

3GPP TSG CN Plenary Meeting #16
5th - 7th June 2002. Marco Island, USA.

NP-020186

Source: CN5 (OSA)
Title: Rel-5 CRs 29.198-03 OSA API Part 3: Framework
Agenda item: 8.2
Document for: APPROVAL

Doc-1 st -Level	Spec	CR	R v	Pha	Subject	Cat	Ver Curr	Ver New	Doc-2 nd -Level	Work item
NP-020186	29.198-03	032	-	Rel-5	Remove redundant data type definition TpServiceSpecString	F	4.5.0	5.0.0	N5-020364	OSA2
NP-020186	29.198-03	034	-	Rel-5	Delete description of P_APPLICATION_NOT_ACTIVATED conflicting with 29.198-02	F	4.5.0	5.0.0	N5-020374	OSA2
NP-020186	29.198-03	036	-	Rel-5	Clarify semantics of service properties of type BOOLEAN_SET	F	4.5.0	5.0.0	N5-020401	OSA2
NP-020186	29.198-03	037	-	Rel-5	Addition of version management support to the Framework (29.198-03) in run-time	B	4.5.0	5.0.0	N5-020467	OSA2
NP-020186	29.198-03	038	-	Rel-5	Enhancements on subscription management error information	B	4.5.0	5.0.0	N5-020470	OSA2
NP-020186	29.198-03	039	-	Rel-5	Delete conflicting description of P_APPLICATION_NOT_ACTIVATED	F	4.5.0	5.0.0	N5-020472	OSA2
NP-020186	29.198-03	040	-	Rel-5	Note added for P_SERVICE_INSTANCE Choice Element Name	D	4.5.0	5.0.0	N5-020485	OSA2
NP-020186	29.198-03	041	-	Rel-5	Correcting the method descriptions for abortAuthentication and for initiateAuthentication	F	4.5.0	5.0.0	N5-020495	OSA2
NP-020186	29.198-03	042	-	Rel-5	Correcting the description of heartbeat failure	F	4.5.0	5.0.0	N5-020496	OSA2
NP-020186	29.198-03	043	-	Rel-5	Correcting erroneous FW<->Service instance sequence diagrams	F	4.5.0	5.0.0	N5-020497	OSA2
NP-020186	29.198-03	044	-	Rel-5	Correcting the scope of TpFwID, which currently is giving it false limitations	F	4.5.0	5.0.0	N5-020498	OSA2

CHANGE REQUEST

⌘ **29.198-03 CR 032** ⌘ rev **-** ⌘ Current version: **4.5.0** ⌘

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

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

Title:	⌘	Remove redundant data type definition TpServiceSpecString
Source:	⌘	CN5
Work item code:	⌘	OSA2
		Date: ⌘ 17/05/2002
Category:	⌘	F
		Use <u>one</u> of the following categories: F (correction) A (corresponds to a correction in an earlier release) B (addition of feature), C (functional modification of feature) D (editorial modification) Detailed explanations of the above categories can be found in 3GPP TR 21.900.
		Release: ⌘ REL-5 Use <u>one</u> of the following releases: 2 (GSM Phase 2) R96 (Release 1996) R97 (Release 1997) R98 (Release 1998) R99 (Release 1999) REL-4 (Release 4) REL-5 (Release 5)

Reason for change:	⌘	In contribution N5-010382, presented in San Diego last May, Lucent proposed to change the definition of servcID from a concatenation of Unique Service Number, Service Name and Service Specialisation to just a simple unique identifying number. This proposal was accepted and TpUniqueServiceNumber was removed from 29.198-03. However, TpServiceSpecString remains (as sub-section 10.1.18 of the 29.198-03 specification). The problem is that TpServiceSpecString is not referenced anywhere else within the specification, and is therefore unused and redundant. This CR removes this type definition. As it is completely unused, this should not cause any issues with backwards compatibility.
Summary of change:	⌘	Remove redundant data type definition TpServiceSpecString
Consequences if not approved:	⌘	Redundant data type.

Clauses affected:	⌘	10.1.18
Other specs affected:	⌘	<input type="checkbox"/> Other core specifications ⌘ <input type="checkbox"/> Test specifications <input type="checkbox"/> O&M Specifications
Other comments:	⌘	

How to create CRs using this form:

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

- 1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.
- 2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be

downloaded from the 3GPP server under <ftp://ftp.3gpp.org/specs/> For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

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

**** FIRST MODIFIED SECTION ****

10 Data Definitions

This clause provides the Framework specific data definitions necessary to support the OSA interface specification.

The general format of a data definition specification is the following:

- Data type, that shows the name of the data type;
- Description, that describes the data type;
- Tabular specification, that specifies the data types and values of the data type;
- Example, if relevant, shown to illustrate the data type.

All data types referenced but not defined in this clause are common data definitions which may be found in 3GPP TS 29.198-2.

10.1 Common Framework Data Definitions

10.1.1 TpClientAppID

This is an identifier for the client application. It is used to identify the client to the Framework. This data type is identical to TpString and is defined as a string of characters that uniquely identifies the application. The content of this string shall be unique for each OSA API implementation (or unique for a network operator's domain). This unique identifier shall be negotiated with the OSA operator and the application shall use it to identify itself.

10.1.2 TpClientAppIDList

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

10.1.3 TpDomainID

Defines the Tagged Choice of Data Elements that specify either the Framework or the type of entity attempting to access the Framework.

	Tag Element Type	
	TpDomainIDType	

Tag Element Value	Choice Element Type	Choice Element Name
P_FW	TpFwID	FwID
P_CLIENT_APPLICATION	TpClientAppID	ClientAppID
P_ENT_OP	TpEntOpID	EntOpID
P_SERVICE_INSTANCE	TpServiceInstanceID	ServiceID
P_SERVICE_SUPPLIER	TpServiceSupplierID	ServiceSupplierID

10.1.4 TpDomainIDType

Defines either the Framework or the type of entity attempting to access the Framework.

Name	Value	Description
P_FW	0	The Framework
P_CLIENT_APPLICATION	1	A client application
P_ENT_OP	2	An enterprise operator
P_SERVICE_INSTANCE	3	A service instance
P_SERVICE_SUPPLIER	4	A service supplier

10.1.5 TpEntOpID

This data type is identical to TpString and is defined as a string of characters that identifies an enterprise operator. In conjunction with the application it uniquely identifies the enterprise operator which uses a particular OSA Service Capability Feature (SCF).

10.1.6 TpPropertyName

This data type is identical to TpString. It is the name of a generic “property”.

10.1.7 TpPropertyValue

This data type is identical to TpString. It is the value (or the list of values) associated with a generic “property”.

10.1.8 TpProperty

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

Sequence Element Name	Sequence Element Type
PropertyName	TpPropertyName
PropertyValue	TpPropertyValue

10.1.9 TpPropertyList

This data type defines a Numbered List of Data Elements of type TpProperty.

10.1.10 TpEntOpIDList

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

10.1.11 TpFwID

This data type is identical to TpString and identifies the Framework to a client application (or Service Capability Feature)

10.1.12 TpService

This data type is a Sequence of Data Elements which describes a registered SCFs. It is a structured type which consists of:

Sequence Element Name	Sequence Element Type	Documentation
ServiceID	TpServiceID	
ServiceDescription	TpServiceDescription	This field contains the description of the service

10.1.13 TpServiceList

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

10.1.14 TpServiceDescription

This data type is a Sequence of Data Elements which describes a registered SCF. It is a structured data type which consists of:

Sequence Element Name	Sequence Element Type	Documentation
ServiceTypeName	TpServiceTypeName	
ServicePropertyList	TpServicePropertyList	

10.1.15 TpServiceID

This data type is identical to a TpString, and is defined as a string of characters that uniquely identifies a registered SCF interface. The string is automatically generated by the Framework.

10.1.16 TpServiceIDList

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

10.1.17 TpServiceInstanceID

This data type is identical to a TpString, and is defined as a string of characters that uniquely identifies an instance of a registered SCF interface. The string is automatically generated by the Framework

~~10.1.18 TpServiceSpecString~~

~~This data type is identical to a TpString, and is defined as a string of characters that uniquely identifies the name of an SCF specialization interface. Other network operator specific capabilities may also be used, but should be preceded by the string "SP_". The following values are defined.~~

Character String Value	Description
<i>NULL</i>	An empty (NULL) string indicates no SCF specialization
P_CALL	The Call specialization of the of the User Interaction SCF

10.1.189 TpServiceTypeProperty

This data type is a Sequence of Data Elements which describes a service property associated with a service type. It defines the name and mode of the service property, and also the service property type: e.g. Boolean, integer. It is similar to, but distinct from, TpServiceProperty. The latter is associated with an actual service: it defines the service property's name and mode, but also defines the list of values assigned to it.

Sequence Element Name	Sequence Element Type	Documentation
ServicePropertyName	TpServicePropertyName	
ServiceTypePropertyMode	TpServiceTypePropertyMode	
ServicePropertyTypeName	TpServicePropertyTypeName	

***** END OF DOCUMENT *****

CHANGE REQUEST

⌘ **29.198-03 CR 034** ⌘ rev **-** ⌘ Current version: **4.5.0** ⌘

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

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

Title:	⌘ Delete description of P_APPLICATION_NOT_ACTIVATED conflicting with 29.198-02				
Source:	⌘ CN5				
Work item code:	⌘ OSA2	Date:	⌘ 17/05/2002		
Category:	⌘ F	Release:	⌘ REL-5		
	Use <u>one</u> of the following categories:		Use <u>one</u> of the following releases:		
	F (correction)		2	(GSM Phase 2)	
	A (corresponds to a correction in an earlier release)		R96	(Release 1996)	
	B (addition of feature),		R97	(Release 1997)	
	C (functional modification of feature)		R98	(Release 1998)	
	D (editorial modification)		R99	(Release 1999)	
	Detailed explanations of the above categories can be found in 3GPP TR 21.900.		REL-4	(Release 4)	
			REL-5	(Release 5)	

Reason for change:	⌘ The description of P_APPLICATION_NOT_ACTIVATED in 29.198-3 is in conflict with Part 2 (29.198-2). The IDL only defines P_APPLICATION_NOT_ACTIVATED in Part 2.				
Summary of change:	⌘ Delete the description of P_APPLICATION_NOT_ACTIVATED in Part 3				
Consequences if not approved:	⌘ Multiple descriptions of P_APPLICATION_NOT_ACTIVATED may lead to confusion.				

Clauses affected:	⌘ 11				
Other specs affected:	⌘ <input type="checkbox"/> Other core specifications	⌘			
	<input type="checkbox"/> Test specifications				
	<input type="checkbox"/> O&M Specifications				
Other comments:	⌘				

How to create CRs using this form:

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

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

11 Exception Classes

The following are the list of exception classes which are used in this interface of the API.

Name	Description
P_ACCESS_DENIED	The client is not currently authenticated with the framework
P_APPLICATION_NOT_ACTIVATED	An application is unauthorised to access information and request services with regards to users that have deactivated that particular application.
P_DUPLICATE_PROPERTY_NAME	A duplicate property name has been received
P_ILLEGAL_SERVICE_ID	Illegal Service ID
P_ILLEGAL_SERVICE_TYPE	Illegal Service Type
P_INVALID_ACCESS_TYPE	The framework does not support the type of access interface requested by the client.
P_INVALID_ACTIVITY_TEST_ID	ID does not correspond to a valid activity test request
P_INVALID_AGREEMENT_TEXT	Invalid agreement text
P_INVALID_ENCRYPTION_CAPABILITY	Invalid encryption capability
P_INVALID_AUTH_TYPE	Invalid type of authentication mechanism
P_INVALID_CLIENT_APP_ID	Invalid Client Application ID
P_INVALID_DOMAIN_ID	Invalid client ID
P_INVALID_ENT_OP_ID	Invalid Enterprise Operator ID
P_INVALID_PROPERTY	The framework does not recognise the property supplied by the client
P_INVALID_SAG_ID	Invalid Subscription Assignment Group ID
P_INVALID_SERVICE_CONTRACT_ID	Invalid Service Contract ID
P_INVALID_SERVICE_ID	Invalid service ID
P_INVALID_SERVICE_PROFILE_ID	Invalid service profile ID
P_INVALID_SERVICE_TOKEN	The service token has not been issued, or it has expired.
P_INVALID_SERVICE_TYPE	Invalid Service Type
P_INVALID_SIGNATURE	Invalid digital signature
P_INVALID_SIGNING_ALGORITHM	Invalid signing algorithm
P_MISSING_MANDATORY_PROPERTY	Mandatory Property Missing
P_NO_ACCEPTABLE_ENCRYPTION_CAPABILITY	An encryption mechanism, which is acceptable to the framework, is not supported by the client
P_PROPERTY_TYPE_MISMATCH	Property Type Mismatch
P_SERVICE_ACCESS_DENIED	The client application is not allowed to access this service.
P_SERVICE_NOT_ENABLED	The service ID does not correspond to a service that has been enabled
P_SERVICE_TYPE_UNAVAILABLE	The service type is not available according to the Framework.
P_UNKNOWN_SERVICE_ID	Unknown Service ID
P_UNKNOWN_SERVICE_TYPE	Unknown Service Type

CHANGE REQUEST

⌘ 29.198-03 CR 036 ⌘ rev - ⌘ Current version: 4.5.0 ⌘

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

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

Title:	⌘ Clarify semantics of service properties of type BOOLEAN_SET		
Source:	⌘ CN5		
Work item code:	⌘ OSA2	Date:	⌘ 17/05/2002
Category:	⌘ F Use <u>one</u> of the following categories: F (correction) A (corresponds to a correction in an earlier release) B (addition of feature), C (functional modification of feature) D (editorial modification) Detailed explanations of the above categories can be found in 3GPP TR 21.900.	Release:	⌘ REL-5 Use <u>one</u> of the following releases: 2 (GSM Phase 2) R96 (Release 1996) R97 (Release 1997) R98 (Release 1998) R99 (Release 1999) REL-4 (Release 4) REL-5 (Release 5)

Reason for change:	⌘ Unclear specification.
Summary of change:	⌘ Clarification added for the use of service properties of type BOOLEAN_SET.
Consequences if not approved:	⌘ Interoperability problems.

Clauses affected:	⌘ 7.3.1.1, 9.1
Other specs affected:	⌘ <input type="checkbox"/> Other core specifications ⌘ <input type="checkbox"/> <input type="checkbox"/> Test specifications <input type="checkbox"/> O&M Specifications
Other comments:	⌘

How to create CRs using this form:

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

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

Introduction

The Service Type defines which properties the supplier of an SCF shall provide when he registers a service. The application specifies which properties an SCF shall contain when discovering a service. The type of a Service Property can be one of:

- ◆ BOOLEAN_SET
- ◆ INTEGER_SET
- ◆ STRING_SET
- ◆ ADDRESSRANGE_SET
- ◆ INTEGER_INTERVAL
- ◆ STRING_INTERVAL
- ◆ INTEGER_INTEGER_MAP

29.198-03 describes the use of Service Properties.

Proposed Change

For Service Properties of type BOOLEAN_SET there are cases for which there are no clear semantics:

- ◆ An SCF can register with an empty set.
- ◆ An application can request a service and specify {TRUE, FALSE} for a BOOLEAN_SET type service property.

Ericsson proposes to clarify that both cases are illegal.

Furthermore a clarification is proposed for `discoverService()` to indicate that an application should only provide the properties that it is interested in.

The next section lists the changes to 29.198-03 v4.2.0 (with revision marks).

9.1 Service Property Types

The service type defines which properties the supplier of an SCF shall provide when he registers an SCF.

At Service Registration the properties of a type shall be interpreted as the set of values that can be supported by the service. If a service type has a certain property (e.g. "CAN_DO_SOMETHING"), a service registers with a property value of {"true", "false"}. This means that the SCS is able to support Service instances where this property is used or allowed and instances where this property is not used or allowed. This clarifies why sets of values shall be used for the property values instead of primitive types.

At establishment of the Service Level Agreement the property can then be set to the value of the specific agreement. The context of the Service Level Agreement thus restricts the set of property values of the SCS and will thus lead to a sub-set of the service property values. When the correct SCF is instantiated during the discovery and selection procedure (see Note), the Service Properties shall thus be interpreted as the requested property values.

NOTE: This is achieved through the `createServiceManager()` operation in the Service Instance Lifecycle Manager interface.

All property values are represented by an array of strings. The following table shows all supported property types.

Property type name	Description	Example value (array of strings)	Interpretation of example value
BOOLEAN_SET	set of Booleans	{"FALSE"}	The set of Booleans consisting of the Boolean "false".
INTEGER_SET	set of integers	{"1", "2", "5", "7"}	The set of integers consisting of the integers 1, 2, 5 and 7.
STRING_SET	set of strings	{"Sophia", "Rijen"}	The set of strings consisting of the string "Sophia" and the string "Rijen"
ADDRESSRANGE_SET	set of address ranges	{"123??*", "*.ericsson.se"}	The set of address ranges consisting of ranges 123??* and *.ericsson.se.
INTEGER_INTERVAL	interval of integers	{"5", "100"}	The integers that are between or equal to 5 and 100.
STRING_INTERVAL	interval of strings	{"Rijen", "Sophia"}	The strings that are between or equal to the strings "Rijen" and "Sophia", in lexicographical order.
INTEGER_INTEGER_MAP	map from integers to integers	{"1", "10", "2", "20", "3", "30"}	The map that maps 1 to 10, 2 to 20 and 3 to 30.

The bounds of the string interval and the integer interval types may hold the reserved value "UNBOUNDED". If the left bound of the interval holds the value "UNBOUNDED", the lower bound of the interval is the smallest value supported by the type. If the right bound of the interval holds the value "UNBOUNDED", the upper bound of the interval is the largest value supported by the type.

When an SCF is registered by the Service Supplier, Service Properties of type BOOLEAN_SET shall not contain an empty set. When a service is discovered by an application, this application shall specify either {TRUE} or {FALSE} as value for service properties of type BOOLEAN_SET.

7.3.1.1 Interface Class IpServiceDiscovery

Method

discoverService()

The discoverService operation is the means by which a client application is able to obtain the service IDs of the services that meet its requirements. The client application passes in a list of desired service properties to describe the service it is looking for, in the form of attribute/value pairs for the service properties. The client application also specifies the maximum number of matched responses it is willing to accept. The framework must not return more matches than the specified maximum, but it is up to the discretion of the Framework implementation to choose to return less than the specified maximum. The discoverService() operation returns a serviceID/Property pair list for those services that match the desired service property list that the client application provided. The service properties returned will form a complete view of what the client application will be able to do with the service, as per the service level agreement. If the framework supports service subscription, the service level agreement will be encapsulated in the subscription properties contained in the contract/profile for the client application, which will be a restriction of the registered properties.

Returns <serviceList> : This parameter gives a list of matching services. Each service is characterised by its service ID and a list of service property {name, mode and value list} tuples associated with the service.

Parameters

serviceName : in TpServiceTypeName

The "serviceName" parameter conveys the required service type. It is key to the central purpose of "service trading". It is the basis for type safe interactions between the service exporters (via registerService) and service importers (via

discoverService). By stating a service type, the importer implies the service type and a domain of discourse for talking about properties of service.

· If the string representation of the "type" does not obey the rules for service type identifiers, then the P_ILLEGAL_SERVICE_TYPE exception is raised.

· If the "type" is correct syntactically but is not recognised as a service type within the Framework, then the P_UNKNOWN_SERVICE_TYPE exception is raised.

The framework may return a service of a subtype of the "type" requested. A service sub-type can be described by the properties of its supertypes.

desiredPropertyList : in TpServicePropertyList

The "desiredPropertyList" parameter is a list of service property {name, mode and value list} tuples that the discovered set of services should satisfy. These properties deal with the non-functional and non-computational aspects of the desired service. The property values in the desired property list must be logically interpreted as "minimum", "maximum", etc. by the framework (due to the absence of a Boolean constraint expression for the specification of the service criterion). It is suggested that, at the time of service registration, each property value be specified as an appropriate range of values, so that desired property values can specify an "enclosing" range of values to help in the selection of desired services.

The desiredPropertyList only contains service properties that are relevant for the application. If an application is not interested in the value of a certain service property, this service property shall not be included in the desiredPropertyList.

P_INVALID_PROPERTY is raised when an application includes an unknown service property name or invalid service property value.

max : in TpInt32

The "max" parameter states the maximum number of services that are to be returned in the "serviceList" result.

Returns

TpServiceList

Raises

TpCommonExceptions, P_ACCESS_DENIED, P_ILLEGAL_SERVICE_TYPE, P_UNKNOWN_SERVICE_TYPE, P_INVALID_PROPERTY

8.3.1.1 Interface Class IpFwServiceRegistration

Method

registerService()

The registerService() operation is the means by which a service is registered in the Framework, for subsequent discovery by the enterprise applications. Registration can only succeed when the Service type of the service is known to the Framework (ServiceType is 'available'). A service-ID is returned to the service supplier when a service is registered in the Framework. When the service is not registered because the ServiceType is 'unavailable', a P_SERVICE_TYPE_UNAVAILABLE is raised. The service-ID is the handle with which the service supplier can identify the registered service when needed (e.g. for withdrawing it). The service-ID is only meaningful in the context of the Framework that generated it.

Returns <serviceID> : This is the unique handle that is returned as a result of the successful completion of this operation. The Service Supplier can identify the registered service when attempting to access it via other operations such as unregisterService(), etc. Enterprise client applications are also returned this service-ID when attempting to discover a service of this type.

Parameters

serviceName : in TpServiceTypeName

The "serviceName" parameter identifies the service type. If the string representation of the "type" does not obey the rules for identifiers, then an P_ILLEGAL_SERVICE_TYPE exception is raised. If the "type" is correct syntactically but the Framework is able to unambiguously determine that it is not a recognised service type, then a P_UNKNOWN_SERVICE_TYPE exception is raised.

servicePropertyList : in TpServicePropertyList

The "servicePropertyList" parameter is a list of property name and property value pairs. They describe the service being registered. This description typically covers behavioural, non-functional and non-computational aspects of the service. Service properties are marked "mandatory" or "readonly". These property mode attributes have the following semantics:

- a. mandatory - a service associated with this service type must provide an appropriate value for this property when registering.
- b. readonly - this modifier indicates that the property is optional, but that once given a value, subsequently it may not be modified.

Specifying both modifiers indicates that a value must be provided and that subsequently it may not be modified. Examples of such properties are those which form part of a service agreement and hence cannot be modified by service suppliers during the life time of service.

If the type or the semantics of the type of any of the property values is not the same as the declared type (declared in the service type), then a P_PROPERTY_TYPE_MISMATCH exception is raised. If an attempt is made to assign a dynamic property value to a readonly property, then the P_READONLY_DYNAMIC_PROPERTY exception is raised. If the "servicePropertyList" parameter omits any property declared in the service type with a mode of mandatory, then a P_MISSING_MANDATORY_PROPERTY exception is raised. If two or more properties with the same property name are included in this parameter, the P_DUPLICATE_PROPERTY_NAME exception is raised.

Returns

TpServiceID

Raises

**TpCommonExceptions, P_ILLEGAL_SERVICE_ID,
P_UNKNOWN_SERVICE_ID, P_PROPERTY_TYPE_MISMATCH, P_DUPLICATE_PROPERTY_NAME,
P_ILLEGAL_SERVICE_TYPE, P_UNKNOWN_SERVICE_TYPE,
P_MISSING_MANDATORY_PROPERTY, P_SERVICE_TYPE_UNAVAILABLE**

CHANGE REQUEST

⌘ **29.198-03 CR 037** ⌘ rev **-** ⌘ Current version: **4.5.0** ⌘

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

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

Title:	⌘	Addition of version management support to the Framework (29.198-03) in run-time		
Source:	⌘	CN5		
Work item code:	⌘	OSA2	Date:	⌘ 17/05/2002
Category:	⌘	B	Release:	⌘ REL-5
		Use <u>one</u> of the following categories: F (correction) A (corresponds to a correction in an earlier release) B (addition of feature), C (functional modification of feature) D (editorial modification) Detailed explanations of the above categories can be found in 3GPP TR 21.900.		Use <u>one</u> of the following releases: 2 (GSM Phase 2) R96 (Release 1996) R97 (Release 1997) R98 (Release 1998) R99 (Release 1999) REL-4 (Release 4) REL-5 (Release 5)

Reason for change:	⌘	Need for support of multiple framework versions by a single instance to be able to handle backward compatibility		
Summary of change:	⌘	Adding support to the application to state it's implemented version in the lplnitial interface, the Framework implementation shall now adopt to work according to that version or return an exception if the chosen version is not supported.		
Consequences if not approved:	⌘	The version handling must then be handled by the chosen implementation, e.g. CORBA/IDL. This results in that different technologies will work differently regarding this issue. A second reason is to make it easy for both the network operators as well as the third-party developers to handle backward compatibility in the Framework access.		

Clauses affected:	⌘	6.2, 6.3.1.3, 7.2		
Other specs affected:	⌘	<input type="checkbox"/> Other core specifications <input type="checkbox"/> Test specifications <input type="checkbox"/> O&M Specifications	⌘	
Other comments:	⌘			

How to create CRs using this form:

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

- 1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.
- 2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under <ftp://ftp.3gpp.org/specs/> For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

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

6.2 Class Diagrams

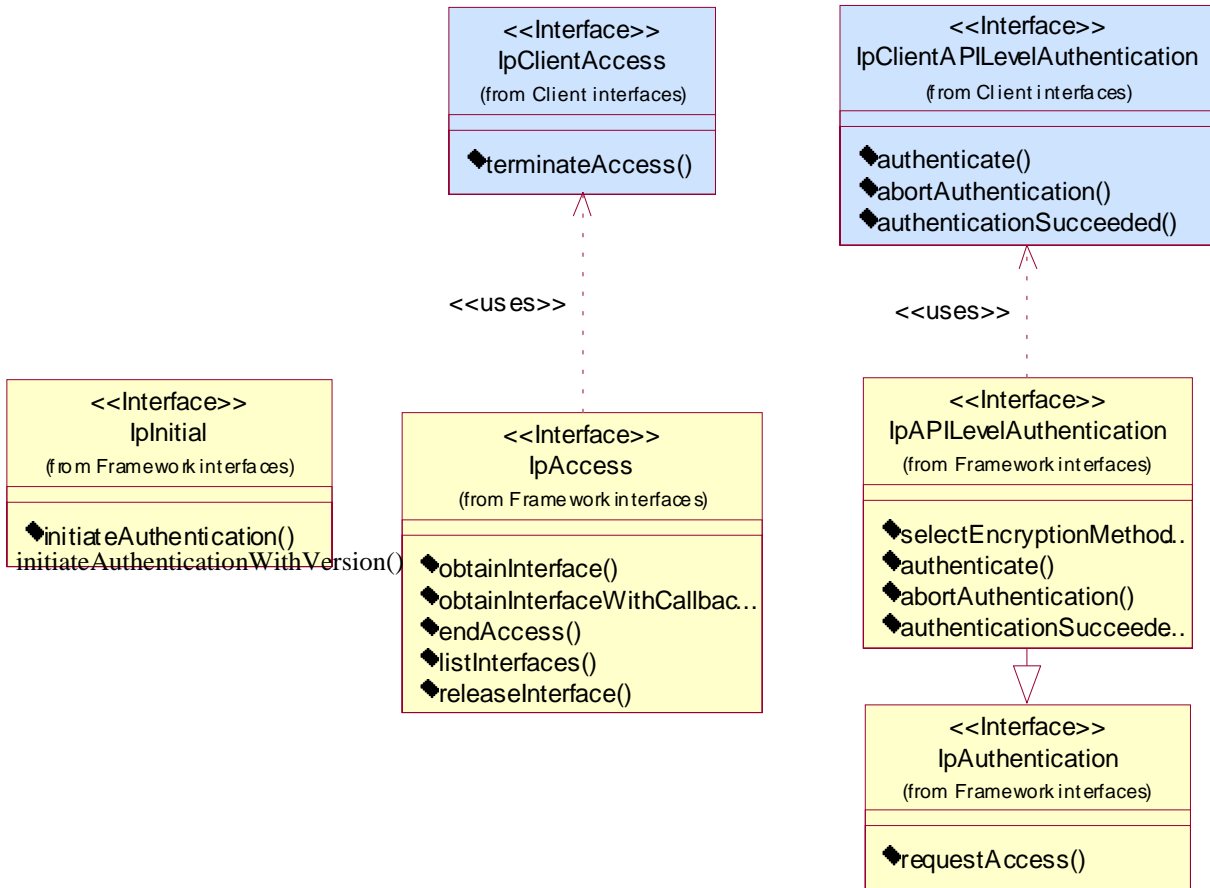
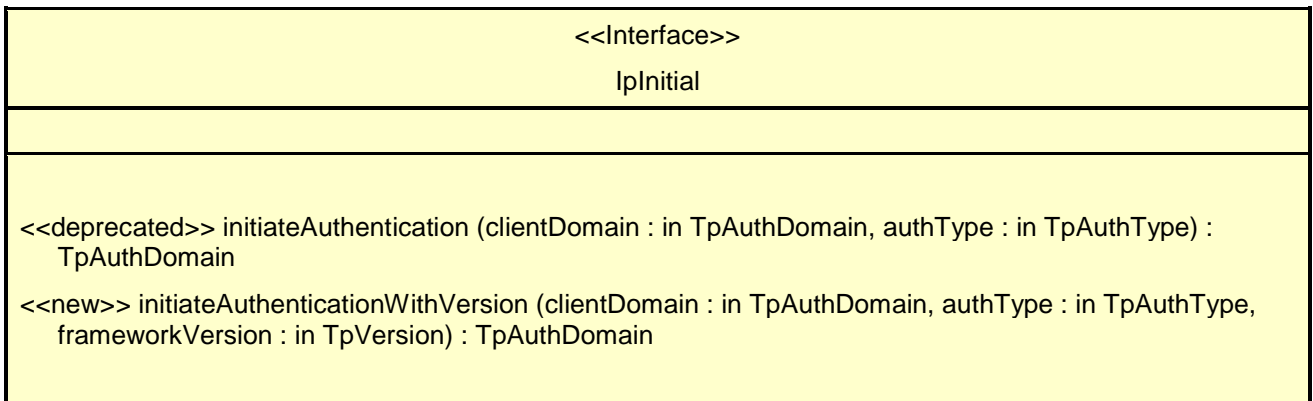


Figure: Trust and Security Management Package Overview

6.3.1.3 Interface Class IpInitial

Inherits from: IpInterface.

The Initial Framework interface is used by the client to initiate the mutual authentication with the Framework.



*Method***initiateAuthentication()**

This method is deprecated in this version, this means that it will be supported until the next major release of this specification.

This method is invoked by the client to start the process of mutual authentication with the framework, and request the use of a specific authentication method.

Returns <fwDomain> : This provides the client with a framework identifier, and a reference to call the authentication interface of the framework.

```
structure TpAuthDomain {
    domainID:    TpDomainID;
    authInterface: IpInterfaceRef;
};
```

The domainID parameter is an identifier for the framework (i.e. TpFwID). It is used to identify the framework to the client.

The authInterface parameter is a reference to the authentication interface of the framework. The type of this interface is defined by the authType parameter. The client uses this interface to authenticate with the framework.

*Parameters***clientDomain : in TpAuthDomain**

This identifies the client domain to the framework, and provides a reference to the domain's authentication interface.

```
structure TpAuthDomain {
    domainID:    TpDomainID;
    authInterface: IpInterfaceRef;
};
```

The domainID parameter is an identifier either for a client application (i.e. TpClientAppID) or for an enterprise operator (i.e. TpEntOpID), or for an existing registered service (i.e. TpServiceID) or for a service supplier (i.e. TpServiceSupplierID). It is used to identify the client domain to the framework, (see authenticate() on IpAPILevelAuthentication). If the framework does not recognise the domainID, the framework returns an error code (P_INVALID_DOMAIN_ID).

The authInterface parameter is a reference to call the authentication interface of the client. The type of this interface is defined by the authType parameter. If the interface reference is not of the correct type, the framework returns an error code (P_INVALID_INTERFACE_TYPE).

authType : in TpAuthType

This identifies the type of authentication mechanism requested by the client. It provides operators and clients with the opportunity to use an alternative to the API level Authentication interface, e.g. an implementation specific authentication mechanism like CORBA Security, using the IpAuthentication interface, or Operator specific Authentication interfaces. OSA API level Authentication is the default authentication mechanism (P_OSA_AUTHENTICATION). If P_OSA_AUTHENTICATION is selected, then the clientDomain and fwDomain authInterface parameters are references to interfaces of type Ip(Client)APILevelAuthentication. If P_AUTHENTICATION is selected, the fwDomain authInterface parameter references to interfaces of type IpAuthentication which is used when an underlying distribution technology authentication mechanism is used.

Returns

TpAuthDomain

Raises

TpCommonExceptions, P_INVALID_DOMAIN_ID, P_INVALID_INTERFACE_TYPE, P_INVALID_AUTH_TYPE

initiateAuthenticationWithVersion()

This method is invoked by the client to start the process of mutual authentication with the framework, and request the use of a specific authentication method using the new method with support for backward compatibility in the framework. The returned fwDomain authInterface will be selected to match the proposed version from the Client in the Framework response. If the Framework can't work with the proposed framework version the framework returns an error code (P_INVALID_VERSION).

Returns <fwDomain> : This provides the client with a framework identifier, and a reference to call the authentication interface of the framework.

```
structure TpAuthDomain {
    domainID:      TpDomainID;
    authInterface: IpInterfaceRef;
};
```

The domainID parameter is an identifier for the framework (i.e. TpFwID). It is used to identify the framework to the client.

The authInterface parameter is a reference to the authentication interface of the framework. The type of this interface is defined by the authType parameter. The client uses this interface to authenticate with the framework.

Parameters

clientDomain : in TpAuthDomain

This identifies the client domain to the framework, and provides a reference to the domain's authentication interface.

```
structure TpAuthDomain {
    domainID:      TpDomainID;
    authInterface: IpInterfaceRef;
};
```

The domainID parameter is an identifier either for a client application (i.e. TpClientAppID) or for an enterprise operator (i.e. TpEntOpID), or for an existing registered service (i.e. TpServiceID) or for a service supplier (i.e. TpServiceSupplierID). It is used to identify the client domain to the framework, (see authenticate() on IpAPILevelAuthentication). If the framework does not recognise the domainID, the framework returns an error code (P_INVALID_DOMAIN_ID).

The authInterface parameter is a reference to call the authentication interface of the client. The type of this interface is defined by the authType parameter. If the interface reference is not of the correct type, the framework returns an error code (P_INVALID_INTERFACE_TYPE).

authType : in TpAuthType

This identifies the type of authentication mechanism requested by the client. It provides operators and clients with the opportunity to use an alternative to the API level Authentication interface, e.g. an implementation specific authentication mechanism like CORBA Security, using the IpAuthentication interface, or Operator specific Authentication interfaces. OSA API level Authentication is the default authentication mechanism (P_OSA_AUTHENTICATION). If P_OSA_AUTHENTICATION is selected, then the clientDomain and fwDomain authInterface parameters are references to interfaces of type Ip(Client)APILevelAuthentication. If P_AUTHENTICATION is selected, the fwDomain authInterface parameter references to interfaces of type IpAuthentication that is used when an underlying distribution technology authentication mechanism is used.

frameworkVersion : in TpVersion

This identifies the version of the Framework implemented in the client. The TpVersion is a String containing the version number. Valid version numbers are defined in the respective framework specification.

Returns

TpAuthDomain

Raises

TpCommonExceptions, P_INVALID_DOMAIN_ID, P_INVALID_INTERFACE_TYPE, P_INVALID_AUTH_TYPE, P_INVALID_VERSION

7.2 Class Diagrams

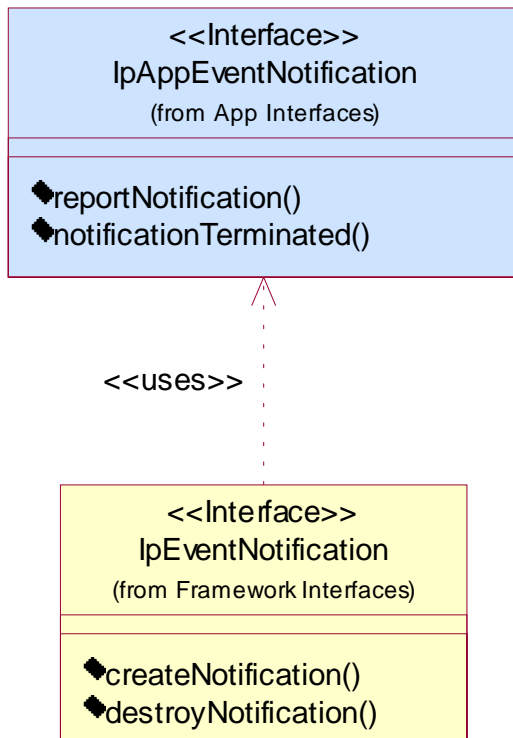


Figure: Event Notification Class Diagram

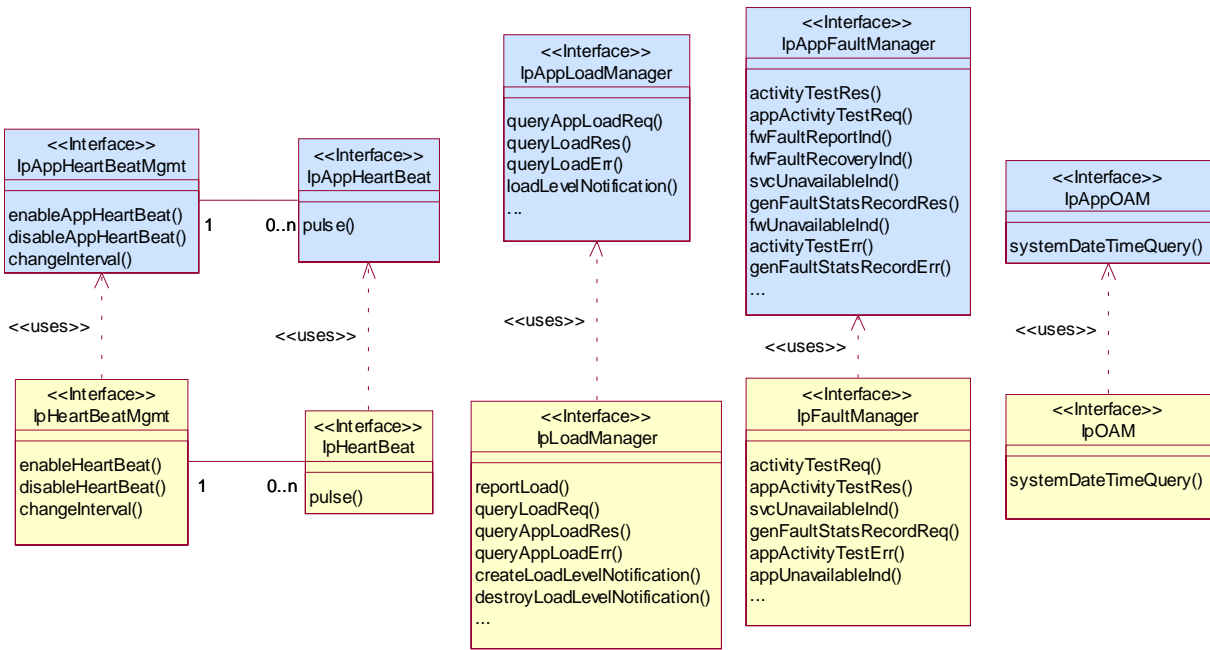


Figure: Integrity Management Package Overview

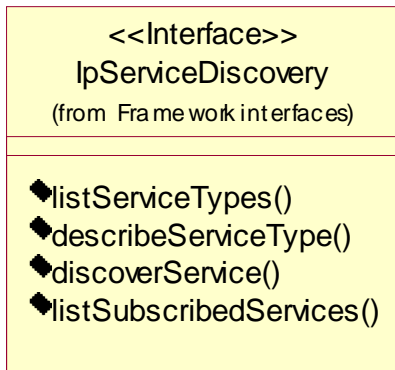


Figure: Service Discovery Package Overview

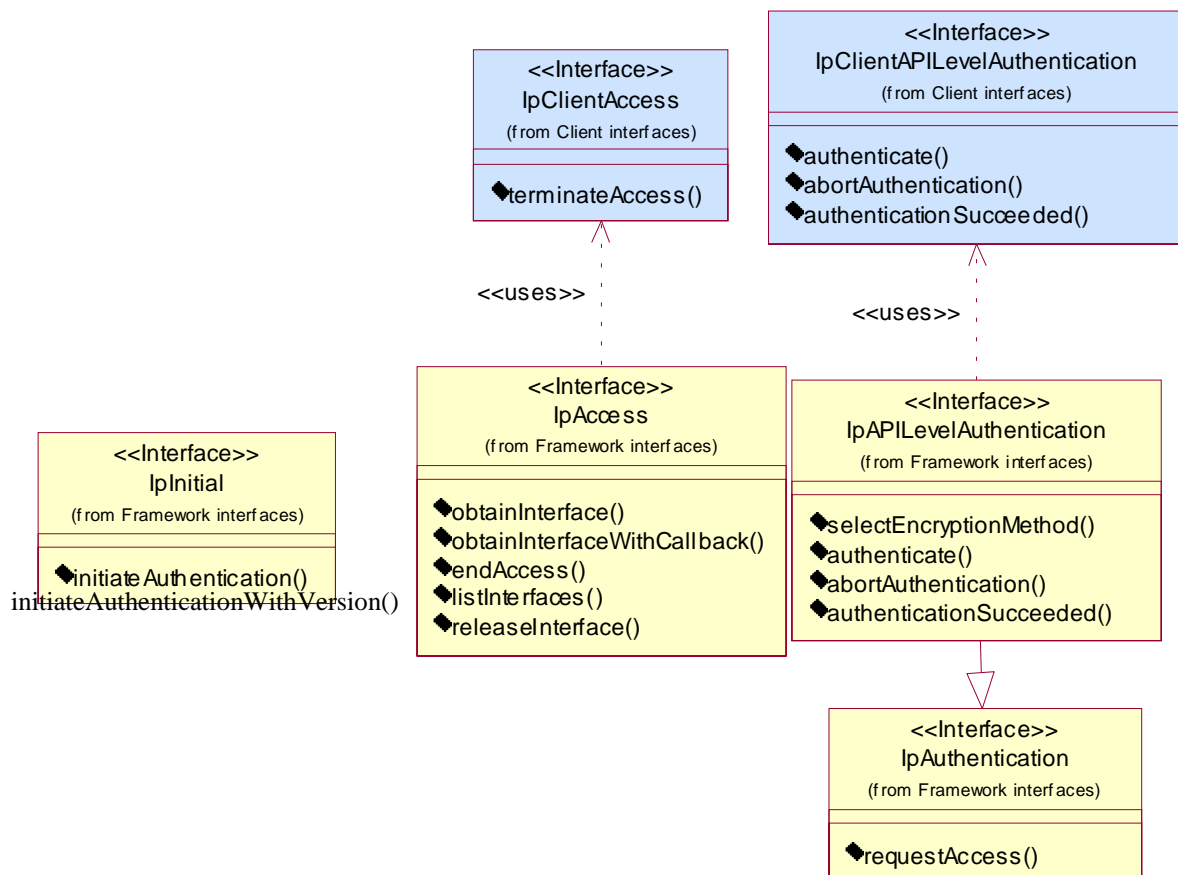


Figure: Trust and Security Management Package Overview

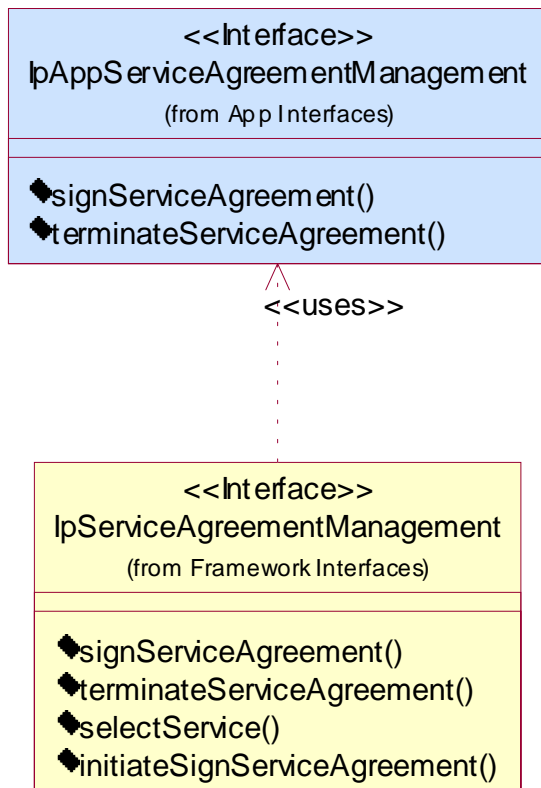


Figure: Service Agreement Management Package Overview

CHANGE REQUEST

⌘ **29.198-03 CR 038** ⌘ rev **-** ⌘ Current version: **4.5.0** ⌘

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

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

Title:	⌘ Enhancements on subscription management error information		
Source:	⌘ CN5		
Work item code:	⌘ OSA2	Date:	⌘ 17/05/2002
Category:	⌘ B	Release:	⌘ REL-5
Use <u>one</u> of the following categories:		Use <u>one</u> of the following releases:	
F (correction)		2 (GSM Phase 2)	
A (corresponds to a correction in an earlier release)		R96 (Release 1996)	
B (addition of feature),		R97 (Release 1997)	
C (functional modification of feature)		R98 (Release 1998)	
D (editorial modification)		R99 (Release 1999)	
Detailed explanations of the above categories can be found in 3GPP TR 21.900.		REL-4 (Release 4)	
		REL-5 (Release 5)	

Reason for change:	⌘ To provide detailed information on subscription management errors
Summary of change:	⌘ Addition of new exceptions and data type definitions for subscription error indication
Consequences if not approved:	⌘ Subscription data and service profiling management problems, interoperability problems

Clauses affected:	⌘ 10.5, 11
Other specs affected:	⌘ <input type="checkbox"/> Other core specifications ⌘ <input type="checkbox"/> Test specifications <input type="checkbox"/> O&M Specifications
Other comments:	⌘

How to create CRs using this form:

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

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

10.5.32 TpServiceProfileDescription

This data type is a Sequence of Data Elements which describes a Service Profile. A service contract contains one or more Service Profiles, one for each SAG in the enterprise operator domain. A service profile is a restriction of the service contract in order to provide restricted service features to a SAG. It is a structured data type which consists of:

Sequence Element Name	Sequence Element Type
ServiceContractID	<u>TpServiceContractID</u>
ServiceStartDate	<u>TpServiceStartDate</u>
ServiceEndDate	<u>TpServiceEndDate</u>
ServiceTypeName	<u>TpServiceTypeName</u>
ServiceSubscriptionProperties	<u>TpServiceSubscriptionProperties</u>

10.5.33 TpSagProfilePair

This data type is a Sequence of Data Elements which describes a pair of aSAG and a Service Profile. It is a structured data type which consists of:

Sequence Element Name	Sequence Element Type
<u>Sag</u>	<u>TpSagID</u>
<u>ServiceProfile</u>	<u>TpServiceProfileID</u>

10.5.34 TpAddSagMembersConflict

This data type is a Sequence of Data Elements which describes a conflict that may occur when client applications are added to a SAG - see method addSagMembers(). This happens, when a client application is assigned to a service twice.

The AlreadyAssignedSagProfilePair describes the SAG and the service profile through which the client application is already assigned to the service. It includes the current service profile. The ConflictGeneratingSagProfilePair describes another SAG, to which the client application should be added, and the corresponding service profile, through which the client application is also connected to this service. This creates a conflict, as there may exist only a single service profile for each service.

The TpAddSagMembersConflict is a structured data type which consists of:

Sequence Element Name	Sequence Element Type
<u>ClientApplication</u>	<u>TpClientAppID</u>
<u>ConflictGeneratingSagProfilePair</u>	<u>TpSagProfilePair</u>
<u>AlreadyAssignedSagProfilePair</u>	<u>TpSagProfilePair</u>
<u>Service</u>	<u>TpServiceID</u>

10.5.35 TpAddSagMembersConflictList

This data type defines a Numbered List of Data Elements of type TpAddSagMembersConflict.

10.5.36 TpAssignSagToServiceProfileConflict

This data type is a Sequence of Data Elements which describes a conflict that may occur when a SAG is assigned to a Service Profile - see method assign().

The AlreadyAssignedSagProfilePair describes the SAG and the service profile through which the client application is already assigned to the service.

The TpAssignSagToServiceProfileConflict is a structured data type which consists of:

<u>Sequence Element Name</u>	<u>Sequence Element Type</u>
<u>ClientApplication</u>	<u>TpClientAppID</u>
<u>AlreadyAssignedSagProfilePair</u>	<u>TpSagProfilePair</u>
<u>Service</u>	<u>TpServiceID</u>

10.5.37 TpAssignSagToServiceProfileConflictList

This data type defines a Numbered List of Data Elements of type TpAssignSagToServiceProfileConflict.

11 Exception Classes

The following are the list of exception classes which are used in this interface of the API.

Name	Description
P_ACCESS_DENIED	The client is not currently authenticated with the framework
P_APPLICATION_NOT_ACTIVATED	An application is unauthorised to access information and request services with regards to users that have deactivated that particular application.
P_DUPLICATE_PROPERTY_NAME	A duplicate property name has been received
P_ILLEGAL_SERVICE_ID	Illegal Service ID
P_ILLEGAL_SERVICE_TYPE	Illegal Service Type
P_INVALID_ACCESS_TYPE	The framework does not support the type of access interface requested by the client.
P_INVALID_ACTIVITY_TEST_ID	ID does not correspond to a valid activity test request
<u>P_INVALID_ADDITION_TO_SAG</u>	<u>A client application cannot be added to the SAG because this would imply that the client application has two concurrent service profiles at a particular moment in time for a particular service.</u>
P_INVALID_AGREEMENT_TEXT	Invalid agreement text
P_INVALID_ENCRYPTION_CAPABILITY	Invalid encryption capability
P_INVALID_AUTH_TYPE	Invalid type of authentication mechanism
P_INVALID_CLIENT_APP_ID	Invalid Client Application ID
P_INVALID_DOMAIN_ID	Invalid client ID
P_INVALID_ENT_OP_ID	Invalid Enterprise Operator ID
P_INVALID_PROPERTY	The framework does not recognise the property supplied by the client
P_INVALID_SAG_ID	Invalid Subscription Assignment Group ID
<u>P_INVALID_SAG_TO_SERVICE_PROFILE_ASSIGNMENT</u>	<u>A SAG cannot be assigned to the service profile because this would imply that a client application has two concurrent service profiles at a particular moment in time for a particular service.</u>
P_INVALID_SERVICE_CONTRACT_ID	Invalid Service Contract ID
P_INVALID_SERVICE_ID	Invalid service ID
P_INVALID_SERVICE_PROFILE_ID	Invalid service profile ID
P_INVALID_SERVICE_TOKEN	The service token has not been issued, or it has expired.
P_INVALID_SERVICE_TYPE	Invalid Service Type
P_INVALID_SIGNATURE	Invalid digital signature
P_INVALID_SIGNING_ALGORITHM	Invalid signing algorithm
P_MISSING_MANDATORY_PROPERTY	Mandatory Property Missing
P_NO_ACCEPTABLE_ENCRYPTION_CAPABILITY	An encryption mechanism, which is acceptable to the framework, is not supported by the client
P_PROPERTY_TYPE_MISMATCH	Property Type Mismatch
P_SERVICE_ACCESS_DENIED	The client application is not allowed to access this service.
P_SERVICE_NOT_ENABLED	The service ID does not correspond to a service that has been enabled
P_SERVICE_TYPE_UNAVAILABLE	The service type is not available according to the Framework.
P_UNKNOWN_SERVICE_ID	Unknown Service ID

Name	Description
P_UNKNOWN_SERVICE_TYPE	Unknown Service Type

Each exception class contains the following structure:

Structure Element Name	Structure Element Type	Structure Element Description
ExtraInformation	TpString	Carries extra information to help identify the source of the exception, e.g. a parameter name

CHANGE REQUEST

⌘ **29.198-03 CR 039** ⌘ rev **-** ⌘ Current version: **4.5.0** ⌘

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

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

Title:	⌘ Delete conflicting description of P_APPLICATION_NOT_ACTIVATED				
Source:	⌘ CN5				
Work item code:	⌘ OSA2	Date:	⌘ 17/05/2002		
Category:	⌘ F	Release:	⌘ REL-5		
	Use <u>one</u> of the following categories:		Use <u>one</u> of the following releases:		
	F (correction)		2 (GSM Phase 2)		
	A (corresponds to a correction in an earlier release)		R96 (Release 1996)		
	B (addition of feature),		R97 (Release 1997)		
	C (functional modification of feature)		R98 (Release 1998)		
	D (editorial modification)		R99 (Release 1999)		
	Detailed explanations of the above categories can be found in 3GPP TR 21.900.		REL-4 (Release 4)		
			REL-5 (Release 5)		

Reason for change:	⌘ The description of P_APPLICATION_NOT_ACTIVATED in this Part 3 is in conflict with Part 2 (29.198-2). The IDL only defines P_APPLICATION_NOT_ACTIVATED in Part 2.
Summary of change:	⌘ Delete the description of P_APPLICATION_NOT_ACTIVATED in Part 3
Consequences if not approved:	⌘ Multiple descriptions of P_APPLICATION_NOT_ACTIVATED may lead to confusion.

Clauses affected:	⌘ 11
Other specs affected:	⌘ <input type="checkbox"/> Other core specifications ⌘ <input type="checkbox"/>
	<input type="checkbox"/> Test specifications ⌘ <input type="checkbox"/>
	<input type="checkbox"/> O&M Specifications ⌘ <input type="checkbox"/>
Other comments:	⌘ <input type="text"/>

How to create CRs using this form:

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

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

11 Exception Classes

The following are the list of exception classes which are used in this interface of the API.

Name	Description
P_ACCESS_DENIED	The client is not currently authenticated with the framework
P_APPLICATION_NOT_ACTIVATED	An application is unauthorised to access information and request services with regards to users that have deactivated that particular application.
P_DUPLICATE_PROPERTY_NAME	A duplicate property name has been received
P_ILLEGAL_SERVICE_ID	Illegal Service ID
P_ILLEGAL_SERVICE_TYPE	Illegal Service Type
P_INVALID_ACCESS_TYPE	The framework does not support the type of access interface requested by the client.
P_INVALID_ACTIVITY_TEST_ID	ID does not correspond to a valid activity test request
P_INVALID_AGREEMENT_TEXT	Invalid agreement text
P_INVALID_ENCRYPTION_CAPABILITY	Invalid encryption capability
P_INVALID_AUTH_TYPE	Invalid type of authentication mechanism
P_INVALID_CLIENT_APP_ID	Invalid Client Application ID
P_INVALID_DOMAIN_ID	Invalid client ID
P_INVALID_ENT_OP_ID	Invalid Enterprise Operator ID
P_INVALID_PROPERTY	The framework does not recognise the property supplied by the client
P_INVALID_SAG_ID	Invalid Subscription Assignment Group ID
P_INVALID_SERVICE_CONTRACT_ID	Invalid Service Contract ID
P_INVALID_SERVICE_ID	Invalid service ID
P_INVALID_SERVICE_PROFILE_ID	Invalid service profile ID
P_INVALID_SERVICE_TOKEN	The service token has not been issued, or it has expired.
P_INVALID_SERVICE_TYPE	Invalid Service Type
P_INVALID_SIGNATURE	Invalid digital signature
P_INVALID_SIGNING_ALGORITHM	Invalid signing algorithm
P_MISSING_MANDATORY_PROPERTY	Mandatory Property Missing
P_NO_ACCEPTABLE_ENCRYPTION_CAPABILITY	An encryption mechanism, which is acceptable to the framework, is not supported by the client
P_PROPERTY_TYPE_MISMATCH	Property Type Mismatch
P_SERVICE_ACCESS_DENIED	The client application is not allowed to access this service.
P_SERVICE_NOT_ENABLED	The service ID does not correspond to a service that has been enabled
P_SERVICE_TYPE_UNAVAILABLE	The service type is not available according to the Framework.
P_UNKNOWN_SERVICE_ID	Unknown Service ID
P_UNKNOWN_SERVICE_TYPE	Unknown Service Type

CHANGE REQUEST

⌘ **29.198-03 CR 040** ⌘ rev **-** ⌘ Current version: **4.5.0** ⌘

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

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

Title:	⌘ Note added for P_SERVICE_INSTANCE Choice Element Name				
Source:	⌘ CN5				
Work item code:	⌘ OSA2	Date:	⌘ 17/05/2002		
Category:	⌘ D	Release:	⌘ REL-5		
	Use <u>one</u> of the following categories:		Use <u>one</u> of the following releases:		
	F (correction)		2	(GSM Phase 2)	
	A (corresponds to a correction in an earlier release)		R96	(Release 1996)	
	B (addition of feature),		R97	(Release 1997)	
	C (functional modification of feature)		R98	(Release 1998)	
	D (editorial modification)		R99	(Release 1999)	
	Detailed explanations of the above categories can be found in 3GPP TR 21.900.		REL-4	(Release 4)	
			REL-5	(Release 5)	

Reason for change:	⌘ The choice element name for P_SERVICE_INSTANCE in TpDomainID should have been ServiceInstanceID, not ServiceID
Summary of change:	⌘ Note added that the ServiceID refers to a service instance
Consequences if not approved:	⌘ Could cause confusion to developers

Clauses affected:	⌘ 10.1.3
Other specs affected:	⌘ <input type="checkbox"/> Other core specifications ⌘ <input type="checkbox"/> Test specifications ⌘ <input type="checkbox"/> O&M Specifications
Other comments:	⌘

How to create CRs using this form:

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

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

10.1.3 TpDomainID

Defines the Tagged Choice of Data Elements that specify either the Framework or the type of entity attempting to access the Framework.

	Tag Element Type	
	<u>TpDomainIDType</u>	

Tag Element Value	Choice Element Type	Choice Element Name
P_FW	TpFwID	FwID
P_CLIENT_APPLICATION	TpClientAppID	ClientAppID
P_ENT_OP	TpEntOpID	EntOpID
P_SERVICE_INSTANCE	TpServiceInstanceID	ServiceID_(See Note)
P_SERVICE_SUPPLIER	TpServiceSupplierID	ServiceSupplierID
NOTE: <u>The Choice Element Name ServiceID of TpDomainID refers to a service instance.</u>		

CHANGE REQUEST

⌘ **29.198-03 CR 041** ⌘ rev **-** ⌘ Current version: **4.5.0** ⌘

For [HELP](#) on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.

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

Title:	⌘	Correcting the method descriptions for abortAuthentication and for initiateAuthentication
Source:	⌘	CN5
Work item code:	⌘	OSA2
		Date: ⌘ 30/05/2002
Category:	⌘	F
		<div style="display: flex; justify-content: space-between;"> <div style="width: 60%;"> <p>Use <u>one</u> of the following categories:</p> <p>F (correction)</p> <p>A (corresponds to a correction in an earlier release)</p> <p>B (addition of feature),</p> <p>C (functional modification of feature)</p> <p>D (editorial modification)</p> <p>Detailed explanations of the above categories can be found in 3GPP TR 21.900.</p> </div> <div style="width: 35%;"> <p>Use <u>one</u> of the following releases:</p> <p>2 (GSM Phase 2)</p> <p>R96 (Release 1996)</p> <p>R97 (Release 1997)</p> <p>R98 (Release 1998)</p> <p>R99 (Release 1999)</p> <p>REL-4 (Release 4)</p> <p>REL-5 (Release 5)</p> </div> </div>

Reason for change:	⌘	The description of the clientDomain parameter in initiateAuthentication is incorrect, as it references TpServiceID being part of the TpDomainID. This data type is not a part of TpDomainID. Also, the description of abortAuthentication is incorrect. This method is called by the client on the framework when the client wants to abort the authentication, but the description erroneously switches "client" and "framework".
Summary of change:	⌘	Alter the descriptions of initiateAuthentication and abortAuthentication to give an accurate representation of their functionality.
Consequences if not approved:	⌘	The method descriptions will not correspond to the functionality contained in the methods, which will result in implementation differences and interoperability issues.

Clauses affected:	⌘	6.3.1.3, 6.3.1.5
Other specs affected:	⌘	<input type="checkbox"/> Other core specifications ⌘ <input type="checkbox"/> Test specifications <input type="checkbox"/> O&M Specifications
Other comments:	⌘	

How to create CRs using this form:

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

- 1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.
- 2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under <ftp://ftp.3gpp.org/specs/> For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

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

6.3.1.3 Interface Class IpInitial

Inherits from: IpInterface.

The Initial Framework interface is used by the client to initiate the mutual authentication with the Framework.

<<Interface>> IpInitial
initiateAuthentication (clientDomain : in TpAuthDomain, authType : in TpAuthType) : TpAuthDomain

Method

initiateAuthentication()

This method is invoked by the client to start the process of mutual authentication with the framework, and request the use of a specific authentication method.

Returns <fwDomain> : This provides the client with a framework identifier, and a reference to call the authentication interface of the framework.

```
structure TpAuthDomain {
    domainID:          TpDomainID;
    authInterface:     IpInterfaceRef;
};
```

The domainID parameter is an identifier for the framework (i.e. TpFwID). It is used to identify the framework to the client.

The authInterface parameter is a reference to the authentication interface of the framework. The type of this interface is defined by the authType parameter. The client uses this interface to authenticate with the framework.

Parameters

clientDomain : in TpAuthDomain

This identifies the client domain to the framework, and provides a reference to the domain's authentication interface.

```
structure TpAuthDomain {
    domainID:          TpDomainID;
    authInterface:     IpInterfaceRef;
};
```

The domainID parameter is an identifier either for a client application (i.e. TpClientAppID) or for an enterprise operator (i.e. TpEntOpID), or for an [instance of a registered service \(i.e. TpServiceInstanceID\)](#)~~existing registered service (i.e. TpServiceID)~~ or for a service supplier (i.e. TpServiceSupplierID). It is used to identify the client domain to the framework, (see authenticate() on

IpAPILevelAuthentication). If the framework does not recognise the domainID, the framework returns an error code (P_INVALID_DOMAIN_ID).

The authInterface parameter is a reference to call the authentication interface of the client. The type of this interface is defined by the authType parameter. If the interface reference is not of the correct type, the framework returns an error code (P_INVALID_INTERFACE_TYPE).

authType : in TpAuthType

This identifies the type of authentication mechanism requested by the client. It provides operators and clients with the opportunity to use an alternative to the API level Authentication interface, e.g. an implementation specific authentication mechanism like CORBA Security, using the IpAuthentication interface, or Operator specific Authentication interfaces. OSA API level Authentication is the default authentication mechanism (P_OSA_AUTHENTICATION). If P_OSA_AUTHENTICATION is selected, then the clientDomain and fwDomain authInterface parameters are references to interfaces of type Ip(Client)APILevelAuthentication. If P_AUTHENTICATION is selected, the fwDomain authInterface parameter references to interfaces of type IpAuthentication which is used when an underlying distribution technology authentication mechanism is used.

Returns

TpAuthDomain

Raises

**TpCommonExceptions, P_INVALID_DOMAIN_ID,
P_INVALID_INTERFACE_TYPE, P_INVALID_AUTH_TYPE**

6.3.1.5 Interface Class IpAPILevelAuthentication

Inherits from: IpAuthentication.

The API Level Authentication Framework interface is used by client 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>> IpAPILevelAuthentication
selectEncryptionMethod (encryptionCaps : in TpEncryptionCapabilityList) : TpEncryptionCapability authenticate (challenge : in TpOctetSet) : TpOctetSet abortAuthentication () : void authenticationSucceeded () : void

Method

selectEncryptionMethod()

The client uses this method to initiate the authentication process. The framework returns its preferred mechanism. This should be within capability of the client. If a mechanism that is acceptable to the framework within the capability of the client cannot be found, the framework throws the `P_NO_ACCEPTABLE_ENCRYPTION_CAPABILITY` exception. Once the framework has returned its preferred mechanism, it will wait for a predefined unit of time before invoking the client's `authenticate()` method (the wait is to ensure that the client can initialise any resources necessary to use the prescribed encryption method). Returns `<prescribedMethod>` : This is returned by the framework to indicate the mechanism preferred by the framework for the encryption process. If the value of the `prescribedMethod` returned by the framework is not understood by the client, it is considered a catastrophic error and the client must abort.

Parameters

encryptionCaps : in TpEncryptionCapabilityList

This is the means by which the encryption mechanisms supported by the client are conveyed to the framework.

Returns

TpEncryptionCapability

Raises

**TpCommonExceptions, P_ACCESS_DENIED,
P_NO_ACCEPTABLE_ENCRYPTION_CAPABILITY**

Method

authenticate()

This method is used by the client to authenticate the framework. The challenge will be encrypted using the mechanism prescribed by `selectEncryptionMethod`. The framework must respond with the correct responses to the challenges presented by the client. The `domainID` received in the `initiateAuthentication()` can be used by the framework to reference the correct public key for the client (the key management system is currently outside of the scope of the OSA APIs). The number of exchanges is dependent on the policies of each side. The whole authentication process is deemed successful when the `authenticationSucceeded` method is invoked. The invocation of this method may be interleaved with `authenticate()` calls by the framework on the client's `APILevelAuthentication` interface.

Returns `<response>` : This is the response of the framework to the challenge of the client in the current sequence. The response will be based on the challenge data, decrypted with the mechanism prescribed by `selectEncryptionMethod()`.

Parameters

challenge : in TpOctetSet

The challenge presented by the client to be responded to by the framework. The challenge mechanism used will be in accordance with the IETF PPP Authentication

Protocols - Challenge Handshake Authentication Protocol [RFC 1994, August 1996].
The challenge will be encrypted with the mechanism prescribed by
selectEncryptionMethod().

Returns

IpOctetSet

Raises

IpCommonExceptions, P_ACCESS_DENIED

Method

abortAuthentication()

The client uses this method to abort the authentication process. This method is invoked if the client no longer wishes to continue the authentication process, (unless the ~~client~~ framework responded incorrectly to a challenge in which case no further communication with the framework ~~client~~ should occur.) If this method has been invoked, calls to the requestAccess operation on IpAPILevelAuthentication will return an error code (P_ACCESS_DENIED), until the client has been properly authenticated.

Parameters

No Parameters were identified for this method

Raises

IpCommonExceptions, P_ACCESS_DENIED

Method

authenticationSucceeded()

The client uses this method to inform the framework of the success of the authentication attempt.

Parameters

No Parameters were identified for this method

Raises

IpCommonExceptions, P_ACCESS_DENIED

CHANGE REQUEST

⌘ **29.198-03 CR 042** ⌘ rev **-** ⌘ Current version: **4.5.0** ⌘

For [HELP](#) on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.

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

Title:	⌘ Correcting the description of heartbeat failure		
Source:	⌘ CN5		
Work item code:	⌘ OSA2	Date:	⌘ 30/05/2002
Category:	⌘ F	Release:	⌘ REL-5
	Use <u>one</u> of the following categories: F (correction) A (corresponds to a correction in an earlier release) B (addition of feature), C (functional modification of feature) D (editorial modification) Detailed explanations of the above categories can be found in 3GPP TR 21.900 .		Use <u>one</u> of the following releases: 2 (GSM Phase 2) R96 (Release 1996) R97 (Release 1997) R98 (Release 1998) R99 (Release 1999) REL-4 (Release 4) REL-5 (Release 5)

Reason for change:	⌘ In the definition of IpHeartBeat.pulse(), the specification incorrectly states that if this method is not invoked, then the framework can be deemed to have failed the heartbeat. It is actually the client application that can be deemed to have failed the heartbeat as it has not made the invocation.
Summary of change:	⌘ Change the text to state that if the heartbeat is not received, then the client application can be deemed to have failed the heartbeat.
Consequences if not approved:	⌘ Implementations of the heartbeat mechanism may be incorrect, which will affect interoperability.

Clauses affected:	⌘ 7.3.3.6		
Other specs affected:	⌘ <input type="checkbox"/> Other core specifications <input type="checkbox"/> Test specifications <input type="checkbox"/> O&M Specifications	⌘	
Other comments:	⌘		

How to create CRs using this form:

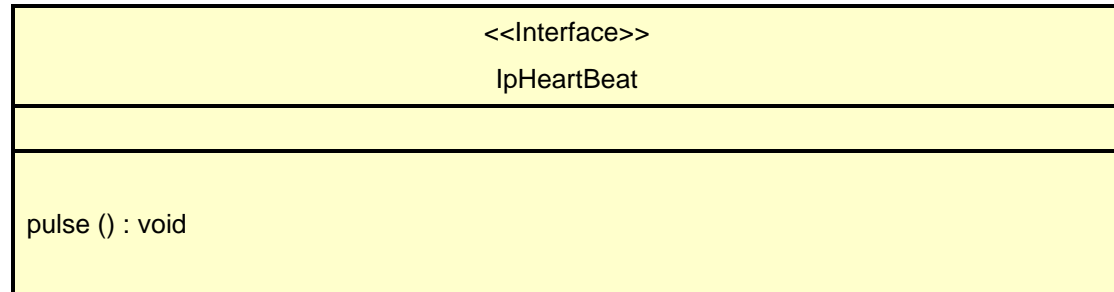
Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

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

7.3.3.6 Interface Class IpHeartBeat

Inherits from: IpInterface.

The Heartbeat Framework interface is used by the client application to send its heartbeat.



Method

pulse ()

The client application uses this method to send its heartbeat to the framework. The framework will be expecting a pulse at the end of every interval specified in the parameter to the IpAppHeartBeatMgmt.enableAppHeartbeat() method. If the pulse() is not received within the specified interval, then the ~~framework~~-client application can be deemed to have failed the heartbeat.

Parameters

No Parameters were identified for this method

Raises

TpCommonExceptions

CHANGE REQUEST

⌘ **29.198-03 CR 043** ⌘ rev **-** ⌘ Current version: **4.5.0** ⌘

For [HELP](#) on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.

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

Title:	⌘	Correcting erroneous FW<->Service instance sequence diagrams		
Source:	⌘	CN5		
Work item code:	⌘	OSA2	Date:	⌘ 30/05/2002
Category:	⌘	F	Release:	⌘ REL-5
		Use <u>one</u> of the following categories:		Use <u>one</u> of the following releases:
		F (correction)	2	(GSM Phase 2)
		A (corresponds to a correction in an earlier release)	R96	(Release 1996)
		B (addition of feature),	R97	(Release 1997)
		C (functional modification of feature)	R98	(Release 1998)
		D (editorial modification)	R99	(Release 1999)
		Detailed explanations of the above categories can be found in 3GPP TR 21.900 .	REL-4	(Release 4)
			REL-5	(Release 5)

Reason for change:	⌘	8.1.2 states erroneously that createServiceManager can be invoked at any time before an application attempts to discover the service. This is incorrect as the method is actually invoked only when the client has signed a service agreement. Also, 8.1.4.5 states that an applID parameter is used to identify which application the service wants to invoke an activity test on. This is incorrect as it should be "service instance", not "service", and the service instance only knows about one application, so there is no applID parameter.
Summary of change:	⌘	Alter 8.1.2 to state that createServiceManager is invoked only when the application has signed a service agreement. Alter 8.1.4.5 to state that the "service instance asks the framework to invoke an activity test on the client application."
Consequences if not approved:	⌘	The behaviour of the framework when a service registers and when a service requests an activity test on an application may vary between implementations, leading to interoperability issues.

Clauses affected:	⌘	8.1.2, 8.1.4.5
Other specs affected:	⌘	<input type="checkbox"/> Other core specifications <input type="checkbox"/> Test specifications <input type="checkbox"/> O&M Specifications
Other comments:	⌘	

How to create CRs using this form:

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

- 1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.
- 2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be

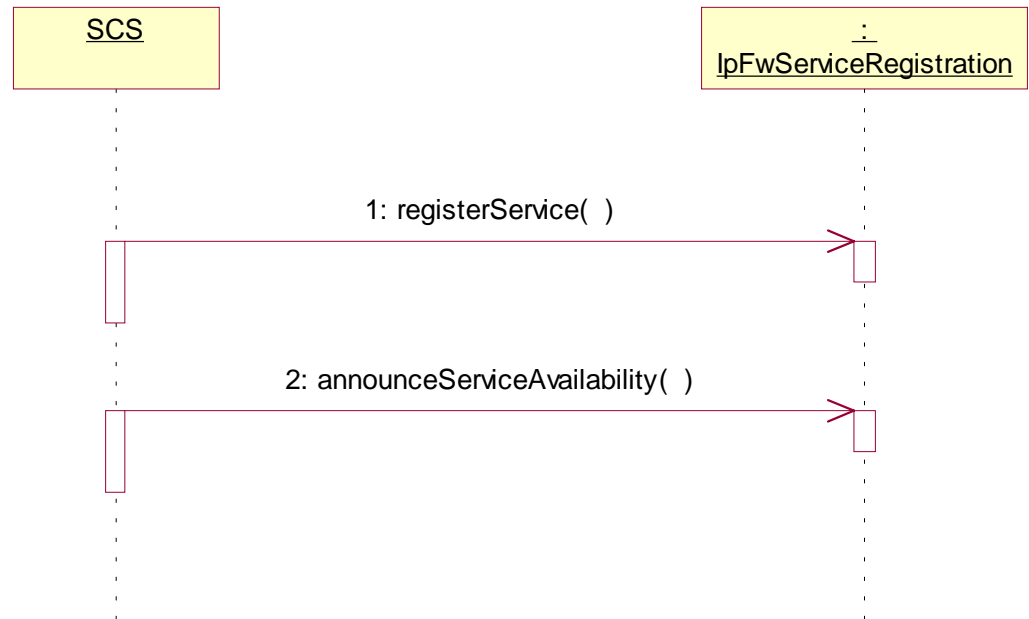
downloaded from the 3GPP server under <ftp://ftp.3gpp.org/specs/> For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

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

8.1.2 Service Registration Sequence Diagrams

8.1.2.1 New SCF Registration

The following figure shows the process of registering a new Service Capability Feature in the Framework. Service Registration is a two step process:



1: Registration: first step - register service

The purpose of this first step in the process of registration is to agree, within the network, on a name to call, internally, a newly installed SCF version. It is necessary because the OSA Framework and SCF in the same network may come from different vendors. The goal is to make an association between the new SCF version, as characterized by a list of properties, and an identifier called serviceID.

This service ID will be the name used in that network (that is, between that network's Framework and its SCSs), whenever it is necessary to refer to this newly installed version of SCF (for example for announcing its availability, or for withdrawing it later).

The following input parameters are given from the SCS to the Framework in this first registration step:

- `serviceTypeName`

This is a string with the name of the SCF, among a list of standard names (e.g. "P_MPCC").

- `servicePropertyList`

This is a list of types `TpServiceProperty`; each `TpServiceProperty` is a pair of (`ServicePropertyName`, `ServicePropertyValueList`).

- `ServicePropertyName` is a string that defines a valid SCF property name (valid SCF property names are listed in the SCF data definition).

- `ServicePropertyValueList` is a numbered set of types `TpServicePropertyValue`; `TpServicePropertyValue` is a string that describes a valid value of a SCF property (valid SCF property values are listed in the SCF data definition).

The following output parameter results from service registration:

- out serviceID

This is a string, automatically generated by the Framework and unique within the Framework.

This is the name by which the newly installed version of SCF, described by the list of properties above, is going to be identified internally in this network.

2: Registration: second step - announce service availability

At this point the network's Framework is aware of the existence of a new SCF, and could let applications know - but they would have no way to use it. Installing the SCS logic and assigning a name to it does not make this SCF available. In order to make the SCF available an "entry point", called lifecycle manager, is used. The role of the lifecycle manager is to control the life cycle of an interface, or set of interfaces, and provide clients with the references that are necessary to invoke the methods offered by these interfaces. The starting point for a client to use an SCF is to obtain an interface reference to a lifecycle manager of the desired SCF.

A Network Operator, upon completion of the first registration phase, and once it has an identifier to the new SCF version, will instantiate a lifecycle manager for it that will allow client to use it. Then it will inform the Framework of the value of the interface associated to the new SCF. After the receipt of this information, the Framework makes the new SCF (identified by the pair [serviceID, serviceInstanceLifecycleManagerRef]) discoverable.

The following input parameters are given from the SCS to the Framework in this second registration step:

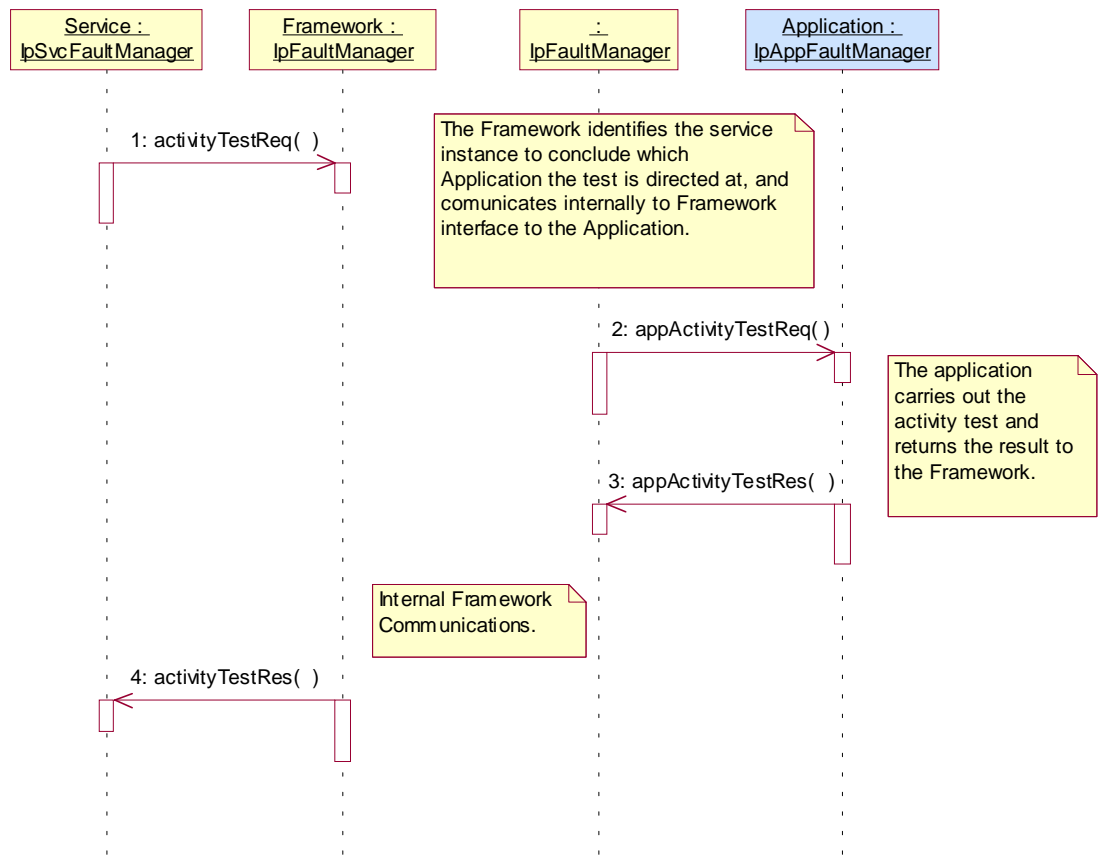
- in serviceID

This is the identifier that has been agreed in the network for the new SCF; any interaction related to the SCF needs to include the serviceID, to know which SCF it is.

- in serviceInstanceLifecycleManagerRef

This is the interface reference at which the lifecycle manager of the new SCF is available. Note that the Framework will have to invoke the method `createServiceManager()` in this interface; when a client application signs an agreement to use the SCF any time between now and when it accepts the first application requests for discovery, so that it can get the service manager interface necessary for applications as an entry point to any SCF.

8.1.4.5 Fault Management: Service requests Application activity test



- 1: The service instance asks the framework to invoke an activity test on a the client application, ~~the application is identified by the appId parameter.~~
- 2: The framework asks the application to do the activity test. It is assumed that there is internal communication between the service facing part of the framework (i.e. IpFwFaultManager interface) and the part that faces the client application.
- 3: The application does the activity test and returns the result to the framework.
- 4: The framework internally passes the result from its application facing interface (IpFaultManager) to its service facing side, and sends the result to the service.

CHANGE REQUEST

⌘ **29.198-03 CR 044** ⌘ rev **-** ⌘ Current version: **4.5.0** ⌘

For [HELP](#) on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.

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

Title:	⌘ Correcting the scope of TpFwID, which currently is giving it false limitations		
Source:	⌘ CN5		
Work item code:	⌘ OSA2	Date:	⌘ 30/05/2002
Category:	⌘ F	Release:	⌘ REL-5
	Use <u>one</u> of the following categories: F (correction) A (corresponds to a correction in an earlier release) B (addition of feature), C (functional modification of feature) D (editorial modification) Detailed explanations of the above categories can be found in 3GPP TR 21.900 .		Use <u>one</u> of the following releases: 2 (GSM Phase 2) R96 (Release 1996) R97 (Release 1997) R98 (Release 1998) R99 (Release 1999) REL-4 (Release 4) REL-5 (Release 5)

Reason for change:	⌘ The description of TpFwID states that it identifies the Framework to a client application or an SCF. The ID is also used to identify the Framework to service suppliers and enterprise operators. The current scope is incorrect and gives the TpFwID false limitations.
Summary of change:	⌘ The text describing TpFwID should be shortened to: This data type is identical to TpString and identifies the Framework.
Consequences if not approved:	⌘ Possible misunderstanding between vendors over the scope of the TpFwID.

Clauses affected:	⌘ 10.1.11
Other specs affected:	⌘ <input type="checkbox"/> Other core specifications ⌘ <input type="checkbox"/> Test specifications <input type="checkbox"/> O&M Specifications
Other comments:	⌘

How to create CRs using this form:

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

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

10.1.11 TpFwID

This data type is identical to [TpString](#) and identifies the Framework ~~to a client application (or Service Capability Feature)~~.