

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

NP-030029

Source: CN5 (OSA)
Title: Rel-5 CRs 29.198-04-1 OSA API Part 4: Call control;
Sub-part 1: Call Control Common Definitions
Agenda item: 8.2
Document for: APPROVAL

Doc-1st-Level	Spec	CR	Rev	Phase	Subject	Cat	Version-Current	Doc-2nd-Level	Workitem
NP-030029	29.198-04-1	003	-	Rel-5	Correction to Application's requirements for supporting methods	F	5.1.0	N5-020891	OSA2

CHANGE REQUEST

⌘ **29.198-04-1 CR 003** ⌘ 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: UICC apps ME Radio Access Network Core Network

Title:	⌘ Correction to Application's requirements for supporting methods		
Source:	⌘ N5		
Work item code:	⌘ OSA2	Date:	⌘ 27/09/2002
Category:	⌘ F	Release:	⌘ REL-5
	Use <u>one</u> of the following categories:		Use <u>one</u> of the following releases:
	F (correction)	2	(GSM Phase 2)
	A (corresponds to a correction in an earlier release)	R96	(Release 1996)
	B (addition of feature),	R97	(Release 1997)
	C (functional modification of feature)	R98	(Release 1998)
	D (editorial modification)	R99	(Release 1999)
	Detailed explanations of the above categories can be found in 3GPP TR 21.900 .	Rel-4	(Release 4)
		Rel-5	(Release 5)
		Rel-6	(Release 6)

Reason for change:	⌘ It is not clear in the specification what methods an application should or should not support. This may lead to interworking problems where different application developers choose options which will not interwork.
Summary of change:	⌘ Clarify what an application has to support.
Consequences if not approved:	⌘ Interworking problems may arise.

Clauses affected:	⌘ 4 Call Control SCF										
Other specs affected:	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">Y</td> <td style="width: 20px; text-align: center;">N</td> </tr> <tr> <td style="text-align: center;"> </td> <td style="text-align: center;">X</td> </tr> <tr> <td style="text-align: center;"> </td> <td style="text-align: center;">X</td> </tr> <tr> <td style="text-align: center;"> </td> <td style="text-align: center;">X</td> </tr> </table>	Y	N		X		X		X	Other core specifications	⌘
Y	N										
	X										
	X										
	X										
		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/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.

4 Call Control SCF

Three flavours of Call Control (CC) APIs have been included in 3GPP Release 5. These are the Generic Call Control (GCC), the Multi-Party Call Control (MPCC) and the Multi-Media Call Control (MMCC). The GCC is the same API as was already present in the Release 99 specification (TS 29.198 v3.3.0). Multi-Party Call Control was introduced in the Release 4 specifications, and Multi-Media Call Control is introduced in Release 5.

The joint work between 3GPP CN5, ETSI SPAN12 and the Parlay CC Working group with collaboration from JAIN has been focussed on the MPCC and MMCC APIs. A number of improvements on CC functionality have been made and are reflected in these APIs. For this it was necessary to break the inheritance that previously existed between GCC and MPCC.

The joint CC group has furthermore decided that the MPCC is to be considered as the future base CC family and the technical work will not be continued on GCC. Errors or technical flaws will of course be corrected.

4.1 Call Model Description

The call model used for the Call Control SCFs has the following objects.

* a call object. A call is a relation between a number of parties. The call object relates to the entire call view from the application. E.g., the entire call will be released when a release is called on the call. Note that different applications can have different views on the same physical call, e.g., one application for the originating side and another application for the terminating side. The applications will not be aware of each other, all 'communication' between the applications will be by means of network signalling. The API currently does not specify any feature interaction mechanisms.

* a call leg object. The leg object represents a logical association between a call and an address. The relationship includes at least the signalling relation with the party. The relation with the address is only made when the leg is routed. Before that the leg object is IDLE and not yet associated with the address.

* an address. The address logically represents a party in the call.

* a terminal. A terminal is the end-point of the signalling and/or media for a party. This object type is currently not addressed.

The call object is used to establish a relation between a number of parties by creating a leg for each party within the call.

Associated with the signalling relationship represented by the call leg, there may also be a bearer connection (e.g., in the traditional voice only networks) or a number (zero or more) of media channels (in multi-media networks).

A leg can be attached to the call or detached from the call. When the leg is attached, this means that media or bearer channels related to the legs are connected to the media or bearer channels of the other legs that are attached to the same call. I.e., only legs that are attached can 'speak' to each other. A leg can have a number of states, depending on the signalling received from or sent to the party associated with the leg. Usually there is a limit to the number of legs that are in being routed (i.e., the connection is being established) or connected to the call (i.e., the connection is established). Also, there usually is a limit to the number of legs that can be simultaneously attached to the same call.

Some networks distinguish between controlling and passive legs. By definition the call will be released when the controlling leg is released. All other legs are called passive legs. There can be at most one controlling leg per call. However, there is currently no way the application can influence whether a Leg is controlling or not.

There are two ways for an application to get the control of a call. The application can request to be notified of calls that meet certain criteria. When a call occurs in the network that meets these criteria, the application is notified and can control the call. Some legs will already be associated with the call in this case. Another way is to create a new call from the application.

4.2 Structure of Call Control SCF Documentation

Each of the Call Control SCFs is specified under the following headings:

- The Sequence diagrams give the reader a practical idea of how each of the SCF is implemented.
- The Class relationships clause shows how each of the interfaces applicable to the SCF, relate to one another.
- The Interface specification clause describes in detail each of the interfaces shown within the Class diagram part.
- The State Transition Diagrams (STD) show transition between states in the SCF. The states and transitions are well-defined; either methods specified in the Interface specification or events occurring in the underlying networks cause state transitions.
- The Data definitions clause show a detailed expansion of each of the data types associated with the methods within the classes. Note that some data types are used in other methods and classes and are therefore defined within the Common Data types part of this specification (TS 29.198-2).

4.3 General requirements on support of methods

An implementation of one of the call control APIs which supports or implements a method described in one of the sub-parts of TS 29.198-04, shall support or implement the functionality described for that method, for at least one valid set of values for the parameters of that method.

Where a method is not supported by an implementation of a Service interface, the exception P_METHOD_NOT_SUPPORTED shall be returned to any call of that method.

Where a method is not supported by an implementation of an Application interface, a call to that method shall be possible, and no exception shall be returned.