3GPP TSG CN Plenary Meeting #20 NP-030247
04-06 June 2003. Hämeenlinna, FINLAND

Source: CN5 (OSA)

Title: Rel-6 CR 29.198-04-3 OSA API Part 4: Call control; Sub-part 3: Multi-Party Call Control SCF

Agenda item: 9.7

Document for: APPROVAL

Doc-1st-Level	Spec	CR	R	Ph	Subject	Cat	Ver-Curr	Doc-2nd-Level	WI
NP-030247	29.198-04-3	012	-	Rel-6	Unclear overlap criteria for rejection of createNotification	F	5.2.0	N5-021139	OSA3
NP-030247	29.198-04-3	013	-	Rel-6	Add support for advanced subscriber presentation	B	5.2.0	N5-030084	OSA3

joint API group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) **N5-021139**
Meeting #21, Dublin, IRELAND, 28 – 31 October 2002

CR-Form-v5

CHANGE REQUEST

⌘ **29.198-04-3** **CR** **012** ⌘ **rev** **-** ⌘ Current version: **5.2.0** ⌘

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:	⌘	Unclear overlap criteria for rejection of createNotification		
Source:	⌘	Appium		
Work item code:	⌘	OSA3	***Date:*** ⌘	06/11/2002
Category:	⌘	**F**	***Release:*** ⌘	*REL-6*

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)*
Rel-6	*(Release 6)*

Reason for change:	⌘	Unclarity about when a createNotification for MPCC can be refused. The OSA Specification currently does not consider the possibility that a network may allow more applications to be invoked on the same call or session. The current restriction in the OSA specification originate from the Single Point of Control principle as defined in IN CS1 allowing only one service at a time to control a call.
Summary of change:	⌘	Add of clarity to the current notification overlap criteria handling describing when a rejection of createNotification may occur. This also caters for the option that the core network may support multi services, i.e. a Multiple Points of Control concept enabling multiple applications to act on the same call or session. The change proposed is solely in the text part.
Consequences if not approved:	⌘	Not clear when a createNotification for MPCC can be refused with P_INVALID_CRITERIA to the client application. This may cause undesired restrictions in the allowed number of applications that can be invoked on a call or session - and hereby cause unnecesary rejection of applications that could have been allowed. It prevents more opportunities for traffic in the networks. .

Clauses affected:	⌘	6.1		
Other specs affected:	⌘	☒ Other core specifications	⌘	29.198-04-3
		□ Test specifications		
		□ O&M Specifications		
Other comments:	⌘	Correlated with 29.198-04-3 CR in N5-021159		

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

It is not clear when a createNotification for MPCC can be refused with P_INVALID_CRITERIA to the client application . There have in the past been raised some concern about the criteria overlap definition and the associated restriction not to allow more than one application to control a call or session.

Add text to clarify when createNotification will be rejected due to overlapping criteria and the option to overrule this criteria check to enable muti services support.

Part 4-3

Changes to 3GPP TS 29.198-04-3

6 MultiParty Call Control Service Interface Classes

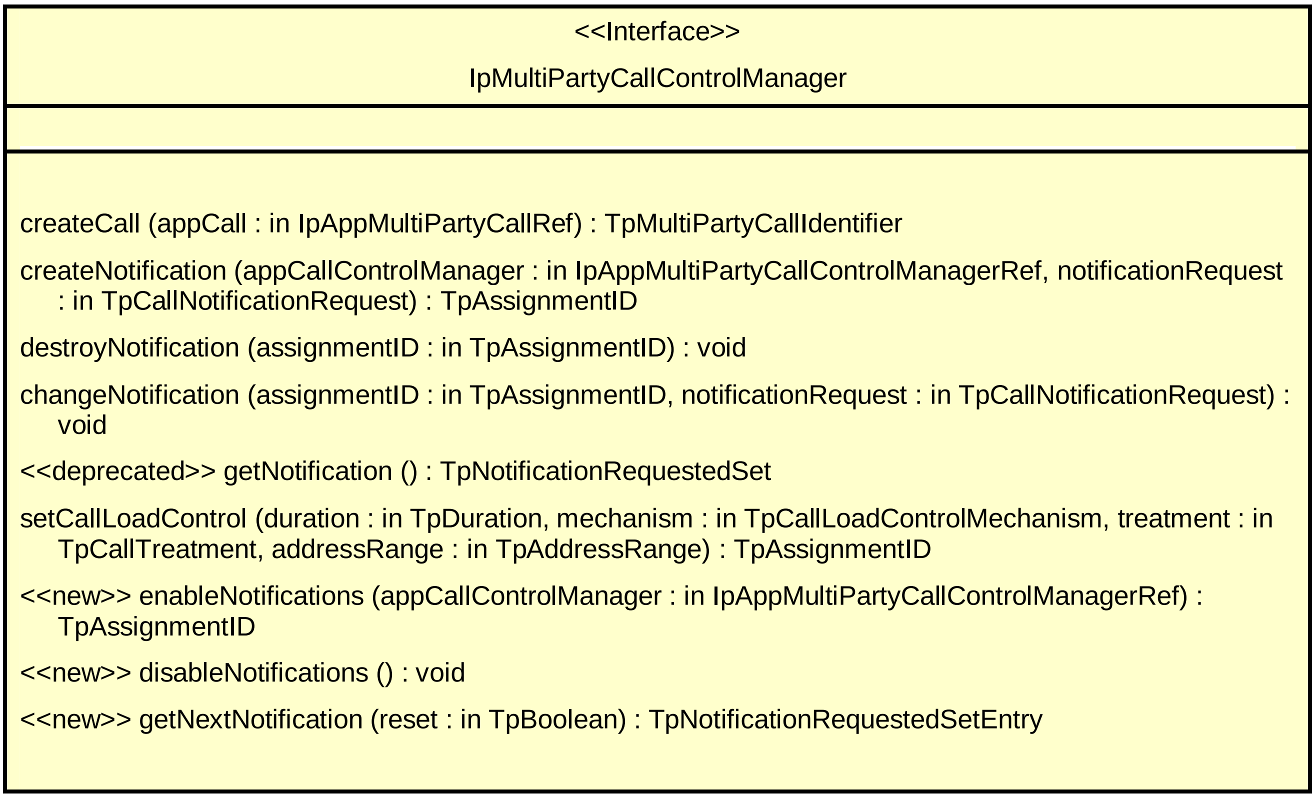
The Multi-party Call Control service enhances the functionality of the Generic Call Control Service with leg management. It also allows for multi-party calls to be established, i.e., up to a service specific number of legs can be connected simultaneously to the same call.

The Multi-party Call Control Service is represented by the IpMultiPartyCallControlManager, IpMultiPartyCall, IpCallLeg interfaces that interface to services provided by the network. Some methods are asynchronous, in that they do not lock a thread into waiting whilst a transaction performs. In this way, the client machine can handle many more calls, than one that uses synchronous message calls. To handle responses and reports, the developer must implement IpAppMultiPartyCallControlManager, IpAppMultiPartyCall and IpAppCallLeg to provide the callback mechanism.

6.1 Interface Class IpMultiPartyCallControlManager

Inherits from: IpService

This interface is the 'service manager' interface for the Multi-party Call Control Service. The multi-party call control manager interface provides the management functions to the multi-party call control service. The application programmer can use this interface to provide overload control functionality, create call objects and to enable or disable call-related event notifications. The action table associated with the STD shows in what state the IpMultiPartyCallControlManager must be if a method can successfully complete. In other words, if the IpMultiPartyCallControlManager is in another state the method will throw an exception immediately.



<<Interface>>

IpMultiPartyCallControlManager

createCall (appCall : in IpAppMultiPartyCallRef) : TpMultiPartyCallIdentifier

createNotification (appCallControlManager : in IpAppMultiPartyCallControlManagerRef, notificationRequest : in TpCallNotificationRequest) : TpAssignmentID

destroyNotification (assignmentID : in TpAssignmentID) : void

changeNotification (assignmentID : in TpAssignmentID, notificationRequest : in TpCallNotificationRequest) : void

<<deprecated>> getNotification () : TpNotificationRequestedSet

setCallLoadControl (duration : in TpDuration, mechanism : in TpCallLoadControlMechanism, treatment : in TpCallTreatment, addressRange : in TpAddressRange) : TpAssignmentID

<<new>> enableNotifications (appCallControlManager : in IpAppMultiPartyCallControlManagerRef) : TpAssignmentID

<<new>> disableNotifications () : void

<<new>> getNextNotification (reset : in TpBoolean) : TpNotificationRequestedSetEntry

6.1.19 Method createNotification()

This method is used to enable call notifications so that events can be sent to the application. This is the first step an application has to do to get initial notifications of calls happening in the network. When such an event happens, the application will be informed by reportNotification(). In case the application is interested in other events during the context of a particular call session it has to use the createAndRouteCallLegReq() method on the call object or the eventReportReq() method on the call leg object. The application will get access to the call object when it receives the reportNotification(). (Note that createNotification() is not applicable if the call is setup by the application).

The createNotification method is purely intended for applications to indicate their interest to be notified when certain call events take place. It is possible to subscribe to a certain event for a whole range of addresses, e.g. the application can indicate it wishes to be informed when a call is made to any number starting with 800.

If some application already requested notifications with criteria that overlap the specified criteria or the specified criteria overlap with criteria already present in the network (when provisioned from within the network), the request is refused with P_INVALID_CRITERIA. The criteria are said to overlap when it leads to more than one application controlling the call or session at the same point in time during call or session processing.

If a notification is requested by an application with monitor mode set to notify, then there is no need to check the rest of the criteria for overlapping with any existing request as the notify mode does not allow control on a call to be passed over. Only one application can place an interrupt request if the criteria overlaps.

If a notification is requested by an application with an event type that is mutually exclusive compared to existing requested event types, then there is no need to check against the rest of the criteria for overlap. An example could be one application that trigger on “user busy” together with another application that trigger on “answer” – both requests should be allowed as only one can occur on the same call or session.

The overlap criteria have been defined to prevent multiple points of control, leading to possible interaction problems in networks that have no multi service support. Notice that dynamic aspects cannot be taken into account in the overlap criteria check. Therefore where dynamic event arming from an application causes a persistent control relationship it can prevent other applications to be invoked in the case single point of application control applies in the network.

However, the criteria check for overlap may as a network option be overruled by Multi Service networks allowing more services or applications to gain control of the same call or session at the same point in time. Refer to Call Control Common Definitions subpart of this specification (TS 29.198-4-1) for further details on application control over a call or session.

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

Returns assignmentID: Specifies the ID assigned by the call control manager interface for this newly-enabled event notification.

joint-API-group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) N5-030084
Meeting #22, Bangkok, THAILAND, 27 – 31 January 2003

CR-Form-v7

CHANGE REQUEST

⌘	**29.198-04-3**	**CR**	**013**	⌘ **rev**	**-**	⌘	Current version:	**5.2.0**	⌘

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

Proposed change affects: UICC apps⌘ □ ME □ Radio Access Network □ Core Network ☒

Title:	⌘	Add support for advanced subscriber presentation
Source:	⌘	Telcordia Technologies – John-Luc Bakker
Work item code:	⌘	OSA3
Date:	⌘	10/02/2003
Category:	⌘	**B**
Release:	⌘	*REL-6*

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)*
Rel-6	*(Release 6)*

Reason for change:	⌘	Multi Party Call Control's support for controlling and inspecting public user identity information is limited. SIP, ISC, and 3GPP TS 23.228 support a more advanced separation of public identity and routing information. The current MPCC service and it derivates do not exploit the full capabilities of a personalization service in 3G networks.
Summary of change:	⌘	Definition of set of standard set of call leg properties, along with methods to get and set the properties. Supported properties to be listed in a service property.
Consequences if not approved:	⌘	No support for controlling and inspecting of advanced public user identity information; not able to exploit 3G personalization capabilities

Clauses affected:	⌘	6.5, 6.5.14-15, 7.3.1.5, 7.3.2.4, 8.1, 9.2.49-53				
		Y	**N**			
Other specs affected:	⌘		**X**	Other core specifications	⌘	
			X	Test specifications		
			X	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/specs/CR.htm. Below is a brief summary:

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

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

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

6.5 Interface Class IpCallLeg

Inherits from: IpService

The call leg interface represents the logical call leg associating a call with an address. The call leg tracks its own states and allows charging summaries to be accessed. The leg represents the signalling relationship between the call and an address. An application that uses the IpCallLeg interface to set up connections has good control, e.g. by defining leg specific event request and can obtain call leg specific report and events.

<<Interface>>

IpCallLeg

routeReq (callLegSessionID : in TpSessionID, targetAddress : in TpAddress, originatingAddress : in TpAddress, appInfo : in TpCallAppInfoSet, connectionProperties : in TpCallLegConnectionProperties) : void

eventReportReq (callLegSessionID : in TpSessionID, eventsRequested : in TpCallEventRequestSet) : void

release (callLegSessionID : in TpSessionID, cause : in TpReleaseCause) : void

getInfoReq (callLegSessionID : in TpSessionID, callLegInfoRequested : in TpCallLegInfoType) : void

getCall (callLegSessionID : in TpSessionID) : TpMultiPartyCallIdentifier

attachMediaReq (callLegSessionID : in TpSessionID) : void

detachMediaReq (callLegSessionID : in TpSessionID) : void

getCurrentDestinationAddress (callLegSessionID : in TpSessionID) : TpAddress

continueProcessing (callLegSessionID : in TpSessionID) : void

setChargePlan (callLegSessionID : in TpSessionID, callChargePlan : in TpCallChargePlan) : void

setAdviceOfCharge (callLegSessionID : in TpSessionID, aOCInfo : in TpAoCInfo, tariffSwitch : in TpDuration) : void

superviseReq (callLegSessionID : in TpSessionID, time : in TpDuration, treatment : in TpCallLegSuperviseTreatment) : void

deassign (callLegSessionID : in TpSessionID) : void

getProperties(callLegSessionID : in TpSessionID, propertyNames : in TpCallLegPropertyNameList) : TpCallLegPropertyList

setProperties(callLegSessionID : in TpSessionID, properties : TpCallLegPropertyList)

6.5.14 Method getProperties()

This synchronous method requests to receive the values of indicated property names if they are available. Examples are a P_CALL_LEG_PROPERTY_ICON (references an image suitable as an iconic representation of the caller or callee), P_CALL_LEG_PROPERTY_INFO (e.g. a web page), or P_CALL_LEG_PROPERTY_CARD (a business card). The caller's properties are available on the call leg object representing the originating address and the callee's properties are available on the call leg object representing callee. If some property value is not available, the property name and value will not be part of the returned list with properties. Note that parts of the caller and callee's public identity are also made available through TpAddress.

The Service Property P_CALL_LEG_PROPERTIES (see section 8.1) indicates the properties that are supported.

Parameters

`callLegSessionID : in TpSessionID`

Specifies the call leg session ID of the call leg.

`propertyNames : in TpCallLegPropertyNameList`

Specifies the property names of the call leg to be made available.

Returns

`TpCallLegPropertyList`

Raises

`TpCommonExceptions, P_INVALID_SESSION_ID, P_INVALID_NETWORK_STATE, P_INFORMATION_NOT_AVAILABLE, P_UNAUTHORISED_PARAMETER_VALUE`

6.5.15 Method setProperties()

This synchronous method requests to set the values of indicated property names and their values if they are supported. Examples are a P_CALL_LEG_PROPERTY_ICON (references an image suitable as an iconic representation of the caller or callee), P_CALL_LEG_PROPERTY_INFO (e.g. a web page), or P_ CALL_LEG_PROPERTY_CARD (a business card). The caller's properties are available on the call leg object representing the originating address and the callee's properties are available on the call leg object representing callee. If some property name is not applicable, it and its value will be ignored. Note that parts of the caller and callee's public identity are also made available through TpAddress.

The Service Property P_CALL_LEG_PROPERTIES (see section 8.1) indicates the properties that are supported.

Parameters

`callLegSessionID : in TpSessionID`

Specifies the call leg session ID of the call leg.

`properties : TpCallLegPropertyList`

Specifies the properties of the call leg to be set.

Raises

`TpCommonExceptions, P_INVALID_SESSION_ID, P_INVALID_NETWORK_STATE, P_INFORMATION_NOT_AVAILABLE, P_UNAUTHORISED_PARAMETER_VALUE`

7.3.1.5 Overview of allowed methods, Originating Call Leg STD

State	Methods allowed
Initiating	getProperties setProperties attachMediaReq (as a request), detachMediaReq, (as a request) getCall , continueProcessing, release (call leg), deassign eventReportReq, getInfoReq, setChargePlan, setAdviceOfCharge, superviseReq
Analysing	getProperties setProperties attachMediaReq (as a request), detachMediaReq, (as a request) getCall , continueProcessing, release (call leg), deassign eventReportReq, getInfoReq, setChargePlan, setAdviceOfCharge, superviseReq
Active	getProperties setProperties attachMediaReq, detachMediaReq, getCall, continueProcessing, release deassign eventReportReq, getInfoReq, setChargePlan, setAdviceOfCharge, superviseReq
Releasing	getCall , continueProcessing, release deassign

7.3.2.4 Overview of allowed methods and trigger events, Terminating Call Leg STD

State	Methods allowed
Idle	routeReq, getCall , getCurrentDestinationAddress, release, deassign eventReportReq, getInfoReq, setChargePlan, setAdviceOfCharge, superviseReq
Active	getProperties setProperties attachMediaReq detachMediaReq getCall , getCurrentDestinationAddress, continueProcessing, release, deassign eventReportReq, getInfoReq, setChargePlan, setAdviceOfCharge, superviseReq
Releasing	getCall , getCurrentDestinationAddress, continueProcessing, release, deassign

8.1 List of Service Properties

The following table lists properties relevant for the MPCC API.

Property	Type	Description / Interpretation
P_TRIGGERING_EVENT_TYPES	INTEGER_SET	Indicates the static event types supported by the SCS. Static events are the events by which applications are initiated.
P_DYNAMIC_EVENT_TYPES	INTEGER_SET	Indicates the dynamic event types supported by the SCS. Dynamic events are the events the application can request for during the context of a call.
P_ADDRESSPLAN	INTEGER_SET	Indicates the supported address plan (defined in TpAddressPlan.) e.g. {P_ADDRESS_PLAN_E164, P_ADDRESS_PLAN_IP})
P_UI_CALL_BASED	BOOLEAN_SET	Value = TRUE : User interaction can be performed on call level and a reference to a Call object can be used in the IpUIManager.createUICall() operation. Value = FALSE: No User interaction on call level is supported.
P_UI_AT_ALL_STAGES	BOOLEAN_SET	Value = TRUE: User Interaction can be performed at any stage during a call . Value = FALSE: User Interaction can be performed in case there is only one party in the call.
P_MEDIA_TYPE	INTEGER_SET	Specifies the media type used by the Service. Values are defined by data-type TpMediaType : P_AUDIO, P_VIDEO, P_DATA
P_MAX_CALLLEGS_PER_CALL	INTEGER_SET	Indicates how many parties can be in one call.
P_UI_CALLLEG_BASED	BOOLEAN_SET	Value = TRUE : User interaction can be performed on leg level and a reference to a CallLeg object can be used in the IpUIManager.createUICall() operation. Value = FALSE : No user interaction on leg level is supported.
P_CALLLEG_PROPERTIES	STRING_SET	Indicates which of the user identity fields are available, valid values are given by TpCallLegPropert~~ies~~yName.
P_PARALLEL_INITIAL_ROUTING_REQUESTS	BOOLEAN_SET	Indicates whether for application initiated calls it is possible to issue multiple routing request methods in parallel or that the application has to wait for the result of the first request before another one can be invoked. Value = TRUE: Multiple routing requests can be invoked in parallel. Value = FALSE: Result of first request has to be received before another request can be issued.

The previous table lists properties related to 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
P_TRIGGERING_ADDRESSES	ADDRESS_RANGE_SET	Indicates for which numbers the notification may be set. For terminating notifications it applies to the terminating number, for originating notifications it applies only to the originating number. See further explanation on which events are originating and which are terminating, below.
P_MONITOR_MODE	INTEGER_SET	Indicates whether the application is allowed to monitor in interrupt and/or notify mode. Set is: P_INTERRUPT P_NOTIFY
P_NUMBERS_TO_BE_CHANGED	INTEGER_SET	Indicates which numbers the application is allowed to change or fill for legs in an incoming call. Allowed value set: {P_ORIGINAL_CALLED_PARTY_NUMBER, P_REDIRECTING_NUMBER, P_TARGET_NUMBER, P_CALLING_PARTY_NUMBER}.
P_CHARGEPLAN_ALLOWED	INTEGER_SET	Indicates which charging is allowed in the setCallChargePlan indicator. Allowed values: {P_TRANSPARANT_CHARGING, P_CHARGE_PLAN}
P_CHARGEPLAN_MAPPING	INTEGER_INTEGER_MAP	Indicates the mapping of chargeplans (we assume they can be indicated with integers) to a logical network chargeplan indicator. When the chargeplan supports indicates P_CHARGE_PLAN then only chargeplans in this mapping are allowed.
P_HIGH_PROBABILITY_OF_COMPLETION	BOOLEAN_SET	Value = TRUE : high probability of call completion field can be set. Value = FALSE : high probability of call completion field can not be set. FALSE is the default value.

The following table explains how the P_TRIGGERING_ADDRESSES property that is inherited via the Generic Call Control properties should be interpreted with respect to which of the notifications apply to originating numbers and which of the notifications apply to terminating numbers.

P_CALL_EVENT_ORIGINATING_CALL_ATTEMPT	Originating
P_CALL_EVENT_ORIGINATING_CALL_ATTEMPT_AUTHORISED	Originating
P_CALL_EVENT_ADDRESS_COLLECTED	Originating
P_CALL_EVENT_ADDRESS_ANALYSED	Originating
P_CALL_EVENT_ORIGINATING_SERVICE_CODE	Originating
P_CALL_EVENT_ORIGINATING_RELEASE	Originating
P_CALL_EVENT_TERMINATING_CALL_ATTEMPT	Terminating
P_CALL_EVENT_TERMINATING_CALL_ATTEMPT_AUTHORISED	Terminating
P_CALL_EVENT_ALERTING	Terminating
P_CALL_EVENT_ANSWER	Terminating
P_CALL_EVENT_TERMINATING_RELEASE	Terminating
P_CALL_EVENT_REDIRECTED	Terminating
P_CALL_EVENT_TERMINATING_SERVICE_CODE	Terminating
P_CALL_EVENT_QUEUED	N/A

9.2.49 TpCallLegPropertyName

This data type is identical to a TpString, and is defined as a string of characters that identify the names of the call leg properties that are to be supported by the Multi Party Call Control API. Other Network operator specific properties may also be used, but should be preceded by the string "SP_". The following values are defined.

Character String Value	Description
P_CALL_LEG_PROPERTY_INFO	The info property name is associated with a URL value that describes the caller or callee in general, for example, through a web page.
P_CALL_LEG_PROPERTY_ICON	The icon parameter property name is associated with a URL value that points to data suitable as an iconic representation of the caller or callee.
P_CALL_LEG_PROPERTY_CARD	The card property name is associated with a business card, for example, in vCard or LDIF formats.

9.2.50 TpCallLegPropertyNameList

This data type defines a `Numbered List of Data Elements` of type TpCallLegPropertyName.

9.2.51 TpCallLegPropertyValue

This data type is identical to `TpString`. It is the value associated with a property.

9.2.52 TpCallLegProperty

This data type is a `Sequence of Data Elements` which describes a property. It is a structured data type consisting of the following {name,value} pair:

Sequence Element Name	Sequence Element Type
CallLegPropertyName	TpCallLegPropertyName
CallLegPropertyValue	TpCallLegPropertyValue

9.2.53 TpCallLegPropertyList

This data type defines a `Numbered List of Data Elements` of type TpCallLegProperty.