3GPP TSG CN Plenary Meeting #14 **NP-010595**
Kyoto, Japan, 12-14 December 2001

Source: **CN5 (OSA)**

Title: **Rel-4 CRs 29.198-02**

Agenda item: **8.5**

Document for: **Decision**

Doc-1st-Level	Spec	CR	R	Pha	Subject	Cat	Ver Cur	Ver -New	Doc-2nd-Level	Workitem
NP-010595	29.198-02	007		Rel-4	**Replace Out Parameters with Return Types**	F	4.2.0	4.3.0	N5-010562	OSA1
NP-010595	29.198-02	008		Rel-4	**Correction to Common Data (CD)**	F	4.2.0	4.3.0	N5-011240	OSA1
NP-010595	29.198-02	009		Rel-4	**Correction to values of TpAddressPlan**	F	4.2.0	4.3.0	N5-011269	OSA1

3GPP TSG_CN5 (Open Service Access – OSA) **N5-010562**
Meeting #12, Sophia Antipolis, FRANCE, 16 – 19 July 2001

CR-Form-v4

CHANGE REQUEST

⌘ **29.198-02 CR 007** ⌘ ev **-** ⌘ Current version: **4.2.0** ⌘

*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		
Source:	⌘	CN5		
Work item code:	⌘	OSA1	***Date:*** ⌘	19/07/2001
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:	⌘	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:	⌘	Removal of TpResult and related types; moving Exceptions specifications from clause 5.8 to 5.4; Correct implementation of N5-010467 (TpAddressPlan changes to support SIP addressing) already agreed at CN5#11
Consequences if not approved:	⌘	If this particular CR is not agreed, TS 29.198-2 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:	⌘	5.4, 5.6, 5.8		
Other specs affected:	⌘	☒ 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.

5.4 Exception Classes

5.4.1 TpCommonExceptions

Defines the structure of the exception class which is applicable to all methods.

Structure Element Name	Structure Element Type	Structure Element Description
ExceptionType	TpInt32	Carries a constant from the list in the table below
ExtraInformation	TpString	Carries extra information to help identify the source of the exception, e.g. a parameter name

5.4.2 Constants associated with TpCommonExceptions

Name	Value	Description
P_RESOURCES_UNAVAILABLE	000Dh	The required resources in the network are not available
P_TASK_REFUSED	000Eh	The requested method has been refused
P_TASK_CANCELLED	000Fh	The requested method has been cancelled
P_NO_CALLBACK_ADDRESS_SET	0011h	The requested method is refused because no callback address has been set (this may be the result of a timing issue between setting the callback address and invoking the method)
P_METHOD_NOT_SUPPORTED	0016h	The method is not allowed or supported within the context of the current service agreement.
P_INVALID_STATE	0306h	Unexpected sequence of methods, i.e., the sequence does not match the specified state diagrams.

5.4.3 Exceptions available to all methods on all interfaces

The following are the list of exception classes which are available to all interfaces of the API.

Name	Description
P_APPLICATION_NOT_ACTIVATED	An application is unauthorised to access information and request services with regards to users that have deactivated that particular application.
P_INFORMATION_NOT_AVAILABLE	An application is unauthorised to access information and request services with regards to users that have set their privacy flag regarding that particular service.
P_INVALID_ADDDRESS	Invalid address specified
P_INVALID_AMOUNT	Invalid amount specified.
P_INVALID_ASSIGNMENT_ID	The assignment ID is invalid
P_INVALID_CRITERIA	Invalid criteria specified
P_INVALID_CURRENCY	Invalid currency specified.
P_INVALID_EVENT_TYPE	Invalid event type
P_INVALID_INTERFACE_NAME	Invalid interface name
P_INVALID_INTERFACE_TYPE	The interface reference supplied by the client is the wrong type.
P_INVALID_NETWORK_STATE	Although the sequence of method calls is allowed by the gateway, the underlying protocol can not support it. E.g., in some protocols some methods are only allowed by the protocol, when the call processing is suspended, e.g., after reporting an event that was monitored in interrupt mode.
P_INVALID_SESSION_ID	Invalid session ID.
P_INVALID_TIME_AND_DATE_FORMAT	Invalid date and time format provided
P_SET_LENGTH_EXCEEDED	The maximum set size is exceeded in a method parameter value.
P_UNKNOWN_SUBSCRIBER	An application is unauthorised to access information and request

Name	Description
	services with regards to users that are not subscribed to the application.
P_UNSUPPORTED_ADDRESS_PLAN	An address contains an address plan which is not supported

~~Method Result Data definitions~~

~~5.4.1 TpResult~~

~~Defines the~~ 5.2.1 Sequence of Data ~~Elements that specify the result of a method call. All methods in the APIs return a result of type TpResult.~~

~~Sequence Element Name~~	~~Sequence Element Type~~
~~ResultType~~	~~TpResultType~~
~~ResultFacility~~	~~TpResultFacility~~
~~ResultInfo~~	~~TpResultInfo~~

~~5.4.2 TpResultType~~

~~Defines whether the method was successful or not.~~

~~Name~~	~~Value~~	~~Description~~
~~P_RESULT_FAILURE~~	~~0~~	~~Method failed~~
~~P_RESULT_SUCCESS~~	~~1~~	~~Method was successful~~

~~5.4.3 TpResultFacility~~

~~Defines the facility code of a result. In phase 2 of the APIs, only P_RESULT_FACILITY_UNDEFINED shall be used.~~

~~Name~~	~~Value~~	~~Description~~
~~P_RESULT_FACILITY_UNDEFINED~~	~~0~~	~~Undefined~~

~~5.4.4 TpResultInfo~~

~~Defines further information relating to the result of the method, such as error codes.~~

~~Name~~	~~Value~~	~~Description~~
~~P_RESULT_INFO_UNDEFINED~~	~~0000h~~	~~No further information present~~
~~P_INVALID_DOMAIN_ID~~	~~0001h~~	~~Invalid client ID~~
~~P_INVALID_AUTH_CAPABILITY~~	~~0002h~~	~~Invalid authentication capability~~
~~P_INVALID_AGREEMENT_TEXT~~	~~0003h~~	~~Invalid agreement text~~
~~P_INVALID_SIGNING_ALGORITHM~~	~~0004h~~	~~Invalid signing algorithm~~
~~P_INVALID_INTERFACE_NAME~~	~~0005h~~	~~Invalid interface name~~
~~P_INVALID_SERVICE_ID~~	~~0006h~~	~~Invalid service ID~~
~~P_INVALID_EVENT_TYPE~~	~~0007h~~	~~Invalid event type~~
~~P_SERVICE_NOT_ENABLED~~	~~0008h~~	~~The service ID does not correspond to a service that has been enabled~~
~~P_INVALID_ASSIGNMENT_ID~~	~~0009h~~	~~The assignment ID is invalid~~
~~P_INVALID_PARAMETER~~	~~000Ah~~	~~The method has been called with an invalid parameter~~
~~P_INVALID_PARAMETER_VALUE~~	~~000Bh~~	~~A method parameter has an invalid value~~
~~P_PARAMETER_MISSING~~	~~000Ch~~	~~A mandatory parameter has not been specified in the method call~~
~~P_RESOURCES_UNAVAILABLE~~	~~000Dh~~	~~The required resources in the network are not available~~
~~P_TASK_REFUSED~~	~~000Eh~~	~~The requested method has been refused~~

~~Name~~	~~Value~~	~~Description~~
~~P_TASK_CANCELLED~~	~~000Fh~~	~~The requested method has been cancelled~~
~~P_INVALID_DATE_TIME_FORMAT~~	~~0010h~~	~~Invalid date and time format provided~~
~~P_NO_CALLBACK_ADDRESS_SET~~	~~0011h~~	~~The requested method is refused because no callback address is set~~
~~P_INVALID_SIGNATURE~~	~~0012h~~	~~Invalid digital signature~~
~~P_INVALID_SERVICE_TOKEN~~	~~0013h~~	~~The service token has not been issued, or it has expired.~~
~~P_ACCESS_DENIED~~	~~0014h~~	~~The client is not currently authenticated with the framework~~
~~P_INVALID_PROPERTY~~	~~0015h~~	~~The framework does not recognise the property supplied by the client~~
~~P_METHOD_NOT_SUPPORTED~~	~~0016h~~	~~The method is not allowed or supported within the context of the current service agreement.~~
~~P_NO_ACCEPTABLE_AUTH_CAPABILITY~~	~~0017h~~	~~An authentication mechanism, which is acceptable to the framework, is not supported by the client~~
~~P_INVALID_INTERFACE_TYPE~~	~~0018h~~	~~The interface reference supplied by the client is the wrong type.~~
~~P_INVALID_ACCESS_TYPE~~	~~0019h~~	~~The framework does not support the type of access interface requested by the client.~~
~~P_SERVICE_ACCESS_DENIED~~	~~001Ah~~	~~The client application is not allowed to access this service.~~
~~P_USER_NOT_SUBSCRIBED~~	~~0030h~~	~~An application is unauthorised to access information and request services with regards to users that are not subscribed to the application.~~
~~P_APPLICATION_NOT_ACTIVATED~~	~~0031h~~	~~An application is unauthorised to access information and request services with regards to users that have deactivated that particular application.~~
~~P_USER_PRIVACY~~	~~0032h~~	~~An application is unauthorised to access information and request services with regards to users that have set their privacy flag regarding that particular service.~~

~~Name~~	~~Value~~	~~Description~~
~~P_GCCS_SERVICE_INFORMATION_MISSING~~	~~0100h~~	~~Information relating to the Call Control service could not be found~~
~~P_GCCS_SERVICE_FAULT_ENCOUNTERED~~	~~0101h~~	~~Fault detected in the Call Control service~~
~~P_GCCS_UNEXPECTED_SEQUENCE~~	~~0102h~~	~~Unexpected sequence of methods, i.e., the sequence does not match the specified state diagrams for the call or the call leg.~~
~~P_GCCS_INVALID_ADDDRESS~~	~~0103h~~	~~Invalid address specified~~
~~P_GCCS_INVALID_CRITERIA~~	~~0104h~~	~~Invalid criteria specified~~
~~P_GCCS_INVALID_NETWORK_STATE~~	~~0105h~~	~~Although the sequence of method calls is allowed by the gateway, the underlying protocol can not support it.~~ ~~E.g., in some protocols some methods are only allowed by the protocol, when the call processing is suspended, e.g., after reporting an event that was monitored in interrupt mode.~~

5.5 Date- and Time-related Data definitions

5.5.1 TpDate

This data type is identical to a TpString. It specifies the data in accordance with International Standard ISO 8601 [4]. This is defined as the string of characters in the following format:

`YYYY-MM-DD`

where the date is specified as:

`YYYY`	four digits year
`MM`	two digits month
`DD`	two digits day

The date elements are separated by a hyphen character (-).

EXAMPLE: The 4 December 1998, is encoded as the string:

`1998-12-04`

5.5.2 TpTime

This data type is identical to a TpString. It specifies the time in accordance with International Standard ISO 8601 [4]. This is defined as the string of characters in the following format:

`HH:MM:SS.mmm`

or

`HH:MM:SS.mmmZ`

where the time is specified as:

`HH`	two digits hours (24h notation)
`MM`	two digits minutes
`SS`	two digits seconds
`mmm`	three digits fractions of a second (i.e. milliseconds)

The time elements are separated by a colon character (`:`). The date and time are separated by a space. Optionally, a capital letter Z may be appended to the time field to indicate Universal Time Co-ordinated (UTC). Otherwise, local time is assumed.

EXAMPLE: 10:30 and 15 seconds is encoded as the string:

`10:30:15.000`

for local time, or in UTC it would be: `10:30:15.000Z`

5.5.3 TpDateAndTime

This data type is identical to a TpString. It specifies the data and time in accordance with International Standard ISO 8601 [4]. This is defined as the string of characters in the following format:

`YYYY-MM-DD HH:MM:SS.mmm`

or

`YYYY-MM-DD HH:MM:SS.mmmZ`

where the date is specified as:

`YYYY`	four digits year
`MM`	two digits month
`DD`	two digits day

The date elements are separated by a hyphen character (-).

The time is specified as:

`HH`	two digits hours (24h notation)
`MM`	two digits minutes
`SS`	two digits seconds
`mmm`	three digits fractions of a second (i.e. milliseconds)

The time elements are separated by a colon character (`:`).The date and time are separated by a space. Optionally, a capital letter `Z` may be appended to the time field to indicate Universal Time Co-ordinated (UTC). Otherwise, local time is assumed.

EXAMPLE: The 4 December 1998, at 10:30 and 15 seconds is encoded as the string:

`1998-12-04 10:30:15.000`

for local time, or in UTC it would be:

`1998-12-04 10:30:15.000Z`

5.5.4 TpDateAndTimeRef

Defines a **Error! Reference source not found.** to type TpDateAndTime.

5.5.5 TpDuration

This data type is a TpInt32 representing a time interval in milliseconds. A value of "-1" defines infinite duration and a value of "-2" represents a default duration.

5.5.6 TpTimeInterval

Defines the Sequence of Data Elements that specify a time interval.

Sequence Element Name	Sequence Element Type
`StartTime`	`TpDateAndTime`
`StopTime`	`TpDateAndTime`

5.6 Address-related Data definitions

5.6.1 TpAddress

Defines the **Error! Reference source not found.** that specify an address.

Sequence Element Name	Sequence Element Type
`Plan`	`TpAddressPlan`
`AddrString`	`TpString`
`Name`	`TpString`
`Presentation`	`TpAddressPresentation`
`Screening`	`TpAddressScreening`
`SubAddressString`	`TpString`

The `AddrString` defines the actual address information and the structure of the string depends on the Plan. The following table gives an overview of the format of the `AddrString` for the different address plans.

Address Plan	AddrString Format Description	Example
`P_ADDRESS_PLAN_NOT_PRESENT`	Not applicable	
`P_ADDRESS_PLAN_UNDEFINED`	Not applicable	
`P_ADDRESS_PLAN_IP`	For Ipv4 the dotted quad notation is used. Also for IPv6 the dotted notation is used. The address can optionally be followed by a port number separated by a colon.	"127.0.0.1:42"
`P_ADDRESS_PLAN_MULTICAST`	An Ipv4 class D address or Ipv6 equivalent in dotted notation.	"224.0.0.0"
`P_ADDRESS_PLAN_UNICAST`	A non multicast or broadcast IP address in dotted notation.	"127.0.0.1"
`P_ADDRESS_PLAN_E164`	An international number without the international access code, including the country code and excluding the leading zero of the area code.	"31161249111"
`P_ADDRESS_PLAN_AESA`	The ATM End System Address in binary format (40 bytes)	01234567890ABCDEF01234567890ABCDEF01234567
`P_ADDRESS_PLAN_URL`	A uniform resource locator as defined in IETF RFC 1738 [6]	"http://www.parlay.org"
`P_ADDRESS_PLAN_NSAP`	The binary representation of the Network Service Access Point	490001AA000400010420
`P_ADDRESS_PLAN_SMTP`	An e-mail address as specified in IETF RFC822 [7]	"webmaster@parlay.org"
`P_ADDRESS_PLAN_MSMAIL`	Identical to P_ADDRESS_PLAN_SMTP	"john.doe@hitech.com"
`P_ADDRESS_PLAN_X400`	The X400 address structured as a set of attribute value pairs separated by semicolons.	"C=nl;ADMD=;PRMD=uninet;O=parlay;S=Doe;I=S;G=John'
`P_ADDRESS_PLAN_SIP (note)`	A valid SIP address string	sip:user@parlay.org <sip:enquiries@1.2.3.4:5060> Enquiries

Note: It should be noted that two SIP addresses will be regarded as equivalent by a gateway if they correspond to the same user at the same network address. The textual form of the two addresses need not be the same. For example, `sip:enquiries@parlay.org` will be deemed to match `<sip:Enquiries@1.2.3.4:5060>Enquiries` (if `parlay.org` resolves to `1.2.3.4`).

5.6.2 TpAddressRef

Defines a **Error! Reference source not found.** to type TpAddress.

5.6.3 TpAddressSet

Defines a Numbered_Set_of_Data_Elements of TpAddress.

5.6.4 TpAddressSetRef

Defines a **Error! Reference source not found.** to type TpAddressSet.

5.6.5 TpAddressPresentation

Defines whether an address can be presented to an end user.

Name	Value	Description
P_ADDRESS_PRESENTATION_UNDEFINED	0	Undefined
P_ADDRESS_PRESENTATION_ALLOWED	1	Presentation Allowed
P_ADDRESS_PRESENTATION_RESTRICTED	2	Presentation Restricted
P_ADDRESS_PRESENTATION_ADDRESS_NOT_AVAILABLE	3	Address not available for presentation

5.6.6 TpAddressScreening

Defines whether an address can be presented to an end user.

Name	Value	Description
P_ADDRESS_SCREENING_UNDEFINED	0	Undefined
P_ADDRESS_SCREENING_USER_VERIFIED_PASSED	1	user provided address verified and passed
P_ADDRESS_SCREENING_USER_NOT_VERIFIED	2	user provided address not verified
P_ADDRESS_SCREENING_USER_VERIFIED_FAILED	3	user provided address verified and failed
P_ADDRESS_SCREENING_NETWORK	4	Network provided address (see Note)
NOTE: Even though the application may provide the address to the gateway, from the end-user point of view it is still regarded as a network provided address.		

5.6.7 TpAddressPlan

Defines the address plan (or numbering plan) used. It is also used to indicate whether an address is actually defined in a TpAddress data element.

Name	Value	Description
P_ADDRESS_PLAN_NOT_PRESENT	-1	No Address Present
P_ADDRESS_PLAN_UNDEFINED	0	Undefined
P_ADDRESS_PLAN_IP	1	IP
P_ADDRESS_PLAN_MULTICAST	2	Multicast
P_ADDRESS_PLAN_UNICAST	3	Unicast
P_ADDRESS_PLAN_E164	4	E.164
P_ADDRESS_PLAN_AESA	5	AESA
P_ADDRESS_PLAN_URL	6	URL
P_ADDRESS_PLAN_NSAP	7	NSAP
P_ADDRESS_PLAN_SMTP	8	SMTP
P_ADDRESS_PLAN_MSMAIL (see Note)	9	Microsoft Mail
P_ADDRESS_PLAN_X400	10	X.400
P_ADDRESS_PLAN_SIP	11	SIP

NOTE: This value is not used in the scope of 3GPP.

For the case where the `P_ADDRESS_PLAN_NOT_PRESENT` is indicated, the rest of the information in the `TpAddress` is not valid.

5.6.8 TpAddressError

Defines the reasons why an address is invalid.

Name	Value	Description
P_ADDRESS_INVALID_UNDEFINED	0	Undefined error
P_ADDRESS_INVALID_MISSING	1	Mandatory address not present
P_ADDRESS_INVALID_MISSING_ELEMENT	2	Mandatory address element not present
P_ADDRESS_INVALID_OUT_OF_RANGE	3	Address is outside of the valid range
P_ADDRESS_INVALID_INCOMPLETE	4	Address is incomplete
P_ADDRESS_INVALID_CANNOT_DECODE	5	Address cannot be decoded

5.6.9 TpAddressRange

This type is identical to TpAddress with the difference that the AddrString can contain wildcards.

Two wildcards are allowed: * which matches zero or more characters and ? which matches exactly one character. For E164 addresses, *wildcards are only allowed at the end of the string whereas ? are allowed at the beginning or end.~~The wildcards are only allowed at the end or at the beginning of the AddrString.~~

Some examples for E164 addresses:

- "123" matches specified number;
- "123*" matches all numbers starting with 123 (including 123 itself);
- "123??*" matches all numbers starting with 123 and at least 5 digits long;
- "123???" matches all numbers starting with 123 and exactly 6 digits long;

~~For e-mail style addresses, the wildcards are allowed at the beginning of the AddrString:~~

~~□"*@parlay.org"matches all email addresses in the parlay.org domain.~~

The following address ranges are illegal:

- 1?3
- 1*3
- ?123*

Legal occurrences of the '*' and '?' characters in AddrString should be escaped by a '\' character. To specify a '\' character '\\' shall be used.

For e-mail style addresses, the wildcards are allowed at the beginning of the AddrString:

- "*@parlay.org" matches all email addresses in the parlay.org domain.

For SIP addresses, wildcards are allowed between the 'sip:' and the '@' in the AddrString, e.g.

- "sip:*@parlay.org" matches all SIP addresses at parlay.org:5060.

5.6.10 TpURL

This data type is identical to a TpString and contains a URL address. The usage of this type is distinct from TpAddress, which can also hold a URL. The latter contains a user address which can be specified in many ways: IP, e-mail, URL etc. On the other hand, the TpURL type does not hold the address of a user and always represents a URL. This type is used in user interaction and defines the URL of the test or stream to be sent to an end-user. It is therefore inappropriate to use a general address here.

5.7 Price-related Data definitions

5.7.1 TpPrice

This data type is identical to a TpString. It specifies price information. This is defined as a string of characters (digits) in the following format:

DDDDDD.DD

5.7.2 TpAoCInfo

Defines the Sequence of Data Elements that specify the Advice Of Charge information to be sent to the terminal.

Sequence Element Name	Sequence Element Type	Description
ChargeOrder	TpAoCOrder	Charge order
Currency	TpString	Currency unit according to ISO-4217:1995 [8]

5.7.3 TpAoCOrder

Defines the Tagged Choice of Data Elements that specify the charge plan for the call.

	Tag Element Type	
	TpAoCOrderCategory	

Tag Element Value	Choice Element Type	Choice Element Name
P_CHARGE_ADVICE_INFO	TpChargeAdviceInfo	ChargeAdviceInfo
P_CHARGE_PER_TIME	TpChargePerTime	ChargePerTime
P_CHARGE_NETWORK	TpString	NetworkCharge

5.7.4 TpCallAoCOrderCategory

Name	Value	Description
P_CHARGE_ADVICE_INFO	0	Set of GSM Charge Advice Information elements according to 3GPP TS 22.024 [5]
P_CHARGE_PER_TIME	1	Charge per time
P_CHARGE_NETWORK	2	Operator specific charge plan specification, e.g. charging table name / charging table entry

5.7.5 TpChargeAdviceInfo

Defines the Sequence of Data Elements that specify the two sets of Advice of Charge parameters. The first set defines the current tariff. The second set may be used in case of a tariff switch in the network.

Sequence Element Name	Sequence Element Type	Description
CurrentCAI	TpCAIElements	Current tariff

NextCAI	TpCAIElements	Next tariff after tariff switch

5.7.6 TpCAIElements

Defines the Sequence of Data Elements that specify the Charging Advice Information elements according to 3GPP TS 22.024 [5].

Sequence Element Name	Sequence Element Type	Description
UnitsPerInterval	TpInt32	Units per interval
SecondsPerTimeInterval	TpInt32	Seconds per time interval
ScalingFactor	TpInt32	Scaling factor
UnitIncrement	TpInt32	Unit increment
UnitsPerDataInterval	TpInt32	Units per data interval
SegmentsPerDataInteral	TpInt32	Segments per data interal
InitialSecsPerTimeInterval	TpInt32	Initial secs per time interval

5.7.7 TpChargePerTime

Defines the Sequence of Data Elements that specify the time based charging information.

Sequence Element Name	Sequence Element Type	Description
InitialCharge	TpInt32	Initial charge amount (in currency units * 0.0001)
CurrentChargePerMinute	TpInt32	Current tariff (in currency units * 0.0001)
NextChargePerMinute	TpInt32	Next tariff (in currency units * 0.0001) after tariff switch Only used in setAdviceOfCharge()

5.7.8 TpLanguage

This data type is identical to a TpString, and defines the language. In case an indication for the language is not needed an empty string shall be used. In other cases valid language strings are defined in ISO 639 [11].

~~5.8 Exception Classes~~

~~5.8.1TpCommonExceptions~~

~~Defines the structure of the exception class which is applicable to all methods.~~

~~Structure Element Name~~	~~Structure Element Type~~	~~Structure Element Description~~
~~exceptionType~~	~~TpInt32~~	~~Carries a constant from the list in the table below~~
~~extraInformation~~	~~TpString~~	~~Carries extra information to help identify the source of the exception, e.g. a parameter name~~

~~5.8.2 Constants associated with TpCommonExceptions~~

~~Name~~	~~Value~~	~~Description~~
~~P_RESOURCES_UNAVAILABLE~~	~~000Dh~~	~~The required resources in the network are not available~~
~~P_TASK_REFUSED~~	~~000Eh~~	~~The requested method has been refused~~
~~P_TASK_CANCELLED~~	~~000Fh~~	~~The requested method has been cancelled~~
~~P_NO_CALLBACK_ADDRESS_SET~~	~~0011h~~	~~The requested method is refused because no callback address is set~~

~~Name~~	~~Value~~	~~Description~~
~~P_METHOD_NOT_SUPPORTED~~	~~0016h~~	~~The method is not allowed or supported within the context of the current service agreement.~~
~~P_INVALID_STATE~~	~~0306h~~	~~Unexpected sequence of methods, i.e., the sequence does not match the specified state diagrams.~~

~~5.8.3 Exceptions available to all methods on all interfaces~~

~~The following are the list of exception classes which are available to all interfaces of the API.~~

~~Name~~	~~Description~~
~~P_APPLICATION_NOT_ACTIVATED~~	~~An application is unauthorised to access information and request services with regards to users that have deactivated that particular application.~~
~~P_INFORMATION_NOT_AVAILABLE~~	~~An application is unauthorised to access information and request services with regards to users that have set their privacy flag regarding that particular service.~~
~~P_INVALID_ADDDRESS~~	~~Invalid address specified~~
~~P_INVALID_AMOUNT~~	~~Invalid amount specified.~~
~~P_INVALID_ASSIGNMENT_ID~~	~~The assignment ID is invalid~~
~~P_INVALID_CRITERIA~~	~~Invalid criteria specified~~
~~P_INVALID_CURRENCY~~	~~Invalid currency specified.~~
~~P_INVALID_EVENT_TYPE~~	~~Invalid event type~~
~~P_INVALID_INTERFACE_NAME~~	~~Invalid interface name~~
~~P_INVALID_INTERFACE_TYPE~~	~~The interface reference supplied by the client is the wrong type.~~
~~P_INVALID_NETWORK_STATE~~	~~Although the sequence of method calls is allowed by the gateway, the underlying protocol can not support it.~~ ~~E.g., in some protocols some methods are only allowed by the protocol, when the call processing is suspended, e.g., after reporting an event that was monitored in interrupt mode.~~
~~P_INVALID_SESSION_ID~~	~~Invalid session ID.~~
~~P_INVALID_TIME_AND_DATE_FORMAT~~	~~Invalid date and time format provided~~
~~P_SET_LENGTH_EXCEEDED~~	~~The maximum set size is exceeded in a method parameter value.~~
~~P_UNKNOWN_SUBSCRIBER~~	~~An application is unauthorised to access information and request services with regards to users that are not subscribed to the application.~~

3GPP TSG_CN5 (Open Service Access – OSA)
Meeting #15, Cancun, MEXICO, 26 – 30 November 2001

N5-011240

CR-Form-v5

CHANGE REQUEST

⌘	**29.198-02**	**CR**	**008**	⌘ **rev**	**-**	⌘	Current version:	**4.2.0**	⌘

*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:	⌘	Corrections to Common Data Types: removal of ...Ref and ...RefRef, update of TpSessionID, description of Underlying Technology Exceptions		
Source:	⌘	CN5		
Work item code:	⌘	OSA1	***Date:*** ⌘	30/11/2001
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 scope of use of TpSessionID is not clear in the description. No information is given related to underlying technology exceptions, on which OSA implementations are dependant. Tp...Ref and Ip...RefRef data types are no longer used and should be removed to simplify the API. A previous CR on TpAddressRange was not correctly implemented.
Summary of change:	⌘	TpSessionID description re-written. New Clause 5.4.1 on underlying technology exceptions added. TpAddressRange description corrected. Removed Tp..Ref and Ip..RefRef. Various editorial spelling errors corrected.
Consequences if not approved:	⌘	Lack of understanding of TpSession ID, or of underlying technology exceptions, will lead to incorrect implementations and therefore to interworking difficulties. Parlay and ETSI have already corrected these errors - they need to be corrected in 29.198 to ensure synchronisation between groups and allow development of common applications across 3 platforms

Clauses affected:	⌘	5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7	
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.

5 Common System Data definitions

These data definitions are assumed to be provided by the client operating system.

5.1 Standard Data types

The APIs assume that the following Data types can be supported.

5.1.1 TpBoolean

Defines a Boolean data type.

5.1.2 TpInt32

Defines a signed 32-bit integer.

~~5.1.3 TpInt32Ref~~

~~Defines a Reference to a TpInt32.~~

5.1.~~4~~3 TpFloat

Defines a single precision real number

~~5.1.5 TpFloatRef~~

~~Defines a Reference to a TpFloat~~

5.1.~~6~~4 TpLong~~s~~String

Defines a Byte string, comprising length and data. The length shall be at least a 32-bit integer.

~~5.1.7 TpLongstringRef~~

~~Defines a Reference to a TpLongstring.~~

5.1.~~8~~5 TpOctet

Defines an 8 bit quantity that is not translated during transmission.

~~5.1.9 TpOctetRef~~

~~Defines a Reference to a TpOctet.~~

5.1.~~10~~6 TpOctetSet

Defines a Numbered Set of Data elements of TpOctet.

5.1.~~11~~7 TpString

Defines a Byte string, comprising length and data. The length shall be at least a 16-bit integer.

~~5.1.12 TpStringRef~~

~~Defines a Reference to a TpString.~~

5.1.~~13~~8 TpAssignmentID

This data type is identical to a TpInt32. It specifies a number which identifies an individual event notification enabled by the application or service.

~~5.1.14 TpAssignmentIDRef~~

~~Defines a Reference to type TpAssignmentID.~~

5.1.~~15~~9 TpSessionID

Defines a session ID with a value that is unique within the context of a specific implementation of an interface. This ID is used to identify different sessions (e.g. different call or call leg sessions) of an interface capable of handling multiple sessions.

Example 1, myCallObject may implement the IpCall interface. If so, myCallObject may handle multiple call sessions, and each call session will be identified by a call session ID value (e.g. 1, 2, 3) that is unique within the context of myCallObject.

Example 2, myCallAndCallLegObject may implement the IpCall and IpCallLeg interfaces. If so, myCallAndCallLegObject may handle multiple call sessions and multiple call leg sessions. Each call session will be identified by a call session ID value (e.g. 1, 2, 3) that is unique within the context of myCallAndCallLegObject. Similarly, each call leg session will be identified by a call leg session ID value (e.g. 1, 2, 3, 4, 5, 6) that is also unique within the context of myCallAndCallLegObject. Because call session IDs and call leg session IDs are different data types, overlapping values are permitted and their uniqueness still remains.~~Defines a network unique session ID. The API uses this ID to identify sessions, e.g. call or call leg sessions, within an object implementing an interface capable of handling multiple sessions. For the different services, the sessionIDs are unique only in the context of a service manager instantiation (e.g., within the context of one Generic Call Control manager). As such if an application creates two instances of the same service manager it shall use different instantiations of the callback objects which implement the callback interfaces.~~

The session ID is identical to a TpInt32 type.

~~5.1.16 TpSessionIDRef~~

~~Defines a Reference to a TpSessionID.~~

5.1.~~17~~10 TpSessionIDSet

Defines a Numbered Set of Data Elements of TpSessionID.

5.2 Other Data sorts

The APIs assumes that the following data syntaxes can be supported

5.2.1 Sequence of Data Elements

This describes a sequence of data types. This may be defined as a structure (for example, in C++) or simply a sequence of data elements within a structure.

EXAMPLE: The TpAddress data type may be defined in C++ as:

```
typedef struct {
  TpAddressPlan       Plan;
  TpString         AddrString;
  TpString         Name;
  TpAddressPresentation.....Presentation;
....TpAddressScreening........Screening;
....TpString..................SubAddressString;
} TpAddress;
```

5.2.2 Tagged Choice of Data Elements

This describes a data type which actually evaluates to one of a choice of a number of data elements. This data element contains two parts: a tag data type (the *tag* part) which is used to identify the chosen data type, and the chosen data type itself (the *union* part). This form of data type is also referred to as a tagged union.

This data type can be implemented (for example, in C++) as a structure containing an integer for the *tag* part, and a union for the *union* part.

This data type is implementation specific. Please refer to the appropriate IDL documents (and the resulting language mappings) to see how this data type is implemented.

EXAMPLE: The `TpCallError` data type may be defined in C++ as:

```
typedef struct {
  TpCallErrorType Tag;
  union {
    TpCallErrorInfoUndefined     Undefined;
    TpCallErrorInfoRoutingAborted  RoutingAborted;
    TpCallErrorInfoCallAbandoned   CallAbandoned;
    TpCallErrorInfoInvalidAddress  InvalidAddress;
    TpCallErrorInfoInvalidState   InvalidState;
    TpCallErrorInfoInvalidCriteria  InvalidCriteria;
  } callErrorInfo;
} TpCallError;
```

5.2.3 Numbered Set of Data Elements

This describes a data type which comprises an integer which indicates the total number of data elements in the set (the *number* part), and an **unordered** set of data elements (the *data* part). *Set* data types do not contain duplicate data elements.

EXAMPLE: The TpAddressSet data type may be defined in MIDL as:

```
typedef struct TpAddressSet
{
TpInt32 Number; [size_is(Number)] TpAddress Set[];
}
TpAddressSet;
```

5.2.4 Reference

This describes a reference (or pointer) to a data type.

~~This is primarily used to describe 'out' method parameters.~~

~~This data type may be implemented (for example, in C++) as a pointer. However, in some languages it may not be necessary for 'out' parameters to be implemented as pointers.~~

~~EXAMPLE: The TpAddressRef data type may be defined in C++ as:~~

~~`typedef TpAddress *    TpAddressRef`~~

5.3 Interface Related Data definitions

5.3.1 IpInterface

Defines the address of a generic interface instance.

5.3.2 IpInterfaceRef

Defines a Reference to type IpInterface.

~~5.3.4 IpInterfaceRefRef~~

~~Defines a Reference to type IpInterfaceRef.~~

5.4 Exception Classes

5.4.1 Underlying Technology Exceptions

All methods contain a signature showing, amonst other things, the explicit exceptions that they may throw. In addition to these exceptions, all methods can throw a number of implicit exceptions. These exceptions do not need to be included within the method signatures and are given below.

These exceptions would be thrown by the underlying technology (e.g. CORBA, Java) as a result of problems encountered, for example, with the way the API method is invoked. They are a minimum set of exceptions that must be throwable by the underlying technology. Depending upon the underlying technology, additional method exceptions may also be thrown.

Description
Invalid Parameter: A method has been passed an invalid parameter argument
Invalid Parameter Value: A method parameter has been passed a value that is out of range
Parameter Missing: A method has not been passed a mandatory parameter argument

5.4.~~1~~2 TpCommonExceptions

Defines the structure of the exception class which is applicable to all methods.

Structure Element Name	Structure Element Type	Structure Element Description
ExceptionType	TpInt32	Carries a constant from the list in the table below
ExtraInformation	TpString	Carries extra information to help identify the source of the exception, e.g. a parameter name

5.4.~~2~~3 Constants associated with TpCommonExceptions

Name	Value	Description
P_RESOURCES_UNAVAILABLE	000Dh	The required resources in the network are not available
P_TASK_REFUSED	000Eh	The requested method has been refused
P_TASK_CANCELLED	000Fh	The requested method has been cancelled
P_NO_CALLBACK_ADDRESS_SET	0011h	The requested method is refused because no callback address has been set (this may be the result of a timing issue between setting the callback address and invoking the method)
P_METHOD_NOT_SUPPORTED	0016h	The method is not allowed or supported within the context of the current service agreement.
P_INVALID_STATE	0306h	Unexpected sequence of methods, i.e., the sequence does not match the specified state diagrams.

5.4.~~3~~4 Exceptions available to all methods on all interfaces

The following are the list of exception classes which are available to all interfaces of the API.

Name	Description
P_APPLICATION_NOT_ACTIVATED	An application is unauthorised to access information and request services with regards to users that have deactivated that particular application. In case the request was for information related to multiple user identities the reference to user identities that are causing this exception will be returned in the extra information of the exception.
P_INFORMATION_NOT_AVAILABLE	The requested information is not available. A reason might be that the information is unavailable in the core network or that the application is unauthorised to access the information.An application is unauthorised to access information and request services with regards to users that have set their privacy flag regarding that particular service. In case the request was for information related to multiple user identities, the reference to user identities that are causing this exception will be returned in the extra information of the exception.
P_INVALID_ADDRESS	Invalid address specified
P_INVALID_AMOUNT	Invalid amount specified.
P_INVALID_ASSIGNMENT_ID	The assignment ID is invalid
P_INVALID_CRITERIA	Invalid criteria specified
P_INVALID_CURRENCY	Invalid currency specified.
P_INVALID_EVENT_TYPE	Invalid event type
P_INVALID_INTERFACE_NAME	Invalid interface name
P_INVALID_INTERFACE_TYPE	The interface reference supplied by the client is the wrong type.
P_INVALID_NETWORK_STATE	Although the sequence of method calls is allowed by the gateway, the underlying protocol can not support it. E.g., in some protocols some methods are only allowed by the protocol, when the call processing is suspended, e.g., after reporting an event that was monitored in interrupt mode.
P_INVALID_SESSION_ID	Invalid session ID.
P_INVALID_TIME_AND_DATE_FORMAT	Invalid date and time format provided
P_SET_LENGTH_EXCEEDED	The maximum set size is exceeded in a method parameter value.
P_UNAUTHORISED_PARAMETER_VALUE	A method parameter value violates the Service Level Agreement
P_UNKNOWN_SUBSCRIBER	The subscriber is not known in the network or the application is An application is unauthorised to access information and request services with regards to users that are not subscribed to the application. In case the request was for information related to multiple user identities, the reference to user identities that are causing this exception will be returned in the extra information of the exception.
P_UNSUPPORTED_ADDRESS_PLAN	An address contains an address plan which is not supported

5.5 Date- and Time-related Data definitions

5.5.1 TpDate

This data type is identical to a TpString. It specifies the data in accordance with International Standard ISO 8601 [4]. This is defined as the string of characters in the following format:

`YYYY-MM-DD`

where the date is specified as:

YYYY	four digits year
MM	two digits month
DD	two digits day

The date elements are separated by a hyphen character (-).

EXAMPLE: The 4 December 1998, is encoded as the string:

`1998-12-04`

5.5.2 TpTime

This data type is identical to a TpString. It specifies the time in accordance with International Standard ISO 8601 [4]. This is defined as the string of characters in the following format:

`HH:MM:SS.mmm`

or

`HH:MM:SS.mmmZ`

where the time is specified as:

HH	two digits hours (24h notation)
MM	two digits minutes
SS	two digits seconds
mmm	three digits fractions of a second (i.e. milliseconds)

The time elements are separated by a colon character (:). The date and time are separated by a space. Optionally, a capital letter Z may be appended to the time field to indicate Universal Time Co-ordinated (UTC). Otherwise, local time is assumed.

EXAMPLE: 10:30 and 15 seconds is encoded as the string:

`10:30:15.000`

for local time, or in UTC it would be: `10:30:15.000Z`

5.5.3 TpDateAndTime

This data type is identical to a TpString. It specifies the data and time in accordance with International Standard ISO 8601 [4]. This is defined as the string of characters in the following format:

`YYYY-MM-DD HH:MM:SS.mmm`

or

`YYYY-MM-DD HH:MM:SS.mmmZ`

where the date is specified as:

YYYY	four digits year
MM	two digits month
DD	two digits day

The date elements are separated by a hyphen character (-).

The time is specified as:

HH	two digits hours (24h notation)
MM	two digits minutes
SS	two digits seconds
mmm	three digits fractions of a second (i.e. milliseconds)

The time elements are separated by a colon character (:).The date and time are separated by a space. Optionally, a capital letter Z may be appended to the time field to indicate Universal Time Co-ordinated (UTC). Otherwise, local time is assumed.

EXAMPLE: The 4 December 1998, at 10:30 and 15 seconds is encoded as the string:

```
1998-12-04 10:30:15.000
```

for local time, or in UTC it would be:

```
1998-12-04 10:30:15.000Z
```

~~5.5.4 TpDateAndTimeRef~~

~~Defines a Referenceto type TpDateAndTime.~~

5.5.~~5~~4 TpDuration

This data type is a TpInt32 representing a time interval in milliseconds. A value of "-1" defines infinite duration and a value of "-2" represents a default duration.

5.5.~~6~~5 TpTimeInterval

Defines the Sequence of Data Elements that specify a time interval.

Sequence Element Name	Sequence Element Type
StartTime	TpDateAndTime
StopTime	TpDateAndTime

5.6 Address-related Data definitions

5.6.1 TpAddress

Defines the Sequence of Data Elements that specify an address.

Sequence Element Name	Sequence Element Type
Plan	TpAddressPlan
AddrString	TpString
Name	TpString
Presentation	TpAddressPresentation
Screening	TpAddressScreening
SubAddressString	TpString

The AddrString defines the actual address information and the structure of the string depends on the Plan.
The following table gives an overview of the format of the AddrString for the different address plans.

Address Plan	AddrString Format Description	Example
P_ADDRESS_PLAN_NOT_PRESENT	Not applicable	
P_ADDRESS_PLAN_UNDEFINED	Not applicable	
P_ADDRESS_PLAN_IP	For Ipv4 the dotted quad notation is used. Also for IPv6 the dotted notation is used. The address can optionally be followed by a port number separated by a colon.	"127.0.0.1:42"
P_ADDRESS_PLAN_MULTICAST	An Ipv4 class D address or Ipv6 equivalent in dotted notation.	"224.0.0.0"
P_ADDRESS_PLAN_UNICAST	A non multicast or broadcast IP address in dotted notation.	"127.0.0.1"
P_ADDRESS_PLAN_E164	An international number without the international access code, including the country code and excluding the leading zero of the area code.	"31161249111"
P_ADDRESS_PLAN_AESA	The ATM End System Address in binary format (40 bytes)	01234567890ABCDEF01234567890ABCDEF01234567
P_ADDRESS_PLAN_URL	A uniform resource locator as defined in IETF RFC 1738 [6]	"http://www.parlay.org"
P_ADDRESS_PLAN_NSAP	The binary representation of the Network Service Access Point	490001AA000400010420
P_ADDRESS_PLAN_SMTP	An e-mail address as specified in IETF RFC822 [7]	"webmaster@parlay.org"
P_ADDRESS_PLAN_MSMAIL	Identical to P_ADDRESS_PLAN_SMTP	"john.doe@hitech.com"
P_ADDRESS_PLAN_X400	The X400 address structured as a set of attribute value pairs separated by semicolons.	"C=nl;ADMD=;PRMD=uninet;O=parlay;S=Doe;I=S;G=John'
P_ADDRESS_PLAN_SIP (Note 1)	A valid SIP address string	sip:user@parlay.org <sip:enquiries@1.2.3.4:5060> Enquiries
P_ADDRESS_PLAN_ANY (Note 2)	Not applicable	

NOTE 1: It should be noted that two SIP addresses will be regarded as equivalent by a gateway if they correspond to the same user at the same network address. The textual form of the two addresses need not be the same. For example, sip:enquiries@parlay.org will be deemed to match <sip:Enquiries@1.2.3.4:5060>Enquiries (if parlay.org resolves to 1.2.3.4).

NOTE 2: This is only to be used with TpAddressRange

~~5.6.2 TpAddressRef~~

~~Defines a Reference to type TpAddress.~~

5.6.~~3~~2 TpAddressSet

Defines a Numbered Set of Data Elements of TpAddress.

~~5.6.4 TpAddressSetRef~~

~~Defines a Reference to type TpAddressSet.~~

5.6.~~5~~3 TpAddressPresentation

Defines whether an address can be presented to an end user.

Name	Value	Description
P_ADDRESS_PRESENTATION_UNDEFINED	0	Undefined
P_ADDRESS_PRESENTATION_ALLOWED	1	Presentation Allowed
P_ADDRESS_PRESENTATION_RESTRICTED	2	Presentation Restricted
P_ADDRESS_PRESENTATION_ADDRESS_NOT_AVAILABLE	3	Address not available for presentation

5.6.~~6~~4 TpAddressScreening

Defines whether an address can be presented to an end user.

Name	Value	Description
P_ADDRESS_SCREENING_UNDEFINED	0	Undefined
P_ADDRESS_SCREENING_USER_VERIFIED_PASSED	1	user provided address verified and passed
P_ADDRESS_SCREENING_USER_NOT_VERIFIED	2	user provided address not verified
P_ADDRESS_SCREENING_USER_VERIFIED_FAILED	3	user provided address verified and failed
P_ADDRESS_SCREENING_NETWORK	4	Network provided address (see Note)
NOTE: Even though the application may provide the address to the gateway, from the end-user point of view it is still regarded as a network provided address.		

5.6.~~7~~5 TpAddressPlan

Defines the address plan (or numbering plan) used. It is also used to indicate whether an address is actually defined in a TpAddress data element.

Name	Value	Description
P_ADDRESS_PLAN_NOT_PRESENT	-1	No Address Present
P_ADDRESS_PLAN_UNDEFINED	0	Undefined
P_ADDRESS_PLAN_IP	1	IP
P_ADDRESS_PLAN_MULTICAST	2	Multicast
P_ADDRESS_PLAN_UNICAST	3	Unicast
P_ADDRESS_PLAN_E164	4	E.164
P_ADDRESS_PLAN_AESA	5	AESA
P_ADDRESS_PLAN_URL	6	URL
P_ADDRESS_PLAN_NSAP	7	NSAP
P_ADDRESS_PLAN_SMTP	8	SMTP
P_ADDRESS_PLAN_MSMAIL (see Note)	9	Microsoft Mail
P_ADDRESS_PLAN_X400	10	X.400
P_ADDRESS_PLAN_SIP	11	SIP
P_ADDRESS_PLAN_ANY	12	Any address plan is deemed to match (This is only used for TpAddressRange)

NOTE: This value is not used in the scope of 3GPP.

For the case where the P_ADDRESS_PLAN_NOT_PRESENT and P_ADDRESS_PLAN_ANY are indicated, the rest of the information in the TpAddress is not valid.

5.6.~~8~~6 TpAddressError

Defines the reasons why an address is invalid.

Name	Value	Description
P_ADDRESS_INVALID_UNDEFINED	0	Undefined error
P_ADDRESS_INVALID_MISSING	1	Mandatory address not present
P_ADDRESS_INVALID_MISSING_ELEMENT	2	Mandatory address element not present
P_ADDRESS_INVALID_OUT_OF_RANGE	3	Address is outside of the valid range
P_ADDRESS_INVALID_INCOMPLETE	4	Address is incomplete
P_ADDRESS_INVALID_CANNOT_DECODE	5	Address cannot be decoded

5.6.~~9~~7 TpAddressRange

Defines the Sequence of Data Elements that specify a range of addresses.

Sequence Element Name	Sequence Element Type
Plan	TpAddressPlan
AddrString	TpString
Name	TpString
SubAddressString	TpString

The AddrString defines the actual address information and the structure of the string depends on the Plan.

An overview of the AddrString formats can be found at the description of the TpAddress data-type.

The difference with TpAddress is that there are no Presentation and Screening elements, the AddrString can contain wildcards and Plan may contain P_ADDRESS_PLAN_ANY.

If `P_ADDRESS_PLAN_ANY` is set then the TpAddressRange will be deemed by the gateway to match any TpAddress. If a specific Plan is set (including `P_ADDRESS_PLAN_NOT_PRESENT`) then the address plan of the range must be identical to the plan contained in an address for the two to match.

Two wildcards are allowed: * which matches zero or more characters and ? which matches exactly one character. For E164 addresses, * which matches zero or more characters and ? are allowed at the beginning or end.
~~For E164 addresses, *wildcards are only allowed at the end of the string whereas ? are allowed at the beginning or end.~~

Some examples for E164 addresses:

- "123" matches specified number;
- "123*" matches all numbers starting with 123 (including 123 itself);
- "123??*" matches all numbers starting with 123 and at least 5 digits long;
- "123???" matches all numbers starting with 123 and exactly 6 digits long;
- "*"matches any address

The following address ranges are illegal:

- "1?3"
- "1*3"
- "?123*"
- ""

~~□1?3~~

~~□1*3~~

~~□?123*~~

Legal occurrences of the '*' and '?' characters in AddrString should be escaped by a '\' character. To specify a '\' character '\\' shall be used.

For e-mail style addresses, the wildcards are allowed at the beginning of the AddrString:

- "*@parlay.org" matches all email addresses in the parlay.org domain.

For SIP addresses, wildcards are allowed between the 'sip:' and the '@' in the AddrString, e.g.

- "sip:*@parlay.org" matches all SIP addresses at parlay.org:5060.

5.6.~~10~~8 TpURL

This data type is identical to a TpString and contains a URL address. The usage of this type is distinct from TpAddress, which can also hold a URL. The latter contains a user address which can be specified in many ways: IP, e-mail, URL etc. On the other hand, the TpURL type does not hold the address of a user and always represents a URL. This type is used in user interaction and defines the URL of the test or stream to be sent to an end-user. It is therefore inappropriate to use a general address here.

5.7 Price-related Data definitions

5.7.1 TpPrice

This data type is identical to a TpString. It specifies price information. This is defined as a string of characters (digits) in the following format:

`DDDDDD.DD`

5.7.2 TpAoCInfo

Defines the Sequence of Data Elements that specify the Advice Of Charge information to be sent to the terminal.

Sequence Element Name	Sequence Element Type	Description
ChargeOrder	TpAoCOrder	Charge order
Currency	TpString	Currency unit according to ISO-4217:1995 [8]

5.7.3 TpAoCOrder

Defines the Tagged Choice of Data Elements that specify the charge plan for the call.

	Tag Element Type	
	TpCallAoCOrderCategory	

Tag Element Value	Choice Element Type	Choice Element Name
P_CHARGE_ADVICE_INFO	TpChargeAdviceInfo	ChargeAdviceInfo
P_CHARGE_PER_TIME	TpChargePerTime	ChargePerTime
P_CHARGE_NETWORK	TpString	NetworkCharge

5.7.4 TpCallAoCOrderCategory

Name	Value	Description
P_CHARGE_ADVICE_INFO	0	Set of GSM Charge Advice Information elements according to 3GPP TS 22.024 [5]
P_CHARGE_PER_TIME	1	Charge per time
P_CHARGE_NETWORK	2	Operator specific charge plan specification, e.g. charging table name / charging table entry

5.7.5 TpChargeAdviceInfo

Defines the Sequence of Data Elements that specify the two sets of Advice of Charge parameters. The first set defines the current tariff. The second set may be used in case of a tariff switch in the network.

Sequence Element Name	Sequence Element Type	Description
CurrentCAI	TpCAIElements	Current tariff
NextCAI	TpCAIElements	Next tariff after tariff switch

5.7.6 TpCAIElements

Defines the Sequence of Data Elements that specify the Charging Advice Information elements according to 3GPP TS 22.024 [5].

Sequence Element Name	Sequence Element Type	Description
UnitsPerInterval	TpInt32	Units per interval
SecondsPerTimeInterval	TpInt32	Seconds per time interval
ScalingFactor	TpInt32	Scaling factor
UnitIncrement	TpInt32	Unit increment
UnitsPerDataInterval	TpInt32	Units per data interval
SegmentsPerDataInterval	TpInt32	Segments per data interval
InitialSecsPerTimeInterval	TpInt32	Initial secs per time interval

5.7.7 TpChargePerTime

Defines the Sequence of Data Elements that specify the time based charging information.

Sequence Element Name	Sequence Element Type	Description
InitialCharge	TpInt32	Initial charge amount (in currency units * 0.0001)
CurrentChargePerMinute	TpInt32	Current tariff (in currency units * 0.0001)
NextChargePerMinute	TpInt32	Next tariff (in currency units * 0.0001) after tariff switch Only used in setAdviceOfCharge()

5.7.8 TpLanguage

This data type is identical to a TpString, and defines the language. In case an indication for the language is not needed an empty string shall be used. In other cases valid language strings are defined in ISO 639 [11].

3GPP TSG_CN5 (Open Service Access – OSA) **N5-011269**
Meeting #15, Cancun, MEXICO, 26 – 30 November 2001

CR-Form-v4

CHANGE REQUEST

⌘	**29.198-02**	**CR**	**009**	⌘	rev	**-**	⌘	Current version:	**4.2.0** ⌘

*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:	⌘	Correction to values of TpAddressPlan			
Source:	⌘	CN5			
Work item code:	⌘	OSA1	***Date:***	⌘	30/11/2001
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:	⌘	TpAddressPlan in the IDL is an enumerated data type. By convention, IDL enumerated types start at value 0. However, TpAddressPlan as documented starts with value -1.
Summary of change:	⌘	Correct TpAddressPlan enumerations to start at 0.
Consequences if not approved:	⌘	Serious discrepancy between implementations based on IDL code, and implementations based on Word document, which will cause interworking problems. This change was already performed in Parlay 2.1 and has not been taken into account in the 3GPP specification set.

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.

5.6.7 TpAddressPlan

Defines the address plan (or numbering plan) used. It is also used to indicate whether an address is actually defined in a TpAddress data element.

Name	Value	Description
`P_ADDRESS_PLAN_NOT_PRESENT`	~~-1~~0	No Address Present
`P_ADDRESS_PLAN_UNDEFINED`	~~0~~1	Undefined
`P_ADDRESS_PLAN_IP`	~~1~~2	IP
`P_ADDRESS_PLAN_MULTICAST`	~~2~~3	Multicast
`P_ADDRESS_PLAN_UNICAST`	~~3~~4	Unicast
`P_ADDRESS_PLAN_E164`	~~4~~5	E.164
`P_ADDRESS_PLAN_AESA`	~~5~~6	AESA
`P_ADDRESS_PLAN_URL`	~~6~~7	URL
`P_ADDRESS_PLAN_NSAP`	~~7~~8	NSAP
`P_ADDRESS_PLAN_SMTP`	~~8~~9	SMTP
`P_ADDRESS_PLAN_MSMAIL (see Note)`	~~9~~10	Microsoft Mail
`P_ADDRESS_PLAN_X400`	~~10~~11	X.400
`P_ADDRESS_PLAN_SIP`	~~11~~12	SIP
`P_ADDRESS_PLAN_ANY`	~~12~~13	Any address plan is deemed to match (This is only used for TpAddressRange)

NOTE: This value is not used in the scope of 3GPP.

For the case where the `P_ADDRESS_PLAN_NOT_PRESENT` and `P_ADDRESS_PLAN_ANY` are indicated, the rest of the information in the `TpAddress` is not valid.