3GPP TSG CN Plenary, Meeting #11
Palm Springs, USA. 14th - 16th March 2001

Tdoc NP-010133

Source: **CN WG5**

Title: **CRs to Open Service Architecture; Application Programming Interface - Part 1 (29.198)**

Agenda item: **6.5.2**

Document for: **Approval**

Doc-	Doc-	Spec	CR	R	Phas	Subject	Cat	Versio	Versi	Meeti	Workit
NP-010133	N5-000331	29.198	045		R99	**Correction of IDL implementation of data-type TpDomainID**	F	3.2.0	3.3.0	N5-08	OSA
NP-010133	N5-010141	29.198	046		R99	**Correction to terminal capability parameter reference**	F	3.2.0	3.3.0	N5-10	OSA

Note also that

Open Service Architecture (OSA)

is now

Open Service Access (OSA)

Consequently,

the Titles of **TS 29.198** and **TR 29.998**,
the **Abbreviations** in all the documents mentioning OSA,
the 3GPP Vocabulary (**TR 21.905**)

should be changed accordingly.

SA#11 will also be informed about this change.

3GPP Meeting CN5 #8
Scottsdale, Arizona, USA, 18 – 20 December 2000

Document N5-000331
e.g. for 3GPP use the format TP-99xxx
or for SMG, use the format P-99-xxx

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

29.198	**CR**	**045**	Current Version:	**3.2.0**

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ *↑ CR number as allocated by MCC support team*

For submission to:	**CN#11**	for approval	**X**	strategic		*(for SMG use only)*
list expected approval meeting # here ↑		for information		non-strategic		

Form: CR cover sheet, version 2 for 3GPP and SMG *The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc*

Proposed change affects: *(at least one should be marked with an X)* (U)SIM [ ] ME [ ] UTRAN / Radio [ ] Core Network [X]

Source: CN5 **Date:** 20/12/2000

Subject: **Correction of IDL implementation of data-type TpDomainID**

Work item: OSA

Category: *(only one category shall be marked with an X)*

F	Correction	**X**	**Release:**	Phase 2	
A	Corresponds to a correction in an earlier release			Release 96	
B	Addition of feature			Release 97	
C	Functional modification of feature			Release 98	
D	Editorial modification			Release 99	**X**
				Release 00	

Reason for change: The data-type TpDomainID, defined as Tagged Choice of Data elements, is implemented in IDL as struct, where it should be a union.

Clauses affected: 9.2.1

Other specs affected:

Other 3G core specifications	[ ]	→ List of CRs:
Other GSM core specifications	[ ]	→ List of CRs:
MS test specifications	[ ]	→ List of CRs:
BSS test specifications	[ ]	→ List of CRs:
O&M specifications	[ ]	→ List of CRs:

Other comments:

help.doc

<--------- double-click here for help and instructions on how to create a CR.

9.2.1 Common Data Types for the Framework

```
#include <OSA.idl>

module org{

module threegpp{

module osa{

module fw{


typedef TpString     TpClientAppID;        // Identifies the client appl to the framework.

typedef sequence     <TpClientAppID> TpClientAppIDList;
            /* Defines either the framework or the type of entity attempting to access the framework
            The framework
            A client application
            An enterprise operator
            A registered service
            A service supplier */
            enum TpDomainIDType
            {
                P_FW,
                P_CLIENT_APPLICATION,
                P_ENT_OP,
                P_REGISTERED_SERVICE,
                P_SERVICE_SUPPLIER
            };

typedef TpString TpEntOpID;

typedef sequence < TpEntOpID >    TpEntOpIDList;

            typedef TpString TpFwID;

            typedef TpString TpServiceSupplierID;

            /* Defines the Tagged Choice of Data Elements that specify either the framework or the
type of entity
            attempting to access the framework.
            Tag Element Type
            TpDomainIDType */

            struct TpDomainID
            {
                TpFwID FwID;
                TpClientAppID ClientAppID;
                TpEntOpID EntOpID;
                TpServiceID ServiceID;
                TpServiceSupplierID ServiceSupplierID;
            };
            union TpDomainID switch (TpDomainIDType)
            {
                case P_FW:
                  TpFwID FwID;
                case P_CLIENT_APPLICATION:
                  TpClientAppID ClientAppID;
                case P_ENT_OP:
                  TpEntOpID EntOpID;
                case P_REGISTERED_SERVICE:
                  TpServiceID ServiceID;
                case P_SERVICE_SUPPLIER:
                  TpServiceSupplierID ServiceSupplierID;
            };

typedef TpString TpPropertyName;

typedef TpString TpPropertyValue;

typedef sequence < TpProperty > TpPropertyList;

       struct TpProperty {
    TpPropertyName      PropertyName;
    TpPropertyValue        PropertyValue;
       };
```

```
typedef  TpString   TpServiceID;    // A string of characters, generated automatically by the
// Framework and comprising a TpUniqueServiceNumber,
// TpServiceNameString, and a number of relevant
// TpServiceSpecString, concatenated using a forward
// separator (/), that uniquely identifies an instance of a
// SCF interface.

typedef sequence <TpServiceID>      TpServiceIDList;

 typedef TpString       TpServiceNameString;        // Uniquely identifies the name of an SCF
// interface. For OSA release 99 the following
// values have been defined: NULL (no SCF name),
// P_CALL_CONTROL, P_USER_INTERACTION,
// P_USER_LOCATION_CAMEL, P_TERMINAL_CAPABILITIES and
// P_USER_STATUS.

typedef TpString        TpServiceSpecString;        // Uniquely identifies the name of a SCF
// specialization interface. For OSA release 99
// the following values have been defined: NULL
// no SCF specialization) and P_CALL.


typedef TpString        TpUniqueServiceNumber;      // A string of characters that represents a
// unique number.
    enum TpServicePropertyMode {
        NORMAL,                     // The value of the corresponding SCF property type may
optionally be
// provided.
MANDATORY,                  // The value of the corresponding SCF property type must be provided
// at SCF registration.
        READONLY,                   // The value of the corresponding SCF property is optional, nut
once
// given a value it may not be modified.
        MANDATORY_READONLY      // The value of the corresponding SCF property type must be provided
// and may not be modified subsequently.
    };

typedef TpString        TpServicePropertyTypeName;

typedef TpString        TpServicePropertyName;

typedef sequence <TpServicePropertyName>    TpServicePropertyNameList;

typedef TpString        TpServicePropertyValue;

typedef sequence <TpServicePropertyValue>   TpServicePropertyValueList;

    struct TpServiceProperty {              // Describes a SCF property
    TpServicePropertyName       ServicePropertyName;
    TpServicePropertyValueList  ServicePropertyValueList;
    TpServicePropertyMode       ServicePropertyMode;
};

typedef sequence <TpServiceProperty>        TpServicePropertyList;

typedef TpString        TpServiceTypeName;

typedef sequence <TpServiceTypeName>        TpServiceTypeNameList;

struct  TpService {                 // Describes a registered SCF.
        TpServiceID     ServiceID;
        TpServicePropertyList   ServicePropertyList;
    };

typedef sequence <TpService>    TpServiceList;

    struct TpServiceDescription {           // Describes the properties of a registered SCF.
        TpServiceTypeName   ServiceTypeName;
        TpServicePropertyList   ServicePropertyList;
    };

struct TpServiceTypeProperty {                  // Describes a SCF property.
        TpServicePropertyName               ServicePropertyName;
        TpServicePropertyMode       ServicePropertyMode;
        TpServicePropertyTypeName   ServicePropertyTypeName;
    };

typedef sequence <TpServiceTypeProperty>    TpServiceTypePropertyList;
```

```
    struct TpServiceTypeDescription {           // Describes a SCF type.
        TpServiceTypePropertyList           ServiceTypePropertyList;
        TpServiceTypeNameList       ServiceTypeNameList;
        TpBoolean                       EnabledOrDisabled;
    };

};};};};
```

3GPP TSG-CN5 Meeting #10
Antwerp, BELGIUM, 5 – 7 March 2001

Tdoc N5-010141

CR-Form-v3

CHANGE REQUEST

⌘ **29.198** **CR** **046** ⌘ rev **-** ⌘ Current version: **3.2.0** ⌘

*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 terminal capability parameter reference**
Source:	⌘	CN5
Work item code:	⌘	OSA
Date:	⌘	7/03/2001
Category:	⌘	**F**
Release:	⌘	R99

Use one of the following categories:
F *(essential 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:	⌘	A suffix that indicates passing by reference is missing from a parameter name
Summary of change:	⌘	"Ref" suffix has been added to an out parameter to show that it is passed by reference
Consequences if not approved:	⌘	Erroneous specification.

Clauses affected:	⌘	6.8.1		
Other specs affected:	⌘	☐ Other core specifications ☐ Test specifications ☐ O&M Specifications	⌘	
Other comments:	⌘			

6.8.1 Terminal Capabilities SCF interface

The Terminal Capabilities SCF interface `IpTerminalCapabilities` contains the synchronous method `getTerminalCapabilities`. The application has to provide the `terminaIdentity` is input to this method. The result indicates whether or not the terminal capabilities are available in the network and, in case they are, it will return the terminal capabilities (see the data definition of `TpTerminalCapabilities` for more information).

<<Interface>> IpTerminalCapabilities
getTerminalCapabilities(terminalIdentity : in TpString, Result : out TpTerminalCapabilitiesRef) : TpResult