the sessionID of the charging session that has aborted or terminated abnormally.

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
creditAmountReq (sessionID : in TpSessionID, applicationDescription : in TpApplicationDescription, amount : in TpChargingPrice, closeReservation : in TpBoolean, requestNumber : in TpInt32) : void
creditUnitReq (sessionID : in TpSessionID, applicationDescription : in TpApplicationDescription, volumes : in TpVolumeSet, closeReservation : in TpBoolean, requestNumber : in TpInt32) : void
debitAmountReq (sessionID : in TpSessionID, applicationDescription : in TpApplicationDescription, amount : in TpChargingPrice, closeReservation : in TpBoolean, requestNumber : in TpInt32) : void
debitUnitReq (sessionID : in TpSessionID, applicationDescription : in TpApplicationDescription, volumes : in TpVolumeSet, closeReservation : in TpBoolean, requestNumber : in TpInt32) : void
directCreditAmountReq (sessionID : in TpSessionID, applicationDescription : in TpApplicationDescription, chargingParameters : in TpChargingParameterSet, amount : in TpChargingPrice, requestNumber : in TpInt32) : void
directCreditUnitReq (sessionID : in TpSessionID, applicationDescription : in TpApplicationDescription, chargingParameters : in TpChargingParameterSet, volumes : in TpVolumeSet, requestNumber : in TpInt32) : void
directDebitAmountReq (sessionID : in TpSessionID, applicationDescription : in TpApplicationDescription, chargingParameters : in TpChargingParameterSet, amount : in TpChargingPrice, requestNumber : in TpInt32) : void
directDebitUnitReq (sessionID : in TpSessionID, applicationDescription : in TpApplicationDescription, chargingParameters : in TpChargingParameterSet, volumes : in TpVolumeSet, requestNumber : in TpInt32) : void
extendLifeTimeReq (sessionID : in TpSessionID) : void
getAmountLeft (sessionID : in TpSessionID) : TpChargingPrice
getLifeTimeLeft (sessionID : in TpSessionID) : TpInt32
getUnitLeft (sessionID : in TpSessionID) : TpVolumeSet
rateReq (sessionID : in TpSessionID, chargingParameters : in TpChargingParameterSet) : void
release (sessionID : in TpSessionID, requestNumber : in TpInt32) : void
reserveAmountReq (sessionID : in TpSessionID, preferredAmount : in TpChargingPrice, minimumAmount : in TpChargingPrice, applicationDescription : in TpApplicationDescription, requestNumber : in TpInt32, chargingParameters : in TpChargingParameterSet) : void
reserveUnitReq (sessionID : in TpSessionID, chargingParameters : in TpChargingParameterSet, volumes : in TpVolumeSet, applicationDescription : in TpApplicationDescription, requestNumber : in TpInt32) : void

*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***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***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***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***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

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***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***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

*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***getAmountLeft()**

[With this method an application can request the remaining amount of the reservation.](#)

[Returns amountLeft: Gives the amount left in the reservation.](#)

Parameters

sessionID : in TpSessionID

The ID of the session.

Returns[TpChargingPrice](#)*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 \(P_TASK_REFUSED\).](#)

[Returns reservationTimeLeft: Indicates the number of seconds that the session remains valid.](#)

*Parameters***sessionID : in TpSessionID**

The ID of the session.

Returns[TpInt32](#)*Raises***TpCommonExceptions, P_INVALID_SESSION_ID***Method***getUnitLeft()**

[With this method an application can request the remaining amount of the reservation.](#)

[Returns volumesLeft: Specifies the remaining volumes in different units, more specifically a sequence of data elements each containing the amount and applied unit.](#)

*Parameters***sessionID : in TpSessionID**

The ID of the session.

Returns[TpVolumeSet](#)*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***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***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.

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***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.

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

8.4 Interface Class IpAppChargingSession

Inherits from: IpInterface.

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



[requestNumberNextRequest : in TpInt32\) : void](#)
[extendLifeTimeErr \(sessionID : in TpSessionID, error : in TpChargingError\) : void](#)
[extendLifeTimeRes \(sessionID : in TpSessionID, SessionTimeLeft : in TpInt32\) : void](#)
[rateErr \(sessionID : in TpSessionID, error : in TpChargingError\) : void](#)
[rateRes \(sessionID : in TpSessionID, rates : in TpPriceVolumeSet, validityTimeLeft : in TpDuration\) : void](#)
[reserveAmountErr \(sessionID : in TpSessionID, error : in TpChargingError, requestNumberNextRequest : in TpInt32\) : void](#)
[reserveAmountRes \(sessionID : in TpSessionID, reservedAmount : in TpChargingPrice, sessionTimeLeft : in TpInt32, requestNumberNextRequest : in TpInt32\) : void](#)
[reserveUnitErr \(sessionID : in TpSessionID, error : in TpChargingError, requestNumberNextRequest : in TpInt32\) : void](#)
[reserveUnitRes \(sessionID : in TpSessionID, reservedUnits : in TpVolumeSet, sessionTimeLeft : in TpInt32, requestNumberNextRequest : in TpInt32\) : void](#)
[sessionEnded \(sessionID : in TpSessionID, report : in TpSessionEndedCause\) : void](#)

*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.

requestNumberNextRequest : in TpInt32

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

*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.

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***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

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.

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***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

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.

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***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

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.

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***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.

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***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.

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***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.

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***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.

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***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.

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***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.

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***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.

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.

*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.

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***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***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***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***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***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.

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***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.

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***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.

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***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.

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 deassign 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.