3GPP TSG CN Plenary Meeting #16 **NP-020194**
5th - 7th June 2002. Marco Island, USA.

Source: **CN5 (OSA)**

Title: **Rel-5 CRs 29.198-12 OSA API Part 12: Charging**

Agenda item: **8.2**

Document for: **APPROVAL**

Doc-1st -Level	Spec	CR	Rev	Pha	Subject	Cat	Ver Curr	Ver New	Doc-2nd -Level	Work item
NP-020194	29.198-12	012	-	Rel-5	Clarify the use of setCallback with charging	F	4.3.0	5.0.0	N5-020066	OSA2
NP-020194	29.198-12	013	-	Rel-5	Adding Service Properties for the Content Based Charging API	B	4.3.0	5.0.0	N5-020300	OSA2
NP-020194	29.198-12	014	-	Rel-5	Addition of support for interactive authorization of payments ("User Confirmation")	B	4.3.0	5.0.0	N5-020358	OSA2
NP-020194	29.198-12	015	-	Rel-5	Addition of support for Split Charging feature	B	4.3.0	5.0.0	N5-020361	OSA2

3GPP TSG-CN5 Meeting #16
Hong Kong, CHINA, 4th – 8th February 2002

N5-020066

CR-Form-v6.1

CHANGE REQUEST

⌘ **29.198-12** **CR** **012** ⌘ **rev** **-** ⌘ Current version: **4.3.0** ⌘

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

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

Title:	⌘	Clarify the use of setCallback with charging
Source:	⌘	CN5
Work item code:	⌘	OSA2
Date:	⌘	30/05/2002
Category:	⌘	**F**
Release:	⌘	REL-5

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:	⌘	It has been indicated e.g. in call control that the application should provide the callback reference before invoking e.g. createCall to make it possible to invoke callAborted. Similarly this holds true for the Charging SCF, but it has not been stated.
Summary of change:	⌘	It is indicated that the setCallback method should be called prior to invoking createChargingSession.
Consequences if not approved:	⌘	sessionAborted cannot be called by the Charging SCS which may lead to erroneous operation in different implementations.

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

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.

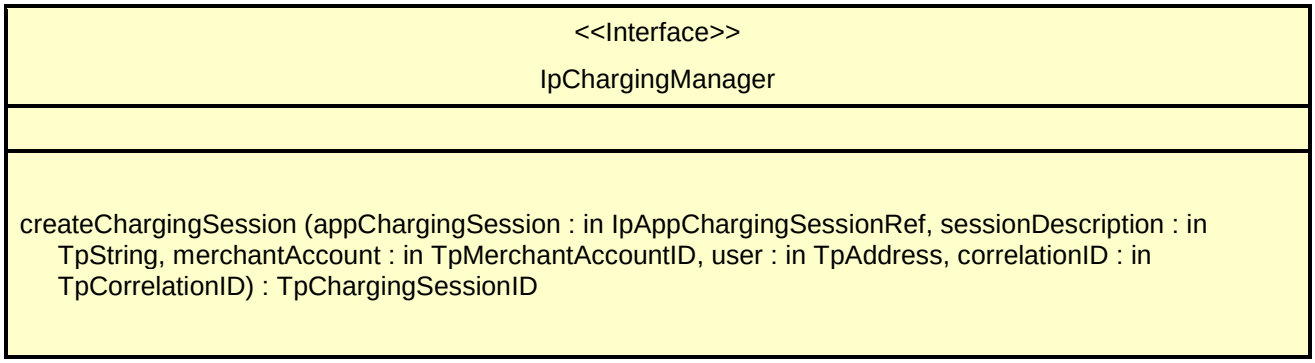
Because responses from the Charging SCF can be delayed in the network the Charging SCF shall guarantee that Request Numbers are unique in a timespan where delayed responsed can arrive. Suppose, for example, that the response from a retried request is received indicating the next request number to use 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.

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.

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. An IpAppChargingManager should already have been passed to the IpChargingManager, otherwise the charging manager will not be able to report a sessionAborted() to the application (the application should invoke setCallback() if it wishes to ensure this).

Returns chargingSession: Defines the session.

Parameters

appChargingSession : in IpAppChargingSessionRef

Callback interface for the session in the application

sessionDescription : in TpString

Descriptive text for informational purposes.

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

CR-Form-v5

CHANGE REQUEST

⌘ **29.198-12** **CR** **013** ⌘ **rev** **-** ⌘ Current version: **4.3.0** ⌘

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

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

Title:	⌘	Adding Service Properties for the Content Based Charging API
Source:	⌘	CN5
Work item code:	⌘	OSA2
Date:	⌘	30/05/2002
Category:	⌘	**B**
Release:	⌘	REL-5

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:	⌘	OSA Release 5 requirement (SA1) for introduction of service properties for the Content Based Charging API.
Summary of change:	⌘	Service Properties are added for the Content Based Charging API.
Consequences if not approved:	⌘	

Clauses affected:	⌘	10
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

For Parlay V4/OSA R5 Service Properties are required for Content Based Charging (see Tdoc N5-011132).

This change request contains a proposal for the final list of service properties applicable to the Content Based Charging API. Note that it also includes a property for split charging that is subject of another CR.

Proposed Change

This section lists the changes to 29.198-12 (with revision marks).

10 Content Based Charging Service Properties

The following table lists properties relevant for CBC SCF.

Property	Type	Description/Interpretation
P_ADDRESSPLAN	INTEGER_SET	Indicates the supported address plan (defined in TpAddressPlan.) E.g. {P_ADDRESS_PLAN_E164, P_ADDRESS_PLAN_IP})
P_SUPPORTED_UNITS	INTEGER_SET	Indicates the unit-types that are supported, e.g. {P_CHS_UNIT_OCTETS, P_CHS_UNIT_SECONDS}
P_SUPPORTED_CURRENCIES	STRING_SET	Indicates the curreny-types that are supported according to ISO-4217:1995, e.g. {“EUR”, “DEM”, “NLG”}
P_UNIT_CHARGING	BOOLEAN_SET	Indicates if charging based on units (rather then amounts) is supported. Value = TRUE: unit based charging is supported Value = FALSE: unit based charging is not supported If unit charging is supported or not or is selected or not does not tell anything about amount charging.
P_AMOUNT_CHARGING	BOOLEAN_SET	Indicates if charging based on amounts (rather then units) is supported. Value = TRUE: amount based charging is supported Value = FALSE: amount based charging is not supported If amount charging is supported or not or is selected or not does not tell anything about unit charging.
P_SPLIT_CHARGING	BOOLEAN_SET	Indicates if split charging feature is available. Value = TRUE: split charging is supported Value = FALSE: split charging is not supported
P_DEBITING	BOOLEAN_SET	Upon service registration, this property describes if the SCS supports debiting at all and if it can be turned off by the application. Upon service instantiation, it describes which mode(s) the client has selected.
P_CREDITING	BOOLEAN_SET	Upon service registration, this property describes if the SCS supports crediting at all and if it can be turned off by the application. Upon service instantiation, it describes which mode(s) the client has selected.

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

Property	Type	Description/Interpretation
P_DEFAULT_LIFETIME	INTEGER_INTERVAL	Defines the default lifetime for a charging reservation in milliseconds.
P_LIFETIME_INCREMENT	INTEGER_INTERVAL	Defines the duration in milliseconds by which the lifetime of a charging reservation can be extended.
P_MAX_LIFETIME	INTEGER_INTERVAL	Defines the maxium lifetime for a charging reservation in milliseconds.
P_MIN_DEBIT_AMOUNT	STRING_SET	Defines the minimum amounts for a debit operation, depending on currency. Each set element is a string that contains the amount, formatted as a string and followed by the currency. Example: {'1.00 EUR', '0.5 GBP'} means that the minimum amount in Euro is 1.00, while in Pound Sterling it is 0.5
P_MAX_DEBIT_AMOUNT	STRING_SET	Defines the maximum amount for a debit operation, similar to P_MIN_DEBIT_AMOUNT
P_CREDIT_AMOUNT	INTEGER_INTERVAL	Defines the range for amounts that can be credited. Valid for any allowed currency.
P_PARALLEL_SESSIONS	INTEGER_INTERVAL	Defines the range for the allowed amount of parallel charging sessions.
P_SESSIONS_HOUR	INTEGER_INTERVAL	Defines the range for the allowed number of charging sessions per hour.

joint API group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) **N5-020358**
Meeting #18, Budapest, HUNGARY, 13 – 17 May 2002

CR-Form-v5

CHANGE REQUEST

⌘ **29.198-12** **CR** **014** ⌘ **rev** **-** ⌘ Current version: **4.3.0** ⌘

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

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

Title:	⌘	Addition of support for interactive authorization of payments ("User Confirmation")
Source:	⌘	CN5
Work item code:	⌘	OSA2 ***Date:*** ⌘ 17/05/2002
Category:	⌘	**B** ***Release:*** ⌘ *REL-5*

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:	⌘	Alignement with 22.127 in order to support interactive authorization of payments in OSA Rel-5. Addition of support for authorization of payments is required by 22.127 v5.3.0.
Summary of change:	⌘	Error codes added, Attribute names and types for ChargingParameter type added.
Consequences if not approved:	⌘	

Clauses affected:	⌘	8.4, 14.1.21 – 14.1.23, 14.1.30		
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.

12.4 Interface Class IpAppChargingSession

Inherits from: IpInterface.

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

<<Interface>> IpAppChargingSession
creditAmountErr (sessionID : in TpSessionID, requestNumber : in TpInt32, error : in TpChargingError, requestNumberNextRequest : in TpInt32) : void creditAmountRes (sessionID : in TpSessionID, requestNumber : in TpInt32, creditedAmount : in TpChargingPrice, reservedAmountLeft : in TpChargingPrice, requestNumberNextRequest : in TpInt32) : void creditUnitErr (sessionID : in TpSessionID, requestNumber : in TpInt32, error : in TpChargingError, requestNumberNextRequest : in TpInt32) : void creditUnitRes (sessionID : in TpSessionID, requestNumber : in TpInt32, creditedVolumes : in TpVolumeSet, reservedUnitsLeft : in TpVolumeSet, requestNumberNextRequest : in TpInt32) : void debitAmountErr (sessionID : in TpSessionID, requestNumber : in TpInt32, error : in TpChargingError, requestNumberNextRequest : in TpInt32) : void debitAmountRes (sessionID : in TpSessionID, requestNumber : in TpInt32, debitedAmount : in TpChargingPrice, reservedAmountLeft : in TpChargingPrice, requestNumberNextRequest : in TpInt32) : void debitUnitErr (sessionID : in TpSessionID, requestNumber : in TpInt32, error : in TpChargingError, requestNumberNextRequest : in TpInt32) : void debitUnitRes (sessionID : in TpSessionID, requestNumber : in TpInt32, debitedVolumes : in TpVolumeSet, reservedUnitsLeft : in TpVolumeSet, requestNumberNextRequest : in TpInt32) : void directCreditAmountErr (sessionID : in TpSessionID, requestNumber : in TpInt32, error : in TpChargingError, requestNumberNextRequest : in TpInt32) : void directCreditAmountRes (sessionID : in TpSessionID, requestNumber : in TpInt32, creditedAmount : in TpChargingPrice, requestNumberNextRequest : in TpInt32) : void directCreditUnitErr (sessionID : in TpSessionID, requestNumber : in TpInt32, error : in TpChargingError, requestNumberNextRequest : in TpInt32) : void directCreditUnitRes (sessionID : in TpSessionID, requestNumber : in TpInt32, creditedVolumes : in TpVolumeSet, requestNumberNextRequest : in TpInt32) : void directDebitAmountErr (sessionID : in TpSessionID, requestNumber : in TpInt32, error : in TpChargingError, requestNumberNextRequest : in TpInt32) : void directDebitAmountRes (sessionID : in TpSessionID, requestNumber : in TpInt32, debitedAmount : in TpChargingPrice, requestNumberNextRequest : in TpInt32) : void directDebitUnitErr (sessionID : in TpSessionID, requestNumber : in TpInt32, error : in TpChargingError, requestNumberNextRequest : in TpInt32) : void directDebitUnitRes (sessionID : in TpSessionID, requestNumber : in TpInt32, debitedVolumes : in TpVolumeSet, 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, requestNumber : in TpInt32, error : in TpChargingError, requestNumberNextRequest : in TpInt32) : void

reserveAmountRes (sessionID : in TpSessionID, requestNumber : in TpInt32, reservedAmount : in TpChargingPrice, sessionTimeLeft : in TpInt32, requestNumberNextRequest : in TpInt32) : void

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

reserveUnitRes (sessionID : in TpSessionID, requestNumber : in TpInt32, 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.

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 and 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

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

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 and 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

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

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 and 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

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

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 and 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

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

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

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

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

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

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, P_CHS_ERR_CONFIRMATION_REQUIRED.

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

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, P_CHS_ERR_CONFIRMATION_REQUIRED.

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

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.

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_CONFIRMATION_REQUIRED, 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.

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

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_CONFIRMATION_REQUIRED, 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.

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 1 000 octets and 10 charging units will result in two TpVolume elements for this parameter: 1 000 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.

14.1.21 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_CONFIRMATION_ID	3	The ID that references a stored confirmation to authorize the required payment
P_CHS_PARAM_CONTRACT	4	Parameter represents a signed confirmation, which shall be of the type P_CHS_PARAMETER_OCTETSET

14.1.22 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
P_CHS_PARAMETER_OCTETSET	TpOctetSet	OctetValue

14.1.23 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
P_CHS_PARAMETER_OCTETSET	4	Parameter represented by a TpOctetSet

14.1.30 TpChargingError

Indicates the error that occurred.

Name	Value	Description
P_CHS_ERR_UNDEFINED	0	Generic error
P_CHS_ERR_ACCOUNT	1	Merchant account unknown
P_CHS_ERR_USER	2	Unknown user
P_CHS_ERR_PARAMETER	3	The set of charging parameters contains an unknown parameter, or a required parameter is missing.
P_CHS_ERR_NO_DEBIT	4	For some reason the application is not allowed to get money from this user.
P_CHS_ERR_NO_CREDIT	5	For some reason the application is not allowed to pay this user.
P_CHS_ERR_VOLUMES	6	Required volumes are missing.
P_CHS_ERR_CURRENCY	7	This currency is not supported for this transaction.
P_CHS_ERR_NO_EXTEND	8	Request to extend the lifetime of a reservation is rejected.
P_CHS_ERR_RESERVATION_LIMIT	9	This amount or volume violates the bounds of the reservation
P_CHS_ERR_CONFIRMATION_REQUIRED	10	A user confirmation is required, but couldn't be obtained by the SCS. The SCS expects that the client initiates a stored confirmation scenario.

joint API group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) **N5-020361**
Meeting #18, Budapest, HUNGARY, 13 – 17 May 2002

CR-Form-v5

CHANGE REQUEST

⌘ **29.198-12** **CR** **015** ⌘ **rev** **-** ⌘ Current version: **4.3.0** ⌘

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

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

Title:	⌘	Addition of support for Split Charging feature
Source:	⌘	CN5
Work item code:	⌘	OSA2
Date:	⌘	17/05/2002
Category:	⌘	**B**
Release:	⌘	*REL-5*

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:	⌘	Alignement with 22.127 in order to support the Split Charging feature in OSA Rel-5. Addition of support for Split Charging feature as required in 22.127 v5.3.0.
Summary of change:	⌘	createSplitChargingSession method added; description of the split charging feature added.
Consequences if not approved:	⌘	

Clauses affected:	⌘	12, 12.1		
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.

12 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 ensures 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 responses can arrive. Suppose, for example, that the response from a retried request is received indicating the next request number to use is 1 000. During the period that the response to the original request (which also carries the next request number to use equal to 1 000) can arrive, this request number may not be used again.

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 debiting the amount. For example, when after a reservation of 10 minutes a debit request for 5 seconds is done, an error will be returned.

Split Charging Functionality

There are cases where a single instance of the merchant application may serve more than a one service user. Examples are multi-user games or conferences. Typically, the costs for the resources consumed by the single service instance will be split amont all service users.

On the other hand, a merchant application may show advertisements within its application, and in turn the company that is advertised may subside a certain percentage of the application cost. A consumer connecting to the merchant application pays only part of the costs, while the remainder is paid by the advertised company.

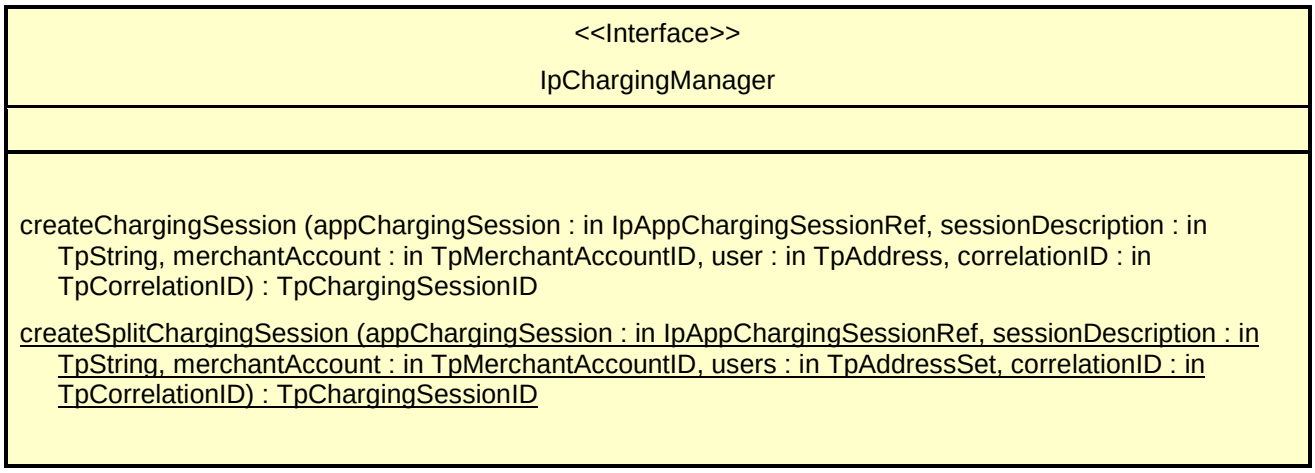
To support this kind of application, multiple users can be specified when a charging session is created. The charging session interface itself is the same no matter if the split charging feature is used or not.

It is subject to service level agreements that are negotiated between the OSA client provider and the network operator how the charge is split between the users.

12.1 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

sessionDescription : in TpString

Descriptive text for informational purposes.

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

Method

createSplitChargingSession()

This method creates an instance of the IpChargingSession interface to handle the charging events related to the specified users and to the application invoking this method. This method differs from createChargingSession() in that it allows to specify multiple users to be charged. The SCS implementation is responsible to figure out how later reserve and charge operations are split among these subscribers. The algorithm may be selected and controlled e.g. through the chargingParameter argument in the respective methods. The algorithms provided and the details how they interpret any parameters are vendor specific.

Returns chargingSession: Defines the session.

Parameters

appChargingSession : in IpAppChargingSessionRef

Callback interface for the session in the application

sessionDescription : in TpString

Descriptive text for informational purposes.

merchantAccount : in TpMerchantAccountID

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

users : in TpAddressSet

Specifies the users that are involved in using the application. This could be all users in a multi-party application (conference call, multi-user-game).

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