3GPP TSG CN Plenary **Tdoc NP-000622**
Meeting #10, Bangkok, Thailand
6th – 8th December 2000

Source: **TSG CN WG 5**

Title: **CRs to R99 Work Item OSA, 3GPP TS 29.198**

Agenda item: **7.23**

Document for: **APPROVAL**

Introduction:

This document contains **18** CRs on **R99** Work Item **"OSA"**, that have been agreed by **TSG CN WG5**, and are forwarded to TSG CN Plenary meeting #10 for approval.

SPEC	CR	REV	TDoc	PHASE	SUBJECT	CAT	OLD VER
29.198	025		N5-000199	R99	Removal of the originatingAddress from the connectReq method in IpDataSession	F	3.1.0
29.198	026	1	N5-000233	R99	Alignment between new ETSI document for common data and TS29.198	F	3.1.0
29.198	027		N5-000243	R99	Correction of the type TpTerminalCapabilities	F	3.1.0
29.198	028		N5-000245	R99	Incorrect Date and Time example in Data Definitions	F	3.1.0
29.198	029		N5-000246	R99	Double IDL definition for TpGCCSException	F	3.1.0
29.198	030		N5-000247	R99	Parameter EnabledOrDisbled in TpServiceTypeDescription	F	3.1.0
29.198	031		N5-000248	R99	readonly is an IDL keyword	F	3.1.0
29.198	032		N5-000249	R99	Error correction in the Scope definition, section 1	F	3.1.0
29.198	033	1	N5-000298	R99	Missing syntax and semantics description for security parameter	F	3.1.0
29.198	034		N5-000253	R99	Specific exceptions for method invocations in invalid states	F	3.1.0
29.198	035		N5-000254	R99	Unclear default value for TpAccessType	F	3.1.0
29.198	036	1	N5-000299	R99	Unclear description for TpAuthType	F	3.1.0
29.198	037		N5-000256	R99	TpInterfaceName in method obtainInterface()	F	3.1.0
29.198	038		N5-000262	R99	Correction on numbering in TpCallAppInfoType	F	3.1.0
29.198	039		N5-000263	R99	Addition of MonitorMode in TpCallEventInfo	F	3.1.0
29.198	040		N5-000264	R99	Renaming of P_CALL_REPORT_REFUSED_BUSY	D	3.1.0
29.198	043		N5-000292	R99	Removal of the parameter serviceProperties in the method selectService	F	3.1.0
29.198	044		N5-000297	R99	Inclusion of missing state transitions in case call related information could not be retrieved.	F	3.1.0

3GPP CN WG5 Meeting #6
Vienna, Austria, 17-19 October 2000

***Document* N5-000199**
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	025	Current Version:	3.1.0

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

For submission to:	CN#10	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: Nokia **Date:** 11th October 2000

Subject: Removal of the originatingAddress from the connectReq method in IpDataSession

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 stage 2 document TS 23.127 does not mention originating address parameter. Furthermore there is no mapping to CAMEL nor other identified reason to have this parameter in Data Session Control SCF which is used towards the GPRS node SGSN. The parameter is also omitted in TR 29.998.

Clauses affected: 6.5.1.4, 9.5

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.

6.5.1.4 IpDataSession

<<Interface>> IpDataSession
connectReq(dataSessionID : in TpSessionID , responseRequested : in TpDataSessionReportRequestSet , targetAddress : in TpAddress , ~~originatingAddress: in TpAddress,~~ assignmentID : out TpAssignmentIDRef) : TpResult release(dataSessionID : in TpSessionID , cause : in TpDataSessionReleaseCause) : TpResult superviseDataSessionReq(dataSessionID : in TpSessionID, treatment : in TpDataSessionSuperviseTreatment , bytes : in TpDataSessionSuperviseVolume) : TpResult setDataSessionChargePlan(dataSessionID: in TpSessionID, dataSessionChargePlan: in TpDataSessionChargePlan): TpResult setAdviceOfCharge(dataSessionID : in TpSessionID, aoCInfo : in TpAoCInfo, tariffSwitch : in TpDuration): TpResult

```
/* This interface is the SCF manager' interface for Data Session Control. */
interface IpDataSessionControlManager : IpService
{
   /* This method is used to enable data session notifications. */
   void enableDataSessionNotification (
   in IpAppDataSessionControlManager appInterface,
   in TpDataSessionEventCriteria eventCriteria,
   out TpAssignmentID assignmentID)
   raises (TpDSCSException, TpGeneralException);

/* This method is used by the application to disable data session notifications.*/
void disableDataSessionNotification
(
   in TpAssignmentID assignmentID)
   raises (TpDSCSException, TpGeneralException);
};

/* This interface provides the means to control a data session. */
interface IpDataSession : IpService
{
   /* This method requests connection of the data session to the destination party.*/
   void connectReq (
   in TpSessionID dataSessionID,
   in TpDataSessionReportRequestSet responseRequested,
   in TpAddress targetAddress,
   in TpAddress originatingAddress,
   out TpAssignmentID assignmentID)
   raises (TpDSCSException, TpGeneralException);
```

3GPP Meeting CN5 #5
Vienna, 17-18 Oct 2000

Document N5-000233
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	026	Current Version:	3.1.0

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

For submission to:	CN#10	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:	Ericsson		**Date:**	8 Nov 2000	
Subject:	Alignment between new ETSI document for common data and TS29.198				
Work item:	OSA				

Category: *(only one category shall be marked with an X)*			**Release:**		
F	Correction	X		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:	During the review of the new ETSI document for the common data and the TS29.198 some misalignments in the data-types were discovered.
Clauses affected:	8.1.4.8, 8.1.4.12, 8.1.4.17, 9.1

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.

8.1.4.8 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_APPLICATION_ID~~ P_INVALID_DOMAIN_ID	0001h	Invalid ~~application~~ client ID
~~P_INVALID_CLIENT_CAPABILITY~~ P_INVALID_AUTH_CAPABILITY	0002h	Invalid authentication ~~client~~ capability
P_INVALID_AGREEMENT_TEXT	0003h	Invalid agreement text
P_INVALID_SIGNING_ALGORITHM	0004h	Invalid signing algorithm
~~P_INVALID_INTERFACE_ID~~ P_INVALID_INTERFACE_NAME	0005h	Invalid interface name~~ID~~
P_INVALID_SERVICE_ID	0006h	Invalid service capability feature ID
P_INVALID_EVENT_TYPE	0007h	Invalid event type
P_SERVICE_NOT_ENABLED	0008h	The service capability feature ID does not correspond to a SCF that has been enabled
P_INVALID_ASSIGNMENT_ID	0009h	The assignment ID d~~oes not correspond to one of the valid assignment Ids~~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 required 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
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 has been refused because no callback address is set
~~P_INVALID_TERMINATION_TEXT~~	~~0012h~~	~~Invalid termination text~~
P_INVALID_SIGNATURE	0012h	Invalid digital signature
P_INVALID_SERVICE_TOKEN	0013h	The service capability feature token does not correspond to a token that had been issued, or the issued token has expired
~~P_INVALID_AUTHENTICATION~~	~~0014h~~	~~The client has not been correctly authenticated~~
~~P_INVALID_SERVICE_PROPERTY~~	~~0015h~~	~~Invalid service capability feature property~~
~~P_METHOD_NOT_SUPPORTED~~	~~001Ah~~	~~The method is not allowed or supported within the context of the current SCF agreement.~~
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.
General security errors		
P_USER_NOT_SUBSCRIBED	0030h	A service (or application) is unauthorised to access information and request SCFs with regards to users that are **not** subscribed to it.
P_APPLICATION_NOT_ACTIVATED	0031h	A service (or application) is unauthorised to access information and request SCFs with regards to its subscribed users that have

		deactivated that particular service (or application).
P_USER_PRIVACY	0032h	A service (or application) is unauthorised to access information and request an SCF with regards to its subscribed users that have **set** their privacy flag regarding that particular SCF.
P_GCCS_SERVICE_INFORMATION_MISSING	0100h	Information relating to the Call Control SCF could not be found
P_GCCS_SERVICE_FAULT_ENCOUNTERED	0101h	Fault detected in the Call Control SCF
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 OSA 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_GUIS_INVALID_CRITERIA	0300h	Invalid criteria specified
P_GUIS_ILLEGAL_ID	0301h	Information id specified is invalid
P_GUIS_ID_NOT_FOUND	0302h	A legal information id is not known to the User Interaction SCF
P_GUIS_ILLEGAL_RANGE	0303h	The values for minimum and maximum collection length are out of range.
P_GUIS_INVALID_COLLECTION_CRITERIA	0304h	Invalid collection criteria specified
P_GUIS_INVALID_NETWORK_STATE	0305h	Although the sequence of method calls is allowed by the OSA 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_GUIS_UNEXPECTED_SEQUENCE	0306h	Unexpected sequence of methods, i.e., the sequence does not match the specified state diagrams.
P_DSCS_SERVICE_INFORMATION_MISSING	0400h	Information relating to the Data Session Control SCF could not be found
P_DSCS_SERVICE_FAULT_ENCOUNTERED	0401h	Fault detected in the Data Session Control SCF
P_DSCS_UNEXPECTED_SEQUENCE	0402h	Unexpected sequence of methods, i.e., the sequence does not match the specified state diagrams for the data session.
P_DSCS_INVALID_ADDDRESS	0403h	Invalid address specified
P_DSCS_INVALID_STATE	0404h	Invalid state specified
P_DSCS_INVALID_CRITERIA	0405h	Invalid criteria specified
P_DSCS_INVALID_NETWORK_STATE	0406h	Although the sequence of method calls is allowed by the OSA gateway, the underlying protocol can not support it.

8.1.4.12 TpAddress

Defines the structure of data elements that specifies an address.

Structure Member Name	Structure Member 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	"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	"webmaster@parlay.org"
P_ADDRESS_PLAN_X400	The X400 address structured as a set of attibute value pairs separated by semicolons.	"C=nl;ADMD= ;PRMD=uninet;O=parlay;S=Doe;I=S;G =John'

8.1.4.17 TpAddressScreening

Defines whether an address has been screened by the application.

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 (Note that 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)

9.1 Generic IDL

```
#ifndef __OSA_DEFINED
#define __OSA_DEFINED

module org
{
    module threegpp
    {
        module osa
        {
..
            typedef sequence < TpSessionID> TpSessionIDSet;

      exception TpGeneralException
    {
      TpInt32 exceptionType;

    };

      const TpInt32 P_RESULT_INFO_UNDEFINED = 0;
      const TpInt32 P_INVALID_DOMAIN_ID = 1;
```

```
        const TpInt32 P_INVALID_AUTH_CAPABILITY = 2;
        const TpInt32 P_INVALID_AGREEMENT_TEXT = 3;
        const TpInt32 P_INVALID_SIGNING_ALGORITHM = 4;
        const TpInt32 P_INVALID_INTERFACE_NAME = 5;
        const TpInt32 P_INVALID_SERVICE_ID = 6;
        const TpInt32 P_INVALID_EVENT_TYPE = 7;
        const TpInt32 P_SERVICE_NOT_ENABLED = 8;
        const TpInt32 P_INVALID_ASSIGNMENT_ID = 9;
        const TpInt32 P_INVALID_PARAMETER = 10;
        const TpInt32 P_INVALID_PARAMETER_VALUE = 11;
        const TpInt32 P_PARAMETER_MISSING = 12;
        const TpInt32 P_RESOURCES_UNAVAILABLE = 13;
        const TpInt32 P_TASK_REFUSED = 14;
        const TpInt32 P_TASK_CANCELLED = 15;
        const TpInt32 P_INVALID_DATE_TIME_FORMAT = 16;
        const TpInt32 P_NO_CALLBACK_ADDRESS_SET = 17;
        const TpInt32 P_INVALID_SIGNATURE = 18;
        const TpInt32 P_INVALID_SERVICE_TOKEN = 19;
        const TpInt32 P_ACCESS_DENIED = 20;
        const TpInt32 P_INVALID_PROPERTY = 21;
        const TpInt32 P_METHOD_NOT_SUPPORTED = 22;
        const TpInt32 P_NO_ACCEPTABLE_AUTH_CAPABILITY = 23;
        const TpInt32 P_INVALID_INTERFACE_TYPE = 24;
        const TpInt32 P_SERVICE_ACCESS_TYPE = 25;
        const TpInt32 P_SERVICE_ACCESS_DENIED = 26;
        const TpInt32 P_USER_NOT_SUBSCRIBED = 48;
        const TpInt32 P_APPLICATION_NOT_ACTIVATED = 49;
        const TpInt32 P_USER_PRIVACY = 50;

            // Defines the general Parlay exception values
            enum TpGeneralExceptionType
            {
                P_RESULT_INFO_UNDEFINED,     // No further information present
                P_INVALID_APPLICATION_ID,    // Invalid application ID
                P_INVALID_CLIENT_CAPABILITY,// Invalid client capability
                P_INVALID_AGREEMENT_TEXT,    // Invalid agreement text
                P_INVALID_SIGNING_ALGORITHM,// Invalid signing algorithm
                P_INVALID_INTERFACE_NAME,    // Invalid interface name
                P_INVALID_SERVICE_ID,        // Invalid service capability feature ID
                P_INVALID_EVENT_TYPE,        // Invalid event type
                P_SERVICE_NOT_ENABLED,       // The SCF ID does not correspond
                                             // to a SCF that has been enabled
                P_INVALID_ASSIGNMENT_ID,     // The assignment ID does not
                                             // correspond to one of the valid assignment IDs
                P_INVALID_PARAMETER,         // The method has been called with an
                                             // invalid parameter
                P_INVALID_PARAMETER_VALUE,  // A method parameter has an invalid value
                P_PARAMETER_MISSING,         // A required parameter has not been
                                             // specified in the method call
                P_RESOURCES_UNAVAILABLE,     // The required resources in the
                                             // network are not available
                P_TASK_REFUSED,              // The requested method has been refused
                P_TASK_CANCELLED,            // The requested method has been cancelled
                P_INVALID_DATE_TIME_FORMAT, // Invalid date and time format provided
                P_NO_CALLBACK_ADDRESS_SET,  // The requested method has been refused
                                             // because no callback address is set
                P_INVALID_TERMINATION_TEXT, // Invalid termination text
                P_INVALID_SERVICE_TOKEN,     // The SCF token does not correspond to a
                                             // token that had been issued, or the issued token
                                             // has expired.
                P_INVALID_AUTHENTICATION,    // The client has not been correctly authenticated
                P_INVALID_SERVICE_PROPERTY, // Invalid service capability feature property.
                P_METHOD_NOT_SUPPORTED       // The method is not allowed or supported within
                                             // the context of the current SCF agreement.
            };

            exception TpGeneralException
            {
                TpGeneralExceptionType exceptionType;
            };

            // Defines the GCCS OSA exception values
            enum TpGCCSExceptionType
            {
                P_GCCS_SERVICE_INFORMATION_MISSING,// Information relating to the Call
                                                   // Control SCF could not be found
                P_GCCS_SERVICE_FAULT_ENCOUNTERED,  // Fault detected in the Call Control SCF
                P_GCCS_UNEXPECTED_SEQUENCE,  // 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,        // Invalid address specified
                P_GCCS_INVALID_CRITERIA,        // Invalid criteria specified
                P_GCCS_INVALID_NETWORK_STATE,// Although the sequence of method calls is
                                                // allowed by the OSA 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.

            };

            exception TpGCCSException
            {
                TpGCCSExceptionType exceptionType;
            };

            // Defined the GUIS OSA exception values
            enum TpGUISExceptionType
            {
                P_GUIS_INVALID_CRITERIA, // Invalid criteria specified
                P_GUIS_ILLEGAL_ID,       // Information id specified is invalid
                P_GUIS_ID_NOT_FOUND,     // A legal information id is not known to the User
                                         // Interaction SCF
                P_GUIS_ILLEGAL_RANGE,    // The values for minimum and maximum collection
                                         // length are out of range.
                P_GUIS_INVALID_COLLECTION_CRITERIA, // Invalid collection criteria specified
                P_GUIS_NETWORK_DEASSIGN, // The relation between the network and the OSA
                                         // gateway is terminated. Therefore, the gateway
                                         // can no longer perform UI operations. This can
                                         // happen after the last requested report is sent
                                         // to the application. To prevent this error, the
                                         /// application should ensure that it has requested
                                         // events which are not yet reported.
                P_GUIS_INVALID_NETWORK_STATE // Although the sequence of method calls is
                                                // allowed by the OSA 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.
            };

            exception TpGUISException
            {
                TpGUISExceptionType exceptionType;
            };
```

..

9.3.1 Common Data Types for Call Control

```
// source file: CC.idl
// Generic Call Data description

#ifndef __OSA_CC_DEFINED
#define __OSA_CC_DEFINED

#include <OSA.idl>
#include <UI.idl>

module org
{
    module threegpp
    {
        module osa
        {
            module cc
            {

..
                /* Define the possible Exceptions. */
                const TpInt32 P_GCCS_SERVICE_INFORMATION_MISSING = 256;
                const TpInt32 P_GCCS_SERVICE_FAULT_ENCOUNTERED = 257;
                const TpInt32 P_GCCS_UNEXPECTED_SEQUENCE = 258;
```

```
                const TpInt32 P_GCCS_INVALID_ADDDRESS = 259;
                const TpInt32 P_GCCS_INVALID_CRITERIA = 2601;
                const TpInt32 P_GCCS_INVALID_NETWORK_STATE = 2612;


                exception TpGCCSException
                {
                    TpInt32 exceptionType;
                };

..

                }; // end module cc
            }; // end module osa
        }; // end module threegpp
    }; // end module org

#endif

// END file CC.idl
```

9.4.1 Common data types for User Interaction

```
// source file: UI.idl
// User Interaction data description

#ifndef __OSA_UI_DEFINED
#define __OSA_UI_DEFINED

#include <OSA.idl>

module org {
 module threegpp {
  module osa {
   module ui {

..
    /* Define the possible Exceptions. */
    exception TpGUISException {
        TpInt32 exceptionType;
    };

    const TpInt32 P_GUIS_INVALID_CRITERIA = 768;            /* Invalid criteria specified */
    const TpInt32 P_GUIS_ILLEGAL_ID = 769;              /* Information id specified is invalid
*/
    const TpInt32 P_GUIS_ID_NOT_FOUND = 770;                /* Information id is not known to
the User Interaction Service */
    const TpInt32 P_GUIS_ILLEGAL_RANGE = 771;               /* The values for minimum and
maximum collection length are out of range */
    const TpInt32 P_GUIS_INVALID_COLLECTION_CRITERIA = 772; /* Invalid collection criteria
specified */
    const TpInt32 P_GUIS_INVALID_NETWORK_STATE = 7734;       /* Although the sequence of
method calls is allowed by the gateway, the underlying protocol can not support it. */
 const TpInt32 P_GUIS_UNEXPECTED_SEQUENCE = 7745;        /* Although the sequence of method
calls is allowed by the gateway, the underlying protocol can not support it. */

   }; // end module ui
  }; // end module osa
 }; // end module threegpp
}; // end module org

#endif

// END file UI.idl
```

3GPP TSG-CN WG5 Meeting #7
Sophia Antipolis, France, 7th – 8th November, 2000

***Document* N5-000243**
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	027	Current Version:	3.1.0

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

For submission to: TSG-CN#10 *list expected approval meeting # here ↑*

for approval [X] for information [ ]

strategic [ ] non-strategic [ ] *(for SMG use only)*

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: Nokia **Date:** 3rd November 2000

Subject: Correction of the type TpTerminalCapabilities

Work item: OSA

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

F	Correction	[X]
A	Corresponds to a correction in an earlier release	[ ]
B	Addition of feature	[ ]
C	Functional modification of feature	[ ]
D	Editorial modification	[ ]

Release:

Phase 2	[ ]
Release 96	[ ]
Release 97	[ ]
Release 98	[ ]
Release 99	[X]
Release 00	[ ]

Reason for change: The type TpServicePropertyList is used to define the properties of an SCF. It is not appropriate to use this for the terminal capabilities which are described as: "CC/PP headers as specified in W3C [12] and adopted in the WAP UAProf specification [13]. It contains URLs; terminal attributes and values, in RDF format; or a combination of both". This error seems to be only in the data definitions, because the IDL already defines this type as TpString which is a proper type for this purpose.
Also the reference numbers have been corrected.

Clauses affected: 8.8.2.2, 9.7

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.

8.2.1.23 TpServicePropertyMode

This type is left as a placeholder but is not used in release 99.This defines SCF property modes.

Name	Value	Documentation
NORMAL	0	The value of the corresponding SCF property type may optionally be provided
MANDATORY	1	The value of the corresponding SCF property type must be provided at service registration time
READONLY	2	The value of the corresponding SCF property type is optional, but once given a value it may not be modified
MANDATORY_READONLY	3	The value of the corresponding SCF property type must be provided and subsequently it may not be modified.

8.2.1.24 TpServicePropertyTypeName

This data type is identical to TpString and describes a valid SCF property name. The valid SCF property names are listed in the SCF data definition.

8.2.1.25 TpServicePropertyName

This data type is identical to TpString. It defines a valid SFC property name. Valid SCF property names are listed in the SCF data definition.

8.2.1.26 TpServicePropertyNameList

This data type defines a Numbered Set of Data Elements of type TpServicePropertyName.

8.2.1.27 TpServicePropertyValue

This data type is identical to TpString and describes a valid value of a SCF property. The valid SCF property values are given in the SCF data definition.

8.2.1.28 TpServicePropertyValueList

This data type defines a Numbered Set of Data Elements of type TpServicePropertyValue

8.2.1.29 TpServiceProperty

This data type is a Sequence of Data Elements which describes an “SCF property”. It is a structured data type which consists of:

Sequence Element Name	Sequence Element Type	Documentation
ServicePropertyName	TpServicePropertyName	
ServicePropertyValueList	TpServicePropertyValueList	
ServicePropertyMode	TpServicePropertyMode	

8.2.1.30 TpServicePropertyList

This data type defines a Numbered Set of Data Elements of type TpServiceProperty.

8.8 Terminal Capabilities Data Definitions

8.8.1 Interface Definitions

8.8.1.1 IpTerminalCapabilities

Defines the address of an IpTerminalCapabilities Interface.

8.8.1.2 IpTerminalCapabilitiesRef

Defines a reference to type IpTerminalCapabilities

8.8.2 Terminal Capabilities Data Definitions

The constants and types defined in the following sections are defined in the *org.threegpp.osa.termcap* package.

8.8.2.1 terminalIdentity

Identifies the terminal.

Name	Type	Documentation
`terminalIdentity`	TpString	Identifies the terminal. It may be a logical address known by the WAP Gateway/PushProxy.

8.8.2.2 TpTerminalCapabilities

This data type is a Sequence_of_Data_Elements that describes the terminal capabilities. It is a structured type that consists of:

Sequence Element Name	Sequence Element Type	Documentation
`StatusCode`	TpBoolean	Indicates whether or not the terminalCapabilities are available.
`TerminalCapabilities`	TpStringServicePropertyList	Specifies the latest available capabilities of the user´s terminal. This information, if available, is returned as CC/PP headers as specified in W3C [612] and adopted in the WAP UAProf specification [139]. It contains URLs; terminal attributes and values, in RDF format; or a combination of both.

8.8.2.3 TpTerminalCapabilitiesError

Defines an error that is reported by the Terminal Capabilities SCF.

Name	Value	Description
`P_TERMCAP_ERROR_UNDEFINED`	0	Undefined.
`P_TERMCAP_INVALID_TERMINALID`	1	The request can not be handled because the terminal id specified is not valid.
`P_TERMCAP_SYSTEM_FAILURE`	2	System failure. The request cannot be handled because of a general problem in the terminal capabilities service or the underlying network.

9.7 Terminal Capabilities: TERMCAP.idl

```
#ifndef __TERMCAP_DEFINED
#define __TERMCAP_DEFINED

#include <OSA.idl>

module org {
module threegpp {
module osa {
module termcap {

   enum TpTerminalCapabilitiesError {
        P_TERMCAP_ERROR_UNDEFINED,      /* Undefined */
        P_TERMCAP_INVALID_TERMINALID,       /* Terminal ID not valid */
        P_TERMCAP_SYSTEM_FAILURE        /* General problem in terminal capabilities SCF or in
underlying network */

      };

     exception TpTermCapException {
         TpTerminalCapabilitiesError error;
   };

    /* TpTerminalCapabilities: Structure containing status code and terminal
    capabilities. */
    struct TpTerminalCapabilities {
        /* statusCode: Indicates whether or not the terminalCapabilities
        are available. */
        TpBoolean StatusCode;
        /* terminalCapabilities: Specifies the latest available capabilities of the user´s terminal.
This information, if available, is returned as CC/PP headers as specified in W3C [126] and adopted
in the WAP UAProf specification [139]. It contains URLs; terminal attributes and values, in RDF
format; or a combination of both.  */
        TpString TerminalCapabilities;
    };

    interface IpTerminalCapabilities : IpService {
        /*  Method: getTerminalCapabilities()
            This method is used by an application to get the capabilities of a
            user's terminal. Direction: Application to Network

            In parameter TerminalIdentity: Identifies the terminal. It may be
            a logical address known by the WAP Gateway/PushProxy.
            Out parameter, see TerminalCapabilityStruct*/
        void getTerminalCapabilities (
            in TpString terminalIdentity,
            out TpTerminalCapabilities result
            )
         raises (TpTermCapException, TpGeneralException);

    };


};};};};

#endif
```

3GPP Meeting CN5 #7
Sophia Antipolis, France, 7-8 November 2000

Document N5-000245
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	028	Current Version:	3.1.0

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

For submission to:	CN#10	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: Lucent Technologies **Date:** 6 November 2000

Subject: Incorrect Date and Time example in Data Definitions

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: Incorrect Date and Time example in Data Definitions.

Clauses affected: 8.1.4.11

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.

8.1.4.11 TpDateAndTime

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

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)**

A colon character separates the time elements (**:**). 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 (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

3GPP Meeting CN5 #7
Sophia Antipolis, France, 7-8 November 2000

Document N5-000246
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	029	Current Version:	3.1.0

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

For submission to:	CN#10	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: Lucent Technologies **Date:** 6 November 2000

Subject: Double IDL definition for TpGCCSException

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 IDL specification in 3G TS 29.198 contains two definitions of TpGCCSException, once in package org.threegpp.osa and once in org.threegpp.osa.cc.

Clauses affected: 9.1

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.

9.1 Generic IDL

```
#ifndef __OSA_DEFINED
#define __OSA_DEFINED

module org
{
   module threegpp
   {
      module osa
      {
         /*******************************************************************/
         //                          Primitive data types
         /*******************************************************************/

         typedef boolean   TpBoolean;   // Defines a Boolean data type
         typedef long      TpInt32;     // Defines a signed 32 bit integer
         typedef float     TpFloat;     // Defines a single precision real number.
         typedef string    TpString;    // Defines a string comprising length and data.

         // Primitive based OSA datatypes

         typedef TpInt32    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 default
                                  // duration.
         typedef TpInt32   TpSessionID;  // Defines a network unique session ID. OSA
                                  // 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
                                  // OSA service capability feature, the sessionIDs are unique
                                  // only in the context of a manager instantiation (e.g., within
                                  // the context of one generic call control manager). As such
                                  // if an application creates two instances of the same SCF
                                  // manager it shall use different instantiations of the
                                  // callback objects which implement the callback interfaces.
         typedef TpInt32 TpAssignmentID; // This data type is identical to a TpInt32. It
                                  // specifies a number which identifies an individual
                                  // event notification enabled by the application or
                                  // OSA service capability feature.

         typedef sequence < TpSessionID> TpSessionIDSet;
```

```
// Defines the general Parlay exception values
enum TpGeneralExceptionType
{
   P_RESULT_INFO_UNDEFINED,    // No further information present
   P_INVALID_APPLICATION_ID,   // Invalid application ID
   P_INVALID_CLIENT_CAPABILITY,// Invalid client capability
   P_INVALID_AGREEMENT_TEXT,   // Invalid agreement text
   P_INVALID_SIGNING_ALGORITHM,// Invalid signing algorithm
   P_INVALID_INTERFACE_NAME,   // Invalid interface name
   P_INVALID_SERVICE_ID,       // Invalid service capability feature ID
   P_INVALID_EVENT_TYPE,       // Invalid event type
   P_SERVICE_NOT_ENABLED,      // The SCF ID does not correspond
                       // to a SCF that has been enabled
   P_INVALID_ASSIGNMENT_ID,    // The assignment ID does not
                       // correspond to one of the valid assignment IDs
   P_INVALID_PARAMETER,        // The method has been called with an
                       // invalid parameter
   P_INVALID_PARAMETER_VALUE,  // A method parameter has an invalid value
   P_PARAMETER_MISSING,        // A required parameter has not been
                       // specified in the method call
   P_RESOURCES_UNAVAILABLE,    // The required resources in the
                       // network are not available
   P_TASK_REFUSED,             // The requested method has been refused
   P_TASK_CANCELLED,           // The requested method has been cancelled
   P_INVALID_DATE_TIME_FORMAT, // Invalid date and time format provided
   P_NO_CALLBACK_ADDRESS_SET,  // The requested method has been refused
                       // because no callback address is set
   P_INVALID_TERMINATION_TEXT, // Invalid termination text
   P_INVALID_SERVICE_TOKEN,    // The SCF token does not correspond to a
                       // token that had been issued, or the issued token
                       // has expired.
   P_INVALID_AUTHENTICATION,   // The client has not been correctly authenticated
   P_INVALID_SERVICE_PROPERTY, // Invalid service capability feature property.
   P_METHOD_NOT_SUPPORTED      // The method is not allowed or supported within
                       // the context of the current SCF agreement.
};

exception TpGeneralException
{
   TpGeneralExceptionType exceptionType;
};
```

```
        // Defines the GCCS OSA exception values
        enum TpGCCSExceptionType
        {
            P_GCCS_SERVICE_INFORMATION_MISSING,// Information relating to the Call
                                              // Control SCF could not be found
            P_GCCS_SERVICE_FAULT_ENCOUNTERED,  // Fault detected in the Call Control SCF
            P_GCCS_UNEXPECTED_SEQUENCE,  // 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,     // Invalid address specified
            P_GCCS_INVALID_CRITERIA,     // Invalid criteria specified
            P_GCCS_INVALID_NETWORK_STATE,// Although the sequence of method calls is
                                         // allowed by the OSA 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.

        };

        exception TpGCCSException
        {
            TpGCCSExceptionType exceptionType;
        };

        // Defined the GUIS OSA exception values
        enum TpGUISExceptionType
        {
            P_GUIS_INVALID_CRITERIA, // Invalid criteria specified
            P_GUIS_ILLEGAL_ID,       // Information id specified is invalid
            P_GUIS_ID_NOT_FOUND,     // A legal information id is not known to the User
                                     // Interaction SCF
            P_GUIS_ILLEGAL_RANGE,    // The values for minimum and maximum collection
                                     // length are out of range.
            P_GUIS_INVALID_COLLECTION_CRITERIA, // Invalid collection criteria specified
            P_GUIS_NETWORK_DEASSIGN, // The relation between the network and the OSA
                                     // gateway is terminated. Therefore, the gateway
                                     // can no longer perform UI operations. This can
                                     // happen after the last requested report is sent
                                     // to the application. To prevent this error, the
```

```
                        /// application should ensure that it has requested
                        // events which are not yet reported.
   P_GUIS_INVALID_NETWORK_STATE // Although the sequence of method calls is
                           // allowed by the OSA 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.
};

exception TpGUISException
{
   TpGUISExceptionType exceptionType;
};
```

3GPP Meeting CN5 #7
Sophia Antipolis, France, 7-8 November 2000

Document N5-000247
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	030	Current Version:	3.1.0

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

For submission to:	CN#10	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: Lucent Technologies **Date:** 6 November 2000

Subject: Parameter EnabledOrDisbled in TpServiceTypeDescription

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: It is not clear what the value of the boolean parameter should mean by just looking at the name of the parameter “enabledOrDisabled”.

Clauses affected: 8.2.1.24

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.

8.2.1.24 TpServiceTypeDescription

This type is left as a placeholder but is not used in release 99.

This data type is a Sequence_of_Data_Elements which describes an SCF type. It is a structured data type. It consists of:

Sequence Element Name	Sequence Element Type	Documentation
ServiceTypePropertyList	TpServiceTypePropertyList	a sequence of property name and property mode tuples associated with the SCF type
ServiceTypeNameList	TpServiceTypeNameList	the names of the super types of the associated SCF type
EnabledOrDisabled	TpBoolean	an indication whether the SCF type is enabled (true) or disabled (false)

3GPP Meeting CN5 #7
Sophia Antipolis, France, 7-8 November 2000

Document N5-000248
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	031	Current Version:	3.1.0

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

For submission to:	CN#10	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: Lucent Technologies **Date:** 6 November 2000

Subject: "readonly" is an IDL keyword

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 IDL keyword "readonly" should not be used as a name for a parameter.

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.

```
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.
};
```

3GPP Meeting CN5 #7
Sophia Antipolis, France, 7-8 November 2000

Document N5-000249
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	032	Current Version:	3.1.0

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

For submission to:	CN#10	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: Lucent Technologies **Date:** 6 November 2000

Subject: Error correction in the Scope definition, section 1

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 scope still contains an erroneous reference to a section on SDL definitions.

Clauses affected: 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.

The stage 3 documentation of the OSA R'99 API consists of two parts:

- **The API specification (Part 1).**
  This is a normative stage 3 specification of the capabilities of the OSA R'99 API and describes the OSA API interface classes, containing class diagrams (see section 6), state transition diagrams (see section 7), ~~SDLs (see section 8),~~ data type definitions (section 8~~9~~), and the IDLs (see section 9~~10~~).

3GPP Meeting CN5 #7
Sophia Antipolis, France, 7-8 November 2000

Document N5-000253
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	034	Current Version:	3.1.0

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

For submission to:	CN#10	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: Lucent Technologies **Date:** 6 November 2000

Subject: Specific exceptions for method invocations in invalid states

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: Section 7 of 3G TS 29.198 states that methods that are invoked in an irrelevant state will cause an exception to be thrown. However, it is not specified which exception specifically

Clauses affected: 7

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.

7 State Transition Diagrams

This section contains the State Transition Diagrams for the objects that implement the interfaces on the gateway side. The State Transition Diagrams show the behaviour of these objects. For each state the methods that can be invoked by the application are shown. Methods not shown for a specific state are not relevant for that state and will return ~~an~~ the P_TASK_REFUSED exception. Apart from the methods that can be invoked by the application also events internal to the gateway or related to network events are shown together with the resulting event or action performed by the gateway. These internal events are shown between quotation marks.

3GPP Meeting CN5 #7
Sophia Antipolis, France, 7-8 November 2000

Document N5-000254
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	035	Current Version:	3.1.0

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

For submission to:	CN#10	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: Lucent Technologies **Date:** 6 November 2000

Subject: Unclear default value for TpAccessType

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: There is no distinction between the two defined values for TpAccessType. Therefore, NULL will be removed.

Clauses affected: 8.2.2.1, 9.2.3

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.

8.2.2.1 TpAccessType

This data type is identical to a TpString. This identifies the type of access interface requested by the client application. If they request P_ACCESS, then a reference to the IpAccess interface is returned. (Network operators can define their own access interfaces to satisfy client requirements for different types of access. These can be selected using the TpAccessType, but should be preceded by the string "SP_". The following values isare defined for OSA release 99:

String Value	Description
NULL	An empty (NULL) string indicates the default access type
P_ACCESS	Access using the OSA Access Interfaces: IpAccess and IpAppAccess

9.2.3 Trust and Security Management IDL

```
#include <fw.idl>

module org{

module threegpp{

module osa{

module fw{

module trust_and_security{


      /*******************************************************************************/
      //                                Data definitions                             //
      /*******************************************************************************/


   typedef TpString     TpAccessType;     // The type of access interface requested by the client
                                          // application. For OSA release 99 the following values
                                          // hasve been defined: NULL (indicates the default access
                                          // type) and P_ACCESS.


   typedef TpString     TpAuthType;       // The type of authentication mechanism requested by the
                                          // client. For OSA release 99 the following values have
                                          // been defined: NULL (indicates OSA authentication),
                                          // P_AUTHENTICATION (indicates use of the OSA
                                          // authentication interfaces.


   typedef TpString  TpAuthCapability;    // The authentication capabilities that could be supported
                                          // by the OSA. For OSA release 99 the following values
                                          // have been defined: NULL (indicates no client
                                          // capabilities, P_DES_56, P_DES_128, P_RSA_512 and
                                          P_RSA_1024).
```

3GPP Meeting CN5 #7
Sophia Antipolis, France, 7-8 November 2000

Document N5-000256
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	037	Current Version:	3.1.0

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

For submission to:	CN#10	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: Lucent Technologies **Date:** 6 November 2000

Subject: TpInterfaceName in method obtainInterface()

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: Method obtainInterface() of the IpAccess interface can request access to an interface, identified in parameter interfaceName of type TpInterfaceName. A possible value for interfaceName is the NULL string. It is not clear why an application would indicate it wants to obtain access to an interface, and then not specify an interface at all. Value NULL will be removed.

Clauses affected: 8.2.2.5, 9.2.3

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.

8.2.2.5 TpInterfaceName

This data type is identical to a TpString, and is defined as a string of characters that identify the names of the framework SCFs that are be supported by the OSA API. Other Network operator specific SCFs may also be used, but should be preceded by the string "SP_".The following values are defined for OSA release 99.

Character String Value	Description
~~*NULL*~~	~~An empty (NULL) string indicates no interface.~~

9.2.3 Trust and Security Management IDL

```
#include <fw.idl>

module org{

module threegpp{

module osa{

module fw{

module trust_and_security{

      /*******************************************************************************/
      //                                Data definitions                             //
      /*******************************************************************************/

   typedef TpString     TpAccessType;     // The type of access interface requested by the client
                                          // application. For OSA release 99 the following values
                                          // have been defined: NULL (indicates the default access
                                          // type) and P_ACCESS.

   typedef TpString     TpAuthType;       // The type of authentication mechanism requested by the
                                          // client. For OSA release 99 the following values have
                                          // been defined: NULL (indicates OSA authentication),
                                          // P_AUTHENTICATION (indicates use of the OSA
                                          // authentication interfaces.

   typedef TpString  TpAuthCapability;    // The authentication capabilities that could be supported
                                          // by the OSA. For OSA release 99 the following values
                                          // have been defined: NULL (indicates no client
                                          // capabilities, P_DES_56, P_DES_128, P_RSA_512 and
                                       P_RSA_1024).

   typedef TpString     TpAuthCapabilityList;   // A string of multiple TpAuthCapability
                                                // concatenated using a commas.
```

```
struct TpAuthDomain
      {
        TpDomainID DomainID;
       IpOSA AuthInterface;
      };

       typedef TpPropertyList TpEndAccessProperties;


  typedef TpString     TpInterfaceName;  // Identifies the names of the framework SCFs that are be
                                         // supported by the OSA API. For release 99 these are NULL,
                                         // P_DISCOVERY, P_OAM
                                         // P_LOAD_MANAGER,
                                         // P_FAULT_MANAGER,
                                         // P_HEARTBEAT_MANAGEMENT,
                                         // P_REGISTRATION
```

3GPP Meeting CN5 #7
Sophia Antipolis, November 7-8

Document N5-000262
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	038	Current Version:	3.1.0

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

For submission to:	CN#10	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: Ericsson **Date:** 27 October 2000

Subject: Correction to make consecutive numbering for TpCallAppInfoType

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 numbering for TpCallAppInfoType is not consecutive

Clauses affected: 8.3.3.3

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.

8.3.3.3 TpCallAppInfoType

Defines the type of application related call information.

Name	Value	Description
P_CALL_APP_UNDEFINED	0	Undefined
P_CALL_APP_ALERTING_MECHANISM	1	The alerting mechanism or pattern to use
P_CALL_APP_NETWORK_ACCESS_TYPE	2	The network access type (e.g. ISDN)
P_CALL_APP_TELE_SERVICE	3~~4~~	Indicates the tele-service (e.g. speech) and related info such as clearing programme
P_CALL_APP_BEARER_SERVICE	4~~5~~	Indicates the bearer service (e.g. 64kb/s unrestricted data).
P_CALL_APP_PARTY_CATEGORY	5~~6~~	The category of the calling or called party
P_CALL_APP_PRESENTATION_ADDRESS	6~~7~~	The address to be presented to other call parties
P_CALL_APP_GENERIC_INFO	7~~8~~	Carries unspecified application-Service Capability Feature information
P_CALL_APP_ADDITIONAL_ADDRESS	8~~9~~	Indicates an additional address

3GPP Meeting CN5 #7
Sophia Antipolis, November 7-8

Document N5-000263
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	039	Current Version:	3.1.0

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

For submission to:	CN#10	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: Ericsson **Date:** 27 October 2000

Subject: Addition of the monitorMode to the callEventInfo parameter

Work item: OSA

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

F	Correction	X
A	Corresponds to a correction in an earlier release	
B	Addition of feature	
C	Functional modification of feature	
D	Editorial modification	

Release:

Phase 2	
Release 96	
Release 97	
Release 98	
Release 99	X
Release 00	

Reason for change: The application should be able to tell from the data received when callEventNotify was invoked whether the monitoring was done in interrupt or in notify mode. Therefore, the monitorMode should be added to the callEventInfo parameter.

Clauses affected: 8.3.2.6, 9.3.1

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.

8.3.2.6 TpCallEventInfo

Defines the Sequence of Data Elements that specify the information returned to the application in a New Call event notification.

Sequence Element Name	Sequence Element Type
DestinationAddress	TpAddress
OriginatingAddress	TpAddress
OriginalDestinationAddress	TpAddress
RedirectingAddress	TpAddress
CallAppInfo	TpCallAppInfoSet
CallEventName	TpCallEventName
CallNotificationType	TpCallNotificationType
MonitorMode	TpCallMonitorMode

9.3 Call Control

9.3.1 Common Data Types for Call Control

```
// source file: CC.idl
// Generic Call Data description

#ifndef __OSA_CC_DEFINED
#define __OSA_CC_DEFINED

..
        //Defines the type of notification.
        //Indicates whether it is related to the originating of the terminating user in the call.
        struct TpCallEventInfo
        {
           TpAddress DestinationAddress;
           TpAddress OriginatingAddress;
           TpAddress OriginalDestinationAddress;
           TpAddress RedirectingAddress;
           TpCallAppInfoSet CallAppInfo;
           TpCallEventName CallEventName;
           TpCallNotificationType CallNotificationType;
           TpCallMonitorMode MonitorMode;
        };

..

        }; // end module cc
      }; // end module osa
    }; // end module threegpp
 }; // end module org

#endif

// END file CC.idl
```

3GPP Meeting CN5 #7
Sophia Antipolis, November 7-8

Document N5-000264
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**	**040**	Current Version:	3.1.0

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

For submission to:	CN#10	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: Ericsson **Date:** 27 October 2000

Subject: The name of P_CALL_REPORT_REFUSED_BUSY should be changed to P_CALL_REPORT_BUSY. Also, the member name should change from RefusedBusy to Busy.

Work item: OSA

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

F	Correction		**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	X		Release 99	X
				Release 00	

Reason for change: The naming should not imply that the end user intentionally refused the call by being busy. Also, there was an inconsistency with the naming in the IDL.

Clauses affected: 8.3.3.26, 9.3.1

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.

8.3.3.26 TpCallAdditionalReportInfo

Defines the Tagged Choice of Data Elements that specify additional call report information for certain types of reports.

	Tag Element Type	
	TpCallReportType	

Tag Element Value	Choice Element Type	Choice Element Name
P_CALL_REPORT_UNDEFINED	NULL	Undefined
P_CALL_REPORT_PROGRESS	NULL	Undefined
P_CALL_REPORT_ALERTING	NULL	Undefined
P_CALL_REPORT_ANSWER	NULL	Undefined
P_CALL_REPORT_~~REFUSED_~~BUSY	TpCallReleaseCause	~~Refused~~Busy
P_CALL_REPORT_NO_ANSWER	NULL	Undefined
P_CALL_REPORT_DISCONNECT	TpCallReleaseCause	CallDisconnect
P_CALL_REPORT_REDIRECTED	TpAddress	ForwardAddress
P_CALL_REPORT_SERVICE_CODE	TpCallServiceCode	ServiceCode
P_CALL_REPORT_ROUTING_FAILURE	TpCallReleaseCause	RoutingFailure

9.3.1 Common Data Types for Call Control

```
// source file: CC.idl
// Generic Call Data description

#ifndef __OSA_CC_DEFINED
#define __OSA_CC_DEFINED

#include <OSA.idl>
#include <UI.idl>

module org
{
 module threegpp
 {
     module osa
     {
       module cc
       {
..

         /* Defines a specific call event report type. */
         enum TpCallReportType
         {
            P_CALL_REPORT_UNDEFINED,       /* Undefined */
            P_CALL_REPORT_PROGRESS,        /* Call routing progress event */
            P_CALL_REPORT_ALERTING,        /* Call alerting at  address */
            P_CALL_REPORT_ANSWER,          /* Call answered at address */
            P_CALL_REPORT_BUSY,            /* Called address refused call due to busy */
            P_CALL_REPORT_NO_ANSWER,       /* No answer at called address */
            P_CALL_REPORT_DISCONNECT,      /* Call disconnect requested by address */
            P_CALL_REPORT_REDIRECTED,
            P_CALL_REPORT_SERVICE_CODE,
            P_CALL_REPORT_ROUTING_FAILURE

         };

         /* Defines the Tagged Choice of Data Elements that specify additional call report
       information. */
         union TpCallAdditionalReportInfo switch(TpCallReportType)
         {
            case P_CALL_REPORT_BUSY: TpCallReleaseCause RefuseBusy;
            case P_CALL_REPORT_DISCONNECT: TpCallReleaseCause CallDisconnect;
            case P_CALL_REPORT_REDIRECTED: TpAddress ForwardAddress;
```

```
            case P_CALL_REPORT_SERVICE_CODE: TpCallReleaseCause ServiceCode;
            case P_CALL_REPORT_ROUTING_FAILURE: TpCallReleaseCause RoutingFailure;

            default:  short Dummy;   // allows initialisation of the union in the default case
         };

         struct TpCallReport
         {
            TpCallMonitorMode MonitorMode;
            TpDateAndTime CallEventTime;
            TpCallReportType CallReportType;
            TpCallAdditionalReportInfo AdditionalReportInfo;
         };

..

         }; // end module cc
       }; // end module osa
     }; // end module threegpp
 }; // end module org

#endif

// END file CC.idl
```

3GPP TSG-CN WG5 Meeting #7
Sophia Antipolis, France, 7th – 8th November, 2000

***Document* N5-000292**
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	043	Current Version:	3.1.0

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

For submission to:	TSG-CN#10	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:	Nokia	**Date:**	7th November 2000

Subject:	Removal of the parameter serviceProperties in the method selectService

Work item:	OSA

Category: *(only one category shall be marked with an X)*				**Release:**	
	F	Correction	X	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 serviceProperties need not be supplied any more at this phase as the specific serviceID has been determined already.

Clauses affected:	6.2.3.4, 9

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.

6.2.3.4 IpAccess

<<Interface>> IpAccess
obtainInterface(interfaceName: in TpInterfaceName, fwInterface: out IpOsaRefRef): TpResult obtainInterfaceWithCallback(interfaceName: in TpInterfaceName, appInterface: in IpOsaRef, fwInterface: out IpOsaRefRef): TpResult accessCheck(serviceToken: in TpServiceToken,securityContext: in TpString, securityDomain: in TpString, group : in TpString, serviceAccessTypes: in TpString, serviceAccessControl: out TpServiceAccessControlRef): TpResult selectService(serviceID: in TpServiceID~~, serviceProperties: in TpServicePropertyList~~, serviceToken: out TpServiceTokenRef): TpResult signServiceAgreement(serviceToken: in TpServiceToken, agreementText: in TpString, signingAlgorithm: in TpSigningAlgorithm, signatureAndServiceMgr: out TpSignatureAndServiceMgrRef): TpResult terminateServiceAgreement(serviceToken: in TpServiceToken, terminationText: in TpString, digitalSignature: in TpString): TpResult endAccess(endAccessProperties: in TpEndAccessProperties) : TpResult

```
/* This method is invoked by the client application to identify the SCF that it wishes
to use. */
void selectService (
in TpServiceID serviceID,                       // Identifies the SCF.
in TpServicePropertyList serviceProperties,     // List the properties that the SCF
// should support.
out TpServiceToken serviceToken                 // A free format text token returned by
// the framework, which can be signed as
// part of a service agreement.
) raises (TpGeneralException);
```

3GPP Meeting CN5 #7
Sophia Antipolis, November 7-8

Document N5-000297
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	044	Current Version:	3.1.0

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

For submission to:	CN#10	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: Ericsson **Date:** 27 October 2000

Subject: Inclusion of missing state transitions in case call related information could not be retrieved.

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: In case the call is ended (either due explicit release by the call or due to specific network events) the SCF will try to collect the call information requested by the application. When this is impossible (due to e.g. dialog failure towards the SSF) there should be an explicit transition. This should be added in the STD.

Clauses affected: 7.2.2.

Other specs affected:

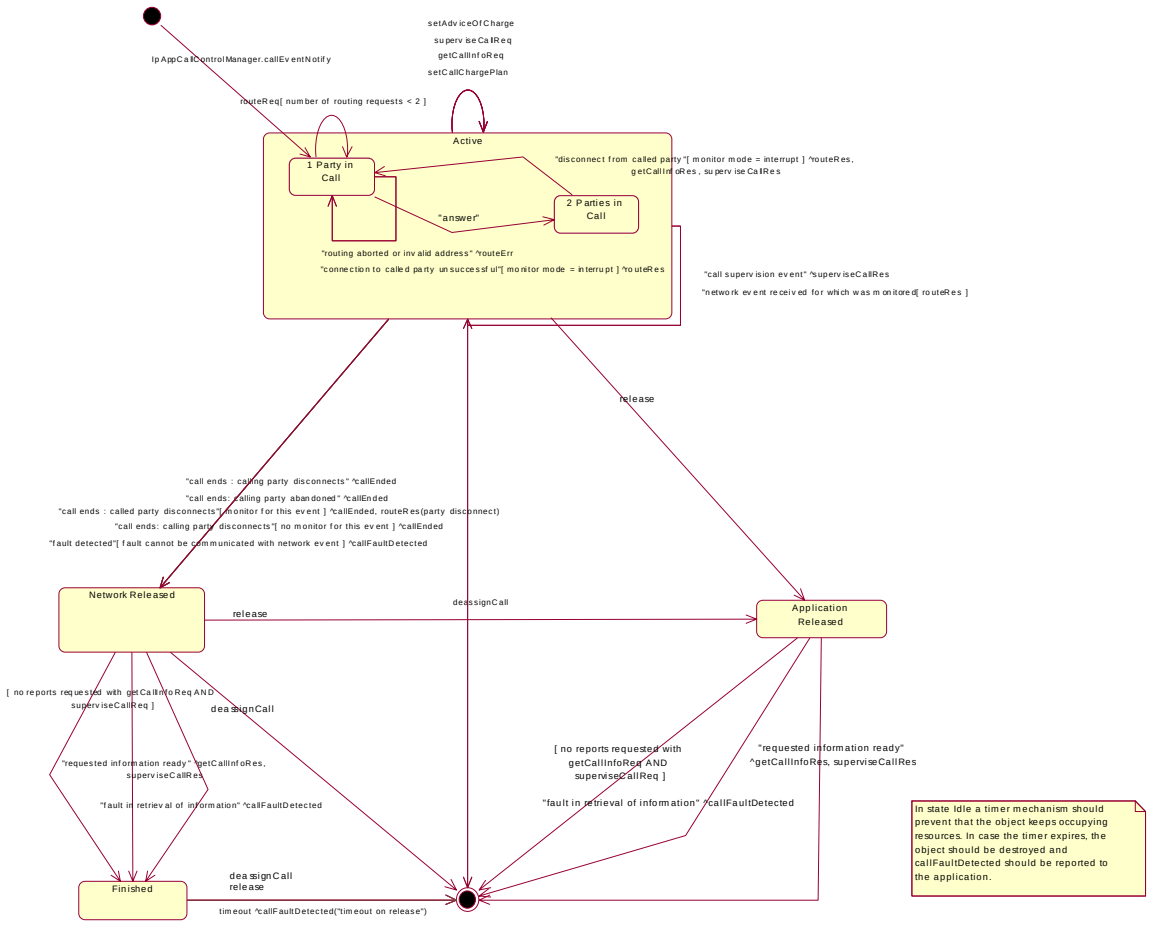
Other 3G core specifications		→ List of CRs:
Other GSM core specifications		→ List of CRs:
MS test specifications		→ List of CRs:
BSS test specifications		→ List of CRs:
O&M specifications		→ List of CRs:

Other comments:

help.doc

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

7.2.2 Call


IpAppCallControlManager.callEventNotify
setAdviceOfCharge
superviseCallReq
getCallInfoReq
setCallChargePlan
routeReq[number of routing requests < 2]
Active
1 Party in Call
2 Parties in Call
"answer"
"disconnect from called party"[monitor mode = interrupt] ^routeRes, getCallInfoRes, superviseCallRes
"routing aborted or invalid address" ^routeErr
"connection to called party unsuccessful"[monitor mode = interrupt] ^routeRes
"call supervision event" ^superviseCallRes
"network event received for which was monitored[routeRes]
release
"call ends : calling party disconnects" ^callEnded
"call ends: calling party abandoned" ^callEnded
"call ends : called party disconnects"[monitor for this event] ^callEnded, routeRes(party disconnect)
"call ends: calling party disconnects"[no monitor for this event] ^callEnded
"fault detected"[fault cannot be communicated with network event] ^callFaultDetected
Network Released
release
deassignCall
Application Released
[no reports requested with getCallInfoReq AND superviseCallReq]
deassignCall
"requested information ready" ^getCallInfoRes, superviseCallRes
"fault in retrieval of information" ^callFaultDetected
[no reports requested with getCallInfoReq AND superviseCallReq]
"requested information ready" ^getCallInfoRes, superviseCallRes
"fault in retrieval of information" ^callFaultDetected
Finished
deassignCall
release
timeout ^callFaultDetected("timeout on release")
In state Idle a timer mechanism should prevent that the object keeps occupying resources. In case the timer expires, the object should be destroyed and callFaultDetected should be reported to the application.

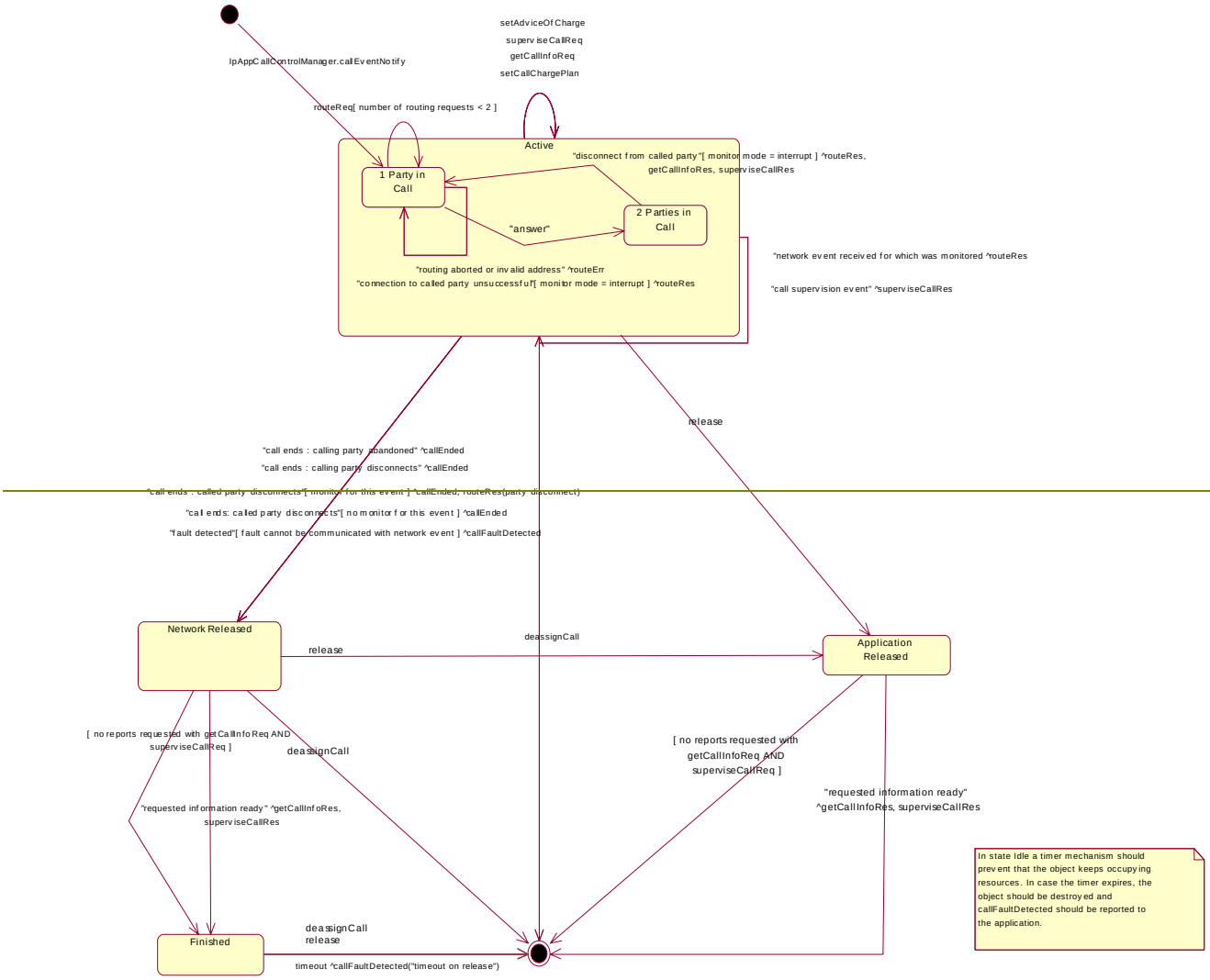


Figure 7-12: State Transition Diagram for Call

7.2.2.1 Active state

In this state a call between two parties is being setup or present. Refer to the substates for more details

The application can request the gateway for a certain type of charging of the call by calling setCallChargePlan(). The application can request for charging related information by calling getCallInfoReq(). Furthermore the application can request supervision of the call by calling superviseCallReq(). It is also allowed to send Advice Of Charge information by calling setAdviceOfCharge().

7.2.2.1.1 1 Party in Call state

When the Call is in this state a calling party is present. The application can now request that a connection to a called party be established by calling the method routeReq(). When the calling party abandons the call before the application has invoked the routeReq() operation, the gateway informs the application by invoking callFaultDetected() and also the operation callEnded() will be invoked. When the calling party abandons the call after the application has invoked routeReq() but before the call has actually been established, the gateway informs the application by invoking callEnded().

When the calling party answers the call, a transition will be made to the 2 Parties in Call state. In case the call can not be established because the application supplied an invalid address or the connection to the called party was unsuccessful while the application was monitoring for the latter in interrupt mode, the Call object will stay in this state

In this state user interaction is possible unless there is an outstanding routing request.

7.2.2.1.2 2 Parties in Call state

A connection between two parties has been established.

In case the calling party disconnects, the gateway informs the application by invoking callEnded().

When the called party disconnects different situations apply:

1. the application is monitoring for this event in interrupt mode: a transition is made to the 1 Party in Call state, the application is informed with routeRes with indication that the called party has disconnected and all requested reports are sent to the application. The application now again has control of the call.

2. the application is monitoring for this event but not in interrupt mode. In this case a transition is made to the Network Released state and the gateway informs the application by invoking the operation routeRes() and callEnded().

3. the application is not monitoring for this event. In this case the application is informed by the gateway invoking the callEnded() operation and a transition is made to the Network Released state.

7.2.2.3 Network released state

In this state the call has ended and the Gateway collects the possible call information requested with getCallInfoReq() and / or superviseCallReq(). The information will be returned to the application by invoking the methods getCallInfoRes() and / or superviseCallRes() on the application. Also when a call was unsuccessful these methods are used.In case the application has not requested additional call related information immediately a transition is made to state Idle.

7.2.2.4 Finished state

In this state the call has ended and no call related information is to be send to the application. The application can only release the Call object. Calling the deassingCall() method has the same effect. Note that the application has to release the object itself as good OO practice requires that when an object was created on behalf of a certain entity, this entity is also responsible for destroying it when the object is no longer needed.

7.2.2.5 Application released state.

In this state the application has requested to release the Call object and the Gateway collects the possible call information requested with getCallInfoReq(). In case the application has not requested additional call related information immediately the Call object is destroyed.

3GPP Meeting CN5 #6
Sophia Antipolis, France, 7-8 November 2000

Document N5-000298
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	033 R1	Current Version:	3.1.0

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

For submission to:	CN#10	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: Lucent Technologies **Date:** 6 November 2000

Subject: Missing syntax and semantics description for a number of security parameters, currently only defined as strings.

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: There is no description of the syntax and semantics of some of the security parameters in section 8 “Data Definitions” of 3G TS 29.198. These parameters are securityContext, securityDomain, group, and serviceAccessType. The descriptive text is the same as in the IDL package org.threegpp.osa.fw.trust_and_security.

Clauses affected: 6.2.3.4, 8.1.4.x, 9.2.3

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.

6.2.3.4 IpAccess

<<Interface>> IpAccess
obtainInterface(interfaceName: in TpInterfaceName, fwInterface: out IpOsaRefRef): TpResult obtainInterfaceWithCallback(interfaceName: in TpInterfaceName, appInterface: in IpOsaRef, fwInterface: out IpOsaRefRef): TpResult accessCheck(serviceToken: in TpServiceToken,securityContext: in ~~TpString~~TpSecurityContext, securityDomain: in ~~TpString~~TpSecurityDomain, group : in ~~TpString~~TpSecurityGroup, serviceAccessTypes: in ~~TpString~~TpServiceAccessType, serviceAccessControl: out TpServiceAccessControlRef): TpResult selectService(serviceID: in TpServiceID, serviceProperties: in TpServicePropertyList, serviceToken: out TpServiceTokenRef): TpResult signServiceAgreement(serviceToken: in TpServiceToken, agreementText: in TpString, signingAlgorithm: in TpSigningAlgorithm, signatureAndServiceMgr: out TpSignatureAndServiceMgrRef): TpResult terminateServiceAgreement(serviceToken: in TpServiceToken, terminationText: in TpString, digitalSignature: in TpString): TpResult endAccess(endAccessProperties: in TpEndAccessProperties) : TpResult

8.1.4.W TpSecurityContext

This data type is identical to a TpString and contains a group of security relevant attributes.

8.1.4.X TpSecurityDomain

This data type is identical to a TpString and contains the security domain in which the client application is operating.

8.1.4.Y TpSecurityGroup

This data type is identical to a TpString and contains a definition of the access rights associated with all clients that belong to that group.

8.1.4.Z TpServiceAccessType

This data type is identical to a TpString and contains a definition of the specific security model in use.

9.2.3 Trust and Security Management IDL

```
#include <fw.idl>

module org{
module threegpp{
module osa{
module fw{
module trust_and_security{

/*******************************************************************************/
```

```
//                                Data definitions                                //
/********************************************************************************/

    typedef TpString        TpAccessType;       // The type of access interface requested by the
client
// application. For OSA release 99 the following values
// have been defined: NULL (indicates the default access
// type) and P_ACCESS.

    typedef TpString        TpAuthType;         // The type of authentication mechanism requested by
the
// client. For OSA release 99 the following values have
// been defined: NULL (indicates OSA authentication),
// P_AUTHENTICATION (indicates use of the OSA
// authentication interfaces.

    typedef TpString    TpAuthCapability;       // The authentication capabilities that could be
supported
// by the OSA. For OSA release 99 the following values
// have been defined: NULL (indicates no client
// capabilities, P_DES_56, P_DES_128, P_RSA_512 and P_RSA_1024).

    typedef TpString        TpAuthCapabilityList;   // A string of multiple TpAuthCapability
// concatenated using a commas.
    struct TpAuthDomain
            {
                TpDomainID DomainID;
       IpOSA AuthInterface;
            };

typedef TpPropertyList TpEndAccessProperties;

    typedef TpString        TpInterfaceName;    // Identifies the names of the framework SCFs that
are be
// supported by the OSA API. For release 99 these are NULL,
// P_DISCOVERY, P_OAM
// P_LOAD_MANAGER,
// P_FAULT_MANAGER,
// P_HEARTBEAT_MANAGEMENT,
// P_REGISTRATION

typedef TpString TpSecurityContext;     // contains a group of security relevant attributes


typedef TpString TpSecurityDomain;      // contains the security domain in which the client
                                        // application is operating

typedef TpString TpSecurityGroup;       // contains a definition of the access rights associated
                                        // with all clients that belong to that group

typedef TpString TpServiceAccessType;   // contains a definition of the specific security model in
                                        // use

    struct TpServiceAccessControl {
        TpString        Policy;         // Access control policy information controlling access to
the
// service feature.
        TpString        TrustLevel;     // The level of trust that the network operator has assigned
to the
// client application.
    };

    typedef TpString        TpServiceToken; // Uniquely identifies a SCF.

    struct TpSignatureAndServiceMgrRef {
        TpString        DigitalSignature;       // The digital signature of the Framework for the
service
// agreement.
        IpOsa       ServiceMgrInterface;
    };

    typedef TpString        TpSigningAlgorithm;     // Identifies the signing algorithm that must be
// used. For OSA release 99 the follwing values have
// been defined: NULL (indicates no signing algorithm
// is required), P_MD5_RSA_512 and P_MD5_RSA_1024.

    typedef TpString        TpFwID;
```

```
    struct TpFwAuth {
        TpFwID  FwID;
        IpOsa   FwAuthInterface;
    };

/*******************************************************************************/
//                                  Interface definitions                      //
/*******************************************************************************/

/* The Initial Framework interface is used by the client application to initiate the mutual
authentication with the Framework and, when this is finished successfully, to request access
to it. */
interface IpInitial : IpOsa {

/* This method is invoked by the client application to start the process of mutual
authentication with the framework, and request the use of a specific authentication method.
*/
void initiateAuthentication (
in TpAuthDomain appDomain,          // Identifies the client to the framework.
in TpAuthType authType,                 // Allows the client application to request a
// specific type of authentication mechanism.

out TpAuthDomain fwDomain          // Provides a framework identifier, and a reference
// to framework authentication interface.
) raises (TpGeneralException);

/* This method is invoked by the client application, once mutual authentication is
achieved, to request access to the framework and specify the type of access desired. */
void requestAccess (
in TpAccessType accessType,         // Identifies the type of access interface requested by
// the client application.
in IpOsa appAccessInterface,        // Provides a reference to the access interface of the
// client application.
out IpOsa fwAccessInterface          // Provides a reference to call the access interface of
 // the framework.
) raises (TpGeneralException);

};


/* The Access Framework interface is used by the client application to perform the mechanisms
necessary for it to obtain access to SCFs. */
interface IpAccess : IpOsa {

/* This method is invoked by the client application to obtain interface references to other
framework interfaces. */
void obtainInterface (
in TpInterfaceName interfaceName,   // The name of the framework interface to which a
// reference to the interface is requested.
out IpOsa fwInterface                   // The requested interface reference.
) raises (TpGeneralException);

/* This method is invoked by the client application to obtain interface references to other
framework interfaces, when it is required to supply a callback interface to the framework. */
void obtainInterfaceWithCallback (
in TpInterfaceName interfaceName,           // The name of the framework interface to which
// a reference to the interface is requested.
in IpOsa appInterface,                      // This is the reference to the client application
// interface which is used for callbacks.
out IpOsa fwInterface                           // The requested interface reference.
) raises (TpGeneralException);

/* This method may be invoked by the client application to check whether it has been
granted permission to access the specified SCF and, if granted, the level of trust that
will be applied. */
void accessCheck (
in TpServiceToken serviceToken,
in ~~TpString~~ TpSecurityContext securityContext,                          // A group of security
relevant
// attributes.
in ~~TpString~~ TpSecurityDomain securityDomain,                            // The security domain in
which
// the client application is
// operating.
in ~~TpString~~ TpSecurityGroup group,                                      // Used to define the access
// rights associated with all
// clients that belong to that
// group.
```

```
in ~~TpString~~ TpServiceAccessType serviceAccessTypes,                    // Defined by the
specific
// security model in use.
out TpServiceAccessControl serviceAccessControl        // The access control policy
// information controlling
// access to the service
// capability feature, and the
// trustLevel that the network
// operator has assigned to the client
// application.
) raises (TpGeneralException);

/* This method is invoked by the client application to identify the SCF that it wishes
to use. */
void selectService (
in TpServiceID serviceID,                      // Identifies the SCF.
in TpServicePropertyList serviceProperties,    // List the properties that the SCF
// should support.
out TpServiceToken serviceToken                // A free format text token returned by
// the framework, which can be signed as
// part of a service agreement.
) raises (TpGeneralException);

/* This method is invoked by the client application to request that the framework sign an
agreement on the SCF, which allows the client application to use the SCF. */
void signServiceAgreement (
in TpServiceToken serviceToken,                          // Used to identify the SCF
// instance requested by the
// client application.
in TpString agreementText,                               // The agreement text to be
// signed by the framework.
in TpSigningAlgorithm signingAlgorithm,                  // The algorithm used to compute
// the digital signature.
out TpSignatureAndServiceMgrRef signatureAndServiceMgr   // A reference to a structure
// that contains the digital
// signature of the framework
// for the service agreement,
// and a reference to the
// SCF manager interface of
// the SCF.
) raises (TpGeneralException);

/* This method is invoked by the client application to terminate an agreement for the
specified SCF. */
void terminateServiceAgreement (
in TpServiceToken serviceToken,     // Identifies the service agreement to be terminated.
in TpString terminationText,        // Describes the reason for the termination of the
// service agreement.
in TpString digitalSignature        // Used by the framework to check that the
// terminationText has been signed by the client.
) raises (TpGeneralException);

/* This method is invoked by the client application to end the access session
with the Framework. */
void endAccess () raises (TpGeneralException);

};



/* The Access client application interface is used by the Framework to perform the steps that
are necessary in order to allow it to SCF access. */
interface IpAppAccess : IpOsa {

/* This method is invoked by the Framework to request that client application sign an
agreement on a specified SCF. */
void signServiceAgreement (
in TpServiceToken serviceToken,             // Identifies the SCF instance to which
 // this service agreement corresponds.
in TpString agreementText,                  // Agreement text that has to be signed by the
 // client application.
in TpSigningAlgorithm signingAlgorithm,     // Algorithm used to compute the digital
 // signature.
out TpString digitalSignature               // Signed version of a hash of the service
// token and agreement text given by the
// framework.
) raises (TpGeneralException);
```

```
/* This method is invoked by the Framework to terminate an agreement for a specified
SCF. */
void terminateServiceAgreement (
in TpServiceToken serviceToken,         // Identifies the SCF agreement to be terminated.
in TpString terminationText,       // Describes the reason for the termination.
in TpString digitalSignature        // Used by the Framework to confirm its identity to the
// client.
) raises (TpGeneralException);

/* This method is invoked by the Framework to end the client application's access session
with the framework. */
void terminateAccess (
in TpString terminationText,                // Describes the reason for the termination of
 // the access session.
in TpSigningAlgorithm signingAlgorithm,     // The algorithm used to compute the digital
// signature.
in TpString digitalSignature                // Used by the Framework to confirm its
// identity to the client.
) raises (TpGeneralException);

};



/* The Authentication Framework interface is used by client application to perform its part of
the mutual authentication process with the Framework necessary to be allowed to use any of the
other interfaces supported by the Framework. */
interface IpAuthentication : IpOsa {

/* This method is invoked by the client application to start the authentication process,
informed the Framework of the authentication mechanisms it supports, and be informed by its
of its preferred choice. */
void selectAuthMethod (
in TpAuthCapabilityList auths,      // Informs the Framework of the authentication
// mechanisms supported by the client
// application.
out TpAuthCapability prescribedMethod           // Indicates the mechanism preferred by the
// framework.
) raises (TpGeneralException);

/* This method is invoked by the client application to authenticate the framework using the
mechanism indicated in the parameter prescribedMethod. */
void authenticate (
in TpAuthCapability prescribedMethod,       // Specifies the method accepted by that the
// framework for authentication.
in TpString challenge,                  // The challenge presented by the client
// application to be responded to by the
// framework.
out TpString response                       // The response of the framework to the
// challenge of the client application.
) raises (TpGeneralException);

/* This method is invoked by the client application to to abort the authentication
process.*/
void abortAuthentication() raises (TpGeneralException);

};



/* The Authentication client application interface is used by the Framework to authenticate

the client application. */

interface IpAppAuthentication : IpOsa {

/* This method is invoked by the Framework to authenticate the client application using the
mechanism indicated in prescribedMethod. */
void authenticate (
in TpAuthCapability prescribedMethod,       // The agreed authentication method.
in TpString challenge,                          // The challenge presented by the Framework.
out TpString response
) raises (TpGeneralException);

/* This method is invoked by the Framework to abort the authentication process. */
void abortAuthentication() raises (TpGeneralException);

};
```

```
};};};};};
```

3GPP Meeting CN5 #6
Sophia Antipolis, France, 7-8 November 2000

Document N5-000299
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	036 R1	Current Version:	3.1.0

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

For submission to: CN#10 *list expected approval meeting # here ↑*	for approval	X	strategic		*(for SMG use only)*
	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: Lucent Technologies **Date:** 6 November 2000

Subject: Unclear description for TpAuthType

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 distinction between using OSA Authentication and using the OSA Authentication Interfaces is not clear from the text in the second column of the table for TpAuthType. In fact, there is no distinction.

Clauses affected: 8.2.2.2, 9.2.3

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.

8.2.2.2 TpAuthType

This data type is identical to a TpString. It identifies the type of authentication mechanism requested by the client. It provides Network operators and client's with the opportunity to use an alternative to the OSA Authentication interface, e.g. CORBA Security. OSA Authentication is the default authentication method. Other Network operator specific capabilities may also be used, but should be preceded by the string "SP_". The following value~~s are~~is defined for OSA release 99:

String Value	Description
~~*NULL*~~	~~An empty (NULL) string indicates the default authentication method: OSA Authentication.~~
P_AUTHENTICATION	~~Authenticate using the OSA Authentication Interfaces: IpAuthentication and IpAppAuthentication~~ Indicates the default authentication method, i.e. the IpAuthentication and IpAppAuthentication interfaces.

9.2.3 Trust and Security Management IDL

```
#include <fw.idl>

module org{

module threegpp{

module osa{

module fw{

module trust_and_security{

      /*******************************************************************************/
      //                                Data definitions                             //
      /*******************************************************************************/

   typedef TpString     TpAccessType;     // The type of access interface requested by the client
                                          // application. For OSA release 99 the following values
                                          // have been defined: NULL (indicates the default access
                                          // type) and P_ACCESS.

   typedef TpString     TpAuthType;       // The type of authentication mechanism requested by the
                                          // client. For OSA release 99 the following values ~~have~~ have been defined: ~~NULL (indicates OSA authentication),~~
                                          // P_AUTHENTICATION (indicates use of the OSA
                                          // authentication interfaces).

   typedef TpString  TpAuthCapability;    // The authentication capabilities that could be supported
                                          // by the OSA. For OSA release 99 the following values
                                          // have been defined: NULL (indicates no client
```

```
   // capabilities, P_DES_56, P_DES_128, P_RSA_512 and
P_RSA_1024).
```