

**3GPP TSG CN Plenary Meeting #19
12- 14 March 2003, Birmingham, UK**

NP-030033

Source: CN5 (OSA)
Title: Rel-5 CRs 29.198-05 OSA API Part 5: Generic user interaction
Agenda item: 8.2
Document for: APPROVAL

Doc-1st-Level	Spec	CR	Rev	Phase	Subject	Cat	Version-Current	Doc-2nd-Level	Workitem
NP-030033	29.198-05	029	-	Rel-5	Inconsistent description of use of secondary callback	F	5.1.0	N5-021037	OSA2

CHANGE REQUEST

⌘ **29.198-05 CR 029** ⌘ rev **-** ⌘ Current version: **5.1.0** ⌘

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

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

Title:	⌘ Inconsistent description of use of secondary callback		
Source:	⌘ N5		
Work item code:	⌘ OSA2	Date:	⌘ 10/10/2002
Category:	⌘ F		Release: ⌘ REL-5
	<i>Use <u>one</u> of the following categories:</i> F (correction) A (corresponds to a correction in an earlier release) B (addition of feature), C (functional modification of feature) D (editorial modification) Detailed explanations of the above categories can be found in 3GPP TR 21.900 .		<i>Use <u>one</u> of the following releases:</i> 2 (GSM Phase 2) R96 (Release 1996) R97 (Release 1997) R98 (Release 1998) R99 (Release 1999) REL-4 (Release 4) REL-5 (Release 5)

Reason for change:	⌘ OSA Specification describes use of secondary callback interface inconsistently between the different parts which confuses application developers.		
Summary of change:	⌘ Describe that most recent call back will be used as the callback interface. Only if this one does not work, the initially provided callback interface is used.		
Consequences if not approved:	⌘ Interoperability problems.		

Clauses affected:	⌘		
Other specs affected:	⌘ <input type="checkbox"/> Other core specifications <input type="checkbox"/> Test specifications <input type="checkbox"/> O&M Specifications	⌘	
Other comments:	⌘		

How to create CRs using this form:

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

- 1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.
- 2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under <ftp://ftp.3gpp.org/specs/>. For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.
- 3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

Introduction

The OSA Specifications contains the following descriptions about the use of a secondary callback interface:

Part 1:

7.12 Notification Handling

It is possible to recreate a (set of) notification(s) or re-register for notifications. This is only useful when providing a different callback interface reference. In this case, the last provided interface is used for reporting notifications. ***The earlier provided callback interface is used as “backup” interface (this can be the one provided with setCallback() or setCallbackWithSessionID() if NULL was provided initially). Notifications are reported on this interface when calls to the most recent provided callback interface fail (object providing the interface is crashed or overloaded).*** When re-creating or re-registering, the same assignment ID is returned.

Part 4-2:

6.1.2 Method enableCallNotification()

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 enableCallNotification 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().

Part 4-3:

6.1.2 Method createNotification()

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().

6.1.7 Method <<new>> enableNotifications()

If the same application requests to enable notifications for a second time with a different IpAppMultiPartyCallControlManager reference (i.e. without first disabling them), ***the second callback will be treated as an additional callback. This means that the callback will only be used in cases when the first callback specified by the application is unable to handle the callEventNotify (e.g. due to overload or failure).***

Part 4-4:

6.1.1 Method createMediaNotification()

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 createMediaNotification contains no callback, at the moment the application needs to be informed the gateway will use as callback the one that has been registered by setCallback().

Part 5:

8.1.3 Method createNotification()

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. This means that the callback will only be used in case when the first callback specified by the application is unable to handle the reportNotification (e.g., due to overload or failure).*

8.1.7 Method <<new>> enableNotifications()

If the same application requests to enable notifications for a second time with a different IpAppUIManager reference (i.e. without first disabling them), *the second callback will be treated as an additional callback. This means that the callback will only be used in cases when the first callback specified by the application is unable to handle the callEventNotify (e.g. due to overload or failure).*

Part 8:

8.4.1 Method <<deprecated>> createNotification()

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().

8.4.5 Method <<new>> enableNotifications()

If the same application requests to enable notifications for a second time with a different IpAppDataSessionControlManager reference (i.e. without first disabling them), *the second callback will be treated as an additional callback. This means that the callback will only be used in cases when the first callback specified by the application is unable to handle the callEventNotify (e.g. due to overload or failure).*

8.4.8 Method <<new>> createNotifications()

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().

Part 11:

8.1.1 Method createNotification()

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 enableCallNotification 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().

8.1.7 Method <<new>> enableNotifications()

If the same application requests to enable notifications for a second time with a different IpAppAccountManager reference (i.e. without first disabling them), *the second callback will be treated as an additional callback. This means that the callback will only be used in cases when the first callback specified by the application is unable to handle the reportNotification (e.g. due to overload or failure).*

Solution

The intended use of the 2nd callback interface is as described in part 1, therefore the changes to the following method descriptions are proposed:

- Part 4-3, method enableNotifications()
- Part 5, method createNotification()
- Part 5, method enableNotifications()
- Part 8, method enableNotifications()
- Part 11, method enableNotifications()

This contribution proposes the changes for Part 5.

Proposed Changes

8.1.3 Method createNotification()

This method is used by the application to install specified notification criteria, for which the reporting is implicitly activated. 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 if both originating and terminating ranges overlap and the same number plan is used and the same servicecode is used.

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. The gateway will always use the most recent callback. In case this most recent callback fails the second most recent is used. ~~This means that the callback will only be used in case when the first callback specified by the application is unable to handle the reportNotification (e.g., due to overload or failure).~~

Returns: assignmentID

Specifies the ID assigned by the generic user interaction manager interface for this newly installed notification criteria.

Parameters

appUIManager : in IpAppUIManagerRef

If this parameter is set (i.e. not NULL) it specifies a reference to the application interface, which is used for callbacks. If set to NULL, the application interface defaults to the interface specified via the setCallback() method.

eventCriteria : in TpUIEventCriteria

Specifies the event specific criteria used by the application to define the event required, like user address and service code.

*Returns***TpAssignmentID***Raises***TpCommonExceptions, P_INVALID_CRITERIA, P_INVALID_INTERFACE_TYPE**

8.1.7 Method <<new>> enableNotifications()

This method is used to indicate that the application is able to receive notifications which are provisioned from within the network (i.e. these notifications are NOT set using createNotification() but via, for instance, a network management system). If notifications provisioned for this application are created or changed, the application is unaware of this until the notification is reported.

If the same application requests to enable notifications for a second time with a different IpAppUIManager reference (i.e. without first disabling them), the second callback will be treated as an additional callback. The gateway will always use the most recent callback. In case this most recent callback fails the second most recent is used. ~~This means that the callback will only be used in cases when the first callback specified by the application is unable to handle the callEventNotify (e.g. due to overload or failure).~~

When this method is used, it is still possible to use createNotification() for service provider provisioned notifications on the same interface as long as the criteria in the network and provided by createNotification() do not overlap. However, it is NOT recommended to use both mechanisms on the same service manager.

The methods changeNotification(), getNotification(), and destroyNotification() do not apply to notifications provisioned in the network and enabled using enableNotifications(). These only apply to notifications created using createNotification().

Returns assignmentID: Specifies the ID assigned by the manager interface for this operation. This ID is contained in any reportNotification() that relates to notifications provisioned from within the network.

*Parameters***appUIManager : in IpAppUIManagerRef**

If this parameter is set (i.e. not NULL) it specifies a reference to the application interface, which is used for callbacks. If set to NULL, the application interface defaults to the interface specified via the setCallback() method.

*Returns***TpAssignmentID***Raises***TpCommonExceptions**