3GPP TSG-CN Meeting #24 **NP-040261**
02 – 04 June 2004, Seoul, KOREA

Source: **CN5 (OSA)**

Title: **12 Rel-5 CRs 29.198-03 OSA API Part 3: Framework**

Agenda item: **8.2 (OSA Enhancements [OSA2])**

Document for: **APPROVAL**

Doc-1st-	Spec	CR	Rev	Phase	Subject	Cat	Version	Doc-2nd-	Workite
NP-040261	29.198-03	102	-	Rel-5	Add ability to identify when a client app/service contract/service profile is being used - Align between ETSI/Parlay and 3GPP	F	5.6.0	N5-040056	OSA2
NP-040261	29.198-03	103	-	Rel-6	Add ability to identify when a client app/service contract/service profile is being used - Align between ETSI/Parlay and 3GPP	A	6.0.1	N5-040057	OSA2
NP-040261	29.198-03	107	-	Rel-5	Introduce a ServiceID field to TpServiceProfileDescription	F	5.6.0	N5-040060	OSA2
NP-040261	29.198-03	108	-	Rel-6	Introduce a ServiceID field to TpServiceProfileDescription	A	6.0.1	N5-040061	OSA2
NP-040261	29.198-03	114	-	Rel-5	Correct description of availStatusReason codes	F	5.6.0	N5-040349	OSA2
NP-040261	29.198-03	115	-	Rel-6	Correct description of availStatusReason codes	A	6.0.1	N5-040350	OSA2
NP-040261	29.198-03	116	-	Rel-5	Correct description for the use of selectSigningAlgorithm	F	5.6.0	N5-040351	OSA2
NP-040261	29.198-03	117	-	Rel-6	Correct description for the use of selectSigningAlgorithm	A	6.0.1	N5-040352	OSA2
NP-040261	29.198-03	118	-	Rel-5	Correct the description of the usage of CHAP within authentication	F	5.6.0	N5-040353	OSA2
NP-040261	29.198-03	119	-	Rel-6	Correct the description of the usage of CHAP within authentication	A	6.0.1	N5-040354	OSA2
NP-040261	29.198-03	120	-	Rel-5	Correct TpSignatureAndServiceMgr to align with description in signServiceAgreement	F	5.6.0	N5-040355	OSA2
NP-040261	29.198-03	121	-	Rel-6	Correct TpSignatureAndServiceMgr to align with description in signServiceAgreement	A	6.0.1	N5-040356	OSA2

CR-Form-v7

CHANGE REQUEST

⌘ 29.198-03 CR 102 ⌘rev - ⌘ Current version: 5.6.0 ⌘

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

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network X

Title:	⌘	Add ability to identify when a client app/service contract/service profile is being used - Align between ETSI/Parlay and 3GPP
Source:	⌘	CN5 Parlay Gareth Carroll (Open API Solutions)
Work item code:	⌘	OSA2
Date:	⌘	09/02/2004
Category:	⌘	F
Release:	⌘	*REL-5*

Use one 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 one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change:	⌘	In the ETSI/Parlay version of OSA, the Enterprise Operator interfaces allow the Enterprise Operator to delete a client application from the Framework. It is not explicitly stated in the specification what the affect of deleting a client application that currently has an access session with the Framework is. If deleteClientApp deletes a client application, then it must, by necessity, end that application's access session and terminate any service instances it may have. It would be useful for the Enterprise Operator to be able to know before calling deleteClientApp whether the application has a session or not. It might be that if they know the client has an access session, then they might postpone the deletion until a later date. Since the intention is to keep the data types common between the ETSI/Parlay version of OSA and the 3GPP OSA specification, any change to the datatypes in the ETSI/Parlay specification must also be reflected in the 3GPP specification.
Summary of change:	⌘	We propose to a) Add a field to the client app description returned in describeClientApp indicating whether the application has an active session or not; and b) Add a field to the contract/profile description returned by describeServiceContract/Profile so that the enterprise operator knows when a contract/profile is being used and can choose whether to do the delete or not.
Consequences if not approved:	⌘	This contribution was accepted for the ETSI/Parlay specifications. If it is not approved for the 3GPP specification, then a misalignment will occur. Any misalignment between the ETSI/Parlay and 3GPP specifications will lead to incompatibilities between equipment developed from the ETSI/Parlay specifications and equipment developed from the 3GPP specifications. It will force vendors to develop different versions of their OSA products, one for operators requiring support of 3GPP specifications, and one for all other operators. This will increase development costs unnecessarily, increase the cost of deploying OSA, and fragment the single developer community which has formed

	around OSA

Clauses affected:	⌘	10.5.22, 10.5.24, 10.5.32			
		Y	**N**		
Other specs affected:	⌘		**X**	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	
Other comments:	⌘	This CR has Rel-6 Mirror CR to 29.198-03 in N5-040057.			

10.5 Service Subscription Data Definitions

10.5.22 TpServiceContractDescription

This data type is a `Sequence of Data Elements` which describes a service contract. This contract should conform to a previously negotiated high-level agreement (regarding OSA services, their usage and the price, etc.), if any, between the enterprise operator and the framework operator. It is a structured data type which consists of:

Sequence Element Name	Sequence Element Type
`ServiceRequestor`	TpServiceRequestor
`BillingContact`	TpBillingContact
`ServiceStartDate`	TpServiceStartDate
`ServiceEndDate`	TpServiceEndDate
`ServiceTypeName`	TpServiceTypeName
`ServiceID`	TpServiceID
`ServiceSubscriptionProperties`	TpServiceSubscriptionProperties
`InUse`	TpBoolean (See note 1)

Note 1: The InUse flag indicates if the contract, or one of its associated profiles, is currently in use by a service instance and will be returned in describeServiceContract(). This flag will be ignored if it is passed in to createServiceContract().

10.5.24 TpClientAppDescription

This data type is a `Sequence of Data Elements` which describes an enterprise client application. It is a structured data type, consisting of a unique "client application ID", password and a list of "client application properties:

Sequence Element Name	Sequence Element Type
`ClientAppID`	TpClientAppID
`ClientAppProperties`	TpClientAppProperties
`hasAccessSession`	TpBoolean (See note 1)
`hasServiceInstances`	TpBoolean(See note 2)

Note 1: The hasAccessSession flag indicates if the client application currently has an access session active with the framework and will be returned in describeClientApp(). This flag will be ignored if it is passed in to createClientApp().

Note 2: The hasServiceInstances flag indicates if the client application currently has service instances active and will be returned in describeClientApp(). This flag will be ignored if it is passed in to createClientApp(). This flag must be false if hasAccessSession is false.

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`	TpServiceContractID
`ServiceStartDate`	TpServiceStartDate
`ServiceEndDate`	TpServiceEndDate
`ServiceTypeName`	TpServiceTypeName
`ServiceSubscriptionProperties`	TpServiceSubscriptionProperties
`InUse`	TpBoolean (See note 1)

Note 1: The InUse flag indicates if the profile is currently in use by a service instance and will be returned in describeServiceProfile(). This flag will be ignored if it is passed in to createServiceProfile().

CR-Form-v7

CHANGE REQUEST

⌘ 29.198-03 CR 103 ⌘rev - ⌘ Current version: 6.0.1 ⌘

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

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network X

Title:	⌘	Add ability to identify when a client app/service contract/service profile is being used - Align between ETSI/Parlay and 3GPP
Source:	⌘	CN5 Parlay Gareth Carroll (Open API Solutions)
Work item code:	⌘	OSA2
Date:	⌘	09/02/2004
Category:	⌘	**A**
Release:	⌘	*REL-6*

Use one 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 one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change:	⌘	In the ETSI/Parlay version of OSA, the Enterprise Operator interfaces allow the Enterprise Operator to delete a client application from the Framework. It is not explicitly stated in the specification what the affect of deleting a client application that currently has an access session with the Framework is. If deleteClientApp deletes a client application, then it must, by necessity, end that application's access session and terminate any service instances it may have. It would be useful for the Enterprise Operator to be able to know before calling deleteClientApp whether the application has a session or not. It might be that if they know the client has an access session, then they might postpone the deletion until a later date. Since the intention is to keep the data types common between the ETSI/Parlay version of OSA and the 3GPP OSA specification, any change to the datatypes in the ETSI/Parlay specification must also be reflected in the 3GPP specification.
Summary of change:	⌘	We propose to a) Add a field to the client app description returned in describeClientApp indicating whether the application has an active session or not; and b) Add a field to the contract/profile description returned by describeServiceContract/Profile so that the enterprise operator knows when a contract/profile is being used and can choose whether to do the delete or not.
Consequences if not approved:	⌘	This contribution was accepted for the ETSI/Parlay specifications. If it is not approved for the 3GPP specification, then a misalignment will occur. Any misalignment between the ETSI/Parlay and 3GPP specifications will lead to incompatibilities between equipment developed from the ETSI/Parlay specifications and equipment developed from the 3GPP specifications. It will force vendors to develop different versions of their OSA products, one for operators requiring support of 3GPP specifications, and one for all other operators. This will increase development costs unnecessarily, increase the cost of deploying OSA, and fragment the single developer community which has formed

	around OSA

Clauses affected:	⌘	10.5.22, 10.5.24, 10.5.32			
		Y	**N**		
Other specs affected:	⌘		**X**	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	
Other comments:	⌘	This is the Rel-6 Mirror CR to Rel-5 CR to 29.198-03 in N5-040056.			

10.5 Service Subscription Data Definitions

10.5.22 TpServiceContractDescription

This data type is a Sequence of Data Elements which describes a service contract. This contract should conform to a previously negotiated high-level agreement (regarding OSA services, their usage and the price, etc.), if any, between the enterprise operator and the framework operator. It is a structured data type which consists of:

Sequence Element Name	Sequence Element Type
ServiceRequestor	TpServiceRequestor
BillingContact	TpBillingContact
ServiceStartDate	TpServiceStartDate
ServiceEndDate	TpServiceEndDate
ServiceTypeName	TpServiceTypeName
ServiceID	TpServiceID
ServiceSubscriptionProperties	TpServiceSubscriptionProperties
InUse	TpBoolean (See note 1)

Note 1: The InUse flag indicates if the contract, or one of its associated profiles, is currently in use by a service instance and will be returned in describeServiceContract(). This flag will be ignored if it is passed in to createServiceContract().

10.5.24 TpClientAppDescription

This data type is a Sequence of Data Elements which describes an enterprise client application. It is a structured data type, consisting of a unique "client application ID", password and a list of client application properties.

Sequence Element Name	Sequence Element Type
ClientAppID	TpClientAppID
ClientAppProperties	TpClientAppProperties
hasAccessSession	TpBoolean (See note 1)
hasServiceInstances	TpBoolean(See note 2)

Note 1: The hasAccessSession flag indicates if the client application currently has an access session active with the framework and will be returned in describeClientApp(). This flag will be ignored if it is passed in to createClientApp().

Note 2: The hasServiceInstances flag indicates if the client application currently has service instances active and will be returned in describeClientApp(). This flag will be ignored if it is passed in to createClientApp(). This flag must be false if hasAccessSession is false.

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	TpServiceContractID
ServiceStartDate	TpServiceStartDate
ServiceEndDate	TpServiceEndDate
ServiceTypeName	TpServiceTypeName
ServiceSubscriptionProperties	TpServiceSubscriptionProperties
InUse	TpBoolean (See note 1)

Note 1: The InUse flag indicates if the profile is currently in use by a service instance and will be returned in describeServiceProfile(). This flag will be ignored if it is passed in to createServiceProfile().

joint-API-group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) **N5-040060**
Meeting #26, Atlanta, GA, USA, 16-20 February 2004

CR-Form-v7

CHANGE REQUEST

⌘ **29.198-03** **CR** **107** ⌘ **rev** **-** ⌘ Current version: **5.6.0** ⌘

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

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network **X**

Title:	⌘	Introduce a ServiceID field to TpServiceProfileDescription
Source:	⌘	CN5 Parlay Gareth Carroll (Open API Solutions)
Work item code:	⌘	OSA2 ***Date:*** ⌘ 09/02/2004
Category:	⌘	**F** ***Release:*** ⌘ *REL-5*

Use one 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 one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change: ⌘ In the ETSI/Parlay version of OSA, if a service contract is for a service type, then it may be desirable and should certainly be possible to create a service profile for that contract that provides restrictions for the use of a specific service of that type. Here a service contract would exist for the service type and the service profile would specify the restrictions applicable for a specific serviceID. In this case, some validation would still have to occur (to check that the service ID, if present, is for a service of the type specified in the contract, or, if the contract is for a service ID, that this value is either the same as the one specified in the contract or ignored).

Since the intention is to keep the data types common between the ETSI/Parlay version of OSA and the 3GPP OSA specification, any change to the datatypes in the ETSI/Parlay specification must also be reflected in the 3GPP specification.

Summary of change: ⌘ We propose to add a ServiceID field to the TpServiceProfileDescription. If this field is added at the end of the structure then it does not impact backwards compatibility. This change is at Level 0 in the Backwards Compatibility white paper – "Already deployed applications are not affected at all" as clients using older IDL will not experience a problem with handling this extended structure.

Consequences if not approved: ⌘ This contribution was accepted for the ETSI/Parlay specifications. If it is not approved for the 3GPP specification, then a misalignment will occur.
Any misalignment between the ETSI/Parlay and 3GPP specifications will lead to incompatibilities between equipment developed from the ETSI/Parlay specifications and equipment developed from the 3GPP specifications.
It will force vendors to develop different versions of their OSA products, one for operators requiring support of 3GPP specifications, and one for all other operators.
This will increase development costs unnecessarily, increase the cost of deploying OSA, and fragment the single developer community which has formed around OSA.

Clauses affected:	⌘	10.5.32			
		Y	**N**		
Other specs affected:	⌘		**X**	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	
Other comments:	⌘	This CR has Rel-6 Mirror CR to 29.198-03 in N5-040061.			

10.5 Service Subscription Data Definitions

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`	TpServiceContractID
`ServiceStartDate`	TpServiceStartDate
`ServiceEndDate`	TpServiceEndDate
`ServiceTypeName`	TpServiceTypeName
`ServiceSubscriptionProperties`	TpServiceSubscriptionProperties
`ServiceID`	TpServiceID (See Note)

Note: The ServiceID field is used to restrict a service type-based service contract to a specific service. When the TpServiceProfileDescription is passed to the Framework by an enterprise operator, the Framework should ensure that the ServiceID field, if not empty, contains a service which is of the service type specified in the service contract. If the corresponding contract is for a service ID then the Framework should ignore this field.

When a TpServiceProfileDescription is returned to the enterprise operator, the contents of this field will depend on the associated service contract. If the contract is for a service ID, then this field should be populated with the correct value. If the contract is for a service type, and the profile is restricted to a specific service ID then this field should be be populated with the correct value. Otherwise, it should contain an empty string.

CR-Form-v7

CHANGE REQUEST

⌘ **29.198-03** **CR** **108** ⌘ **rev** **-** ⌘ Current version: **6.0.1** ⌘

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

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network **X**

Title:	⌘	Introduce a ServiceID field to TpServiceProfileDescription
Source:	⌘	CN5 Parlay Gareth Carroll (Open API Solutions)
Work item code:	⌘	OSA2 ***Date:*** ⌘ 09/02/2004
Category:	⌘	**A** ***Release:*** ⌘ *REL-6*

Use one 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 one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change: ⌘ In the ETSI/Parlay version of OSA, if a service contract is for a service type, then it may be desirable and should certainly be possible to create a service profile for that contract that provides restrictions for the use of a specific service of that type. Here a service contract would exist for the service type and the service profile would specify the restrictions applicable for a specific serviceID. In this case, some validation would still have to occur (to check that the service ID, if present, is for a service of the type specified in the contract, or, if the contract is for a service ID, that this value is either the same as the one specified in the contract or ignored).

Since the intention is to keep the data types common between the ETSI/Parlay version of OSA and the 3GPP OSA specification, any change to the datatypes in the ETSI/Parlay specification must also be reflected in the 3GPP specification.

Summary of change: ⌘ We propose to add a ServiceID field to the TpServiceProfileDescription. If this field is added at the end of the structure then it does not impact backwards compatibility. This change is at Level 0 in the Backwards Compatibility white paper – "Already deployed applications are not affected at all" as clients using older IDL will not experience a problem with handling this extended structure.

Consequences if not approved: ⌘ This contribution was accepted for the ETSI/Parlay specifications. If it is not approved for the 3GPP specification, then a misalignment will occur.
Any misalignment between the ETSI/Parlay and 3GPP specifications will lead to incompatibilities between equipment developed from the ETSI/Parlay specifications and equipment developed from the 3GPP specifications.
It will force vendors to develop different versions of their OSA products, one for operators requiring support of 3GPP specifications, and one for all other operators.
This will increase development costs unnecessarily, increase the cost of deploying OSA, and fragment the single developer community which has formed around OSA.

Clauses affected:	⌘	10.5.32			
		Y	**N**		
Other specs affected:	⌘		**X**	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	
Other comments:	⌘	This is the Rel-6 Mirror CR to Rel-5 CR 29.198-03 in N5-040060.			

10.5 Service Subscription Data Definitions

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	TpServiceContractID
ServiceStartDate	TpServiceStartDate
ServiceEndDate	TpServiceEndDate
ServiceTypeName	TpServiceTypeName
ServiceSubscriptionProperties	TpServiceSubscriptionProperties
ServiceID	TpServiceID (See Note)

Note: The ServiceID field is used to restrict a service type-based service contract to a specific service. When the TpServiceProfileDescription is passed to the Framework by an enterprise operator, the Framework should ensure that the ServiceID field, if not empty, contains a service which is of the service type specified in the service contract. If the corresponding contract is for a service ID then the Framework should ignore this field.

When a TpServiceProfileDescription is returned to the enterprise operator, the contents of this field will depend on the associated service contract. If the contract is for a service ID, then this field should be populated with the correct value. If the contract is for a service type, and the profile is restricted to a specific service ID then this field should be be populated with the correct value. Otherwise, it should contain an empty string.

joint-API-group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) ***N5-040349***
Meeting #27, Miami, FL, USA, 10-14 May 2004

CR-Form-v7

CHANGE REQUEST

⌘ **29.198-03** **CR** **114** ⌘ **rev** **-** ⌘ Current version: **5.6.0** ⌘

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

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network **X**

Title:	⌘	Correct description of availStatusReason codes
Source:	⌘	CN5 Lucent Technologies
Work item code:	⌘	OSA2
Date:	⌘	18/05/2004
Category:	⌘	**F**
Release:	⌘	Rel-5

Use one 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 one of the following releases:
2 (GSM Phase 2)
R96 (Release 1996)
R97 (Release 1997)
R98 (Release 1998)
R99 (Release 1999)
Rel-4 (Release 4)
Rel-5 (Release 5)
Rel-6 (Release 6)

Reason for change:	⌘	The descriptions of the TpSvcAvailStatusReason and TpAppAvailStatusReason codes are vague and can cause ambiguity across implementations resulting in inoperability between SCFs and clients.
Summary of change:	⌘	This CR addreses the changes needed to clarify the intent of the reason codes: 1) The reason codes apply to a service instance (not an SCF). 2) Mention that the 'expected' recovery time could be defined within the SLA so the client doesn't wait indefinitely for the service instance to become available. 3) Explicitly state which reason codes are temporary and which are permanent.
Consequences if not approved:	⌘	There would remain confusion as to what action the Framework and Client should take. Without explicitly stating the expected behavior, interoperability between SCFs and Applications will suffer.

Clauses affected:	⌘	10.4.22, 10.4.23

Other specs affected: ⌘	Y	N	
		X	Other core specifications
		X	Test specifications
		X	O&M Specifications

⌘

Other comments:	⌘	Mirror CR for Rel-6 in N5-040350

Change in Clause 10.4.22

10.4.22 TpSvcAvailStatusReason

Defines the reason detailing the change in status of Service Instance availability~~Defines the reason why a SCF is unavailable~~.

Name	Value	Description
SVC_UNAVAILABLE_UNDEFINED	0	Undefined. A permanent failure.[1]
SVC_UNAVAILABLE_LOCAL_FAILURE	1	The Local API software or hardware has failed. A permanent failure.[1]~~Normally take longer time to correct~~
SVC_UNAVAILABLE_GATEWAY_FAILURE	2	The gateway API software or hardware has failed . A permanent failure.[1]~~Normally take longer time to correct~~
SVC_UNAVAILABLE_OVERLOADED	3	The Service Instance~~CF~~ is fully overloaded. A~~Normally a~~ temporary problem.[2]
SVC_UNAVAILABLE_CLOSED	4	The Service Instance~~CF~~ has closed itself (e.g. to protect from fraud or malicious attack). A permanent failure.[1] ~~Normally take longer time to correct~~
SVC_UNAVAILABLE_NO_RESPONSE	5	The Framework has detected that ~~the service~~a Service Instance has failed: e.g. non-response from an activity test, failure to return heartbeats. A permanent failure.[1]
SVC_UNAVAILABLE_SW_UPGRADE	6	The Service Instance is unavailable due to ~~SW~~ software upgrade or other similar maintenance. A permanent failure.[1] ~~Normally a temporary problem~~
SVC_AVAILABLE	7	The Service has become available again.

[1] - The client application must act to reset its use of the specified service instance (using the normal mechanisms, such as the discovery and authentication interfaces, to stop use of this service instance and begin use of a different service instance).

[2] - The "expected" recovery time could be defined within the SLA.

End of change in Clause 10.4.22

Change in Clause 10.4.23

10.4.23 TpAppAvailStatusReason

Defines the reason detailing the change in status of Application availability~~Defines the reason why the Application is unavailable~~.

Name	Value	Description
APP_UNAVAILABLE_UNDEFINED	0	Undefined. A permanent failure. [1]
APP_UNAVAILABLE_LOCAL_FAILURE	1	A local failure in the Application has been detected. A permanent failure. [1] ~~Normally take longer time to correct~~
APP_UNAVAILABLE_REMOTE_FAILURE	2	A remote failure to the application has been detected, e.g. a database is not working. A permanent failure. [1] ~~Normally take longer time to correct~~
APP_UNAVAILABLE_OVERLOADED	3	The Application is fully overloaded. A temporary problem. [2] ~~Often a temporary problem~~
APP_UNAVAILABLE_CLOSED	4	The Application has closed itself (e.g. to protect from fraud or malicious attack) . A permanent failure. [1] ~~Normally take longer time to correct~~
APP_UNAVAILABLE_NO_RESPONSE	5	The Framework has detected that the application has failed: e.g. non-response from an activity test, failure to return heartbeats. A permanent failure. [1]
APP_UNAVAILABLE_SW_UPGRADE	6	The Application is unavailable due to SW upgrade or other similar maintenance. A permanent failure. [1] ~~Often a temporary problem~~
APP_AVAILABLE	7	The Application has become available.

[1] – The client application is unable (or does not wish) to continue using the service instance.

[2] - The "expected" recovery time could be defined within the SLA.

End of change in Clause 10.4.23

Annex D (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Mar 2001	CN_11	NP-010134	047	--	CR 29.198: for moving TS 29.198 from R99 to Rel 4 (N5-010158)	3.2.0	4.0.0
Jun 2001	CN_12	NP-010330	001	--	Corrections to OSA API Rel4	4.0.0	4.0.1
Sep 2001	CN_13	NP-010466	002	--	Changing references to JAIN	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	003	--	Update to the definitions of method svcUnavailableInd	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	004	--	Only one subject per method invocation for fault and load management	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	005	--	Fault management is missing some *Err methods	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	006	--	Method balance on Fault management interfaces	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	007	--	Change "TpString" into "TpOctetSets" in authentication and access	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	008	--	Replacement of register/unregisterLoadController	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	009	--	Redundant Framework Heartbeat Mechanism	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	010	--	Add a releaseInterface() method to IpAccess	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	011	--	Removal of serviceID from queryAppLoadReq()	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	012	--	Addition of listInterfaces() method	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	013	--	Introduction and use of new Service Instance ID	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	014	--	P_UNAUTHORISED_PARAMETER_VALUE thrown if non-accessible serviceID is provided	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	015	--	Introduction of Service Instance Lifecycle Management	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	016	--	Add support for multi-vendorship	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	017	--	Removal of P_SERVICE_ACCESS_TYPE	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	018	--	Confusing meaning of prescribedMethod	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	019	--	A client should only have one instance of a given service	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	020	--	Some methods on the IpApp interfaces should throw exceptions	4.1.0	4.2.0
Dec 2001	CN_14	NP-010596	021	--	Replace Out Parameters with Return Types	4.2.0	4.3.0
Dec 2001	CN_14	NP-010596	022	--	Correctionto Framework (FW)	4.2.0	4.3.0
Mar 2002	CN_15	NP-020105	023	--	Add P_INVALID_INTERFACE_TYPE exception to IpService.setCallback() and IpService.setCallbackWithSessionID()	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	024	--	Replace erroneous mention of P_OSA_ACCESS by the correct value P_OSA_AUTHENTICATION	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	025	--	Add missing inheritance in service agreement management interfaces	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	026	--	Include Operation Set as part of General Service Properties	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	027	--	Improved description of activityTestReq with respect to ServiceInstanceID	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	028	--	OSA Framework - Generate statistics records on behalf of another entity using genFaultStatsRecordReq	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	029	--	Update the interface names for alignment between 3GPP and ETSI/Parlay	4.3.0	4.4.0
Jun 2002	CN_16	NP-020179	030	--	Solving the problem in the OSA Framework with method appUnavailableInd() in a scenario with multiple service sessions per access session	4.4.0	4.5.0
Jun 2002	CN_16	NP-020179	031	--	Adding missing mandatory method (authenticationSucceeded) to sequence flow	4.4.0	4.5.0
Jun 2002	CN_16	NP-020186	032	--	Remove redundant data type definition TpServiceSpecString	4.5.0	5.0.0
Jun 2002	CN_16	NP-020181	033	--	Addition of support for Java API technology realisation	4.5.0	5.0.0
Jun 2002	CN_16	NP-020182	035	--	Addition of support for WSDL realisation	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	036	--	Clarify semantics of service properties of type BOOLEAN_SET	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	037	--	Addition of version management support to the Framework (29.198-03) in run-time	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	038	--	Enhancements on subscription management error information	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	039	--	Delete conflicting description of P_APPLICATION_NOT_ACTIVATED	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	040	--	Note added for P_SERVICE_INSTANCE Choice Element Name	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	041	--	Correcting the method descriptions for abortAuthentication and for initiateAuthentication	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	042	--	Correcting the description of heartbeat failure	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	043	--	Correcting erroneous FW<->Service instance sequence diagrams	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	044	--	Correcting the scope of TpFwID, which currently is giving it false limitations	4.5.0	5.0.0
Sep 2002	CN_17	NP-020428	046		Correction to description of TpServicePropertyTypeName	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	047		Remove undefined exception in registerService	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	048		Remove ServiceIDs from IpFwFaultManager.genFaultStatsRecordReq()	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	049		Correct appUnavailableInd and related methods	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	050		Remove unusable exception from IpFaultManager.appActivityTestRes()	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	051		Clarify the sequence of events in signing the service agreement	5.0.0	5.1.0

Sep 2002	CN_17	NP-020428	052		Correct use of electronic signatures	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	053		Addition of Sequence Diagrams for terminateAccess	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	054		Add indication what part of service agreement must be signed	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	055		Add text to clarify requirements on support of methods	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	056		Introduce types and modes for generic properties	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	057		Correction on use of NULL in Framework API	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	058		Add Negotiation of Authentication Mechanism for OSA level Authentication	5.0.0	5.1.0
Sep 2002	CN_17	NP-020395	058		Add text to clarify relationship between 3GPP and ETSI/Parlay OSA specifications	5.0.0	5.1.0
Mar 2003	CN_19	NP-030019	063	-	Correction to Initial Access Sequence Diagram	5.1.0	5.2.0
Mar 2003	CN_19	NP-030019	065	-	Enable creation/destruction of load level notifications at the request of Framework	5.1.0	5.2.0
Mar 2003	CN_19	NP-030019	067	-	Correction of Sequence for Framework – Service load management	5.1.0	5.2.0
Mar 2003	CN_19	NP-030019	074	-	Add Initial Load Notification report for Framework Integrity Management Load Notification model	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	068	--	Correction to Application's requirements for supporting methods	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	069	--	Correction of status of methods to interfaces in clause 7.3	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	070	--	Correction of status of methods to interfaces in clause 8.3	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	071	--	Correction of status of methods to interfaces in clause 6.3	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	075	--	Adding the appAvailStatusInd() and svcAvailStatusInd() methods	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	076	--	Remove race condition in signServiceAgreement	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	077	--	Change reference to deprecated method "authenticate" in TpAuthMechanism to "challenge"	5.1.0	5.2.0
Jun 2003	CN_20	NP-030237	079	--	Correction to TpEncryptionCapability to correct support for Triple-DES	5.2.0	5.3.0
Jun 2003	CN_20	NP-030237	081	--	Correction of the Framework Service Instance Lifecycle Manager Sequence Diagram	5.2.0	5.3.0
Jun 2003	CN_20	NP-030237	083	--	Correction of the use of TpDomainID in Framework initiateAuthentication method	5.2.0	5.3.0
Sep 2003	CN_21	NP-030352	085	--	Correction to Java Realisation Annex	5.3.0	5.4.0
Dec 2003	CN_22	NP-030549	086	--	Correction of the sequence diagram for Fault Management	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	087	--	Correction of State Transition Diagram for IpAccess	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	088	--	Correction of Correlation Behaviour in Load Management	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	089	--	Correction of Correlation Behaviour in Fault Management	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	090	--	Correction and Clarification of Framework Access Session Behaviour	5.4.0	5.5.0
Apr 2004	CN_23bis	NP-040155	101	--	Correct Java Code to conform with Java Rulebook in TS 29.198-01 and to remove errors	5.5.0	5.6.0

joint-API-group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) ***N5-040350***
Meeting #27, Miami, FL, USA, 10-14 May 2004

CR-Form-v7

CHANGE REQUEST

⌘ **29.198-03** **CR** **115** ⌘ **rev** **-** ⌘ Current version: **6.0.1** ⌘

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

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network **X**

Title:	⌘	Correct description of availStatusReason codes
Source:	⌘	CN5 Lucent Technologies
Work item code:	⌘	OSA2
Date:	⌘	18/05/2004
Category:	⌘	**A**
Release:	⌘	Rel-6

Use one 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 one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change:	⌘	The descriptions of the TpSvcAvailStatusReason, TpAppAvailStatusReason, and TpFwAvailStatusReason codes are vague and can cause ambiguity across implementations resulting in inoperability between SCFs, clients, and the framework.
Summary of change:	⌘	This CR addreses the changes needed to clarify the intent of the reason codes: 1) The reason codes apply to a service instance (not an SCF). 2) Mention that the 'expected' recovery time could be defined within the SLA so the client doesn't wait indefinitely for the service instance to become available. 3) Explicitly state which reason codes are temporary and which are permanent.
Consequences if not approved:	⌘	There would remain confusion as to what action the Framework and Client should take. Without explicitly stating the expected behavior, interoperability between SCFs, applications, and the framework will suffer.

Clauses affected:	⌘	10.4.22, 10.4.23, 10.4.27

Other specs affected: ⌘	Y	N		⌘
		X	Other core specifications	
		X	Test specifications	
		X	O&M Specifications	

Other comments:	⌘	Mirror CR for Rel-5 in N5-040349

Change in Clause 10.4.22

10.4.22 TpSvcAvailStatusReason

Defines the reason detailing the change in status of Service Instnace availability.

Name	Value	Description
SVC_UNAVAILABLE_UNDEFINED	0	Undefined. A permanent failure.[1]
SVC_UNAVAILABLE_LOCAL_FAILURE	1	The Local API software or hardware has failed. . A permanent failure.[1] ~~Normally take longer time to correct~~
SVC_UNAVAILABLE_GATEWAY_FAILURE	2	The gateway API software or hardware has failed . A permanent failure.[1] ~~Normally take longer time to correct~~
SVC_UNAVAILABLE_OVERLOADED	3	The ~~SCF~~ Service Instance is fully overloaded ~~Normally A~~a temporary problem[2].
SVC_UNAVAILABLE_CLOSED	4	The ~~SCF~~ Service Instance has closed itself (e.g. to protect from fraud or malicious attack) A permanent failure.[1] ~~Normally take longer time to correct~~
SVC_UNAVAILABLE_NO_RESPONSE	5	The Framework has detected that the ~~service~~ Service Instance has failed: e.g. non-response from an activity test, failure to return heartbeats A permanent failure.[1]
SVC_UNAVAILABLE_SW_UPGRADE	6	The Service Instance is unavailable due to ~~SW~~ software upgrade or other similar maintenance A permanent failure.[1] ~~Normally a temporary problem~~
SVC_AVAILABLE	7	The Service has become available again

[1] - The client application must act to reset its use of the specified service instance (using the normal mechanisms, such as the discovery and authentication interfaces, to stop use of this service instance and begin use of a different service instance).

[2] - The "expected" recovery time could be defined within the SLA.

End of change in Clause 10.4.22

Change in Clause 10.4.23

10.4.23 TpAppAvailStatusReason

Defines the reason detailing the change in status of Application availability.

Name	Value	Description
APP_UNAVAILABLE_UNDEFINED	0	Undefined. A permanent failure.[1]
APP_UNAVAILABLE_LOCAL_FAILURE	1	A local failure in the Application has been detected. A permanent failure.[1] ~~Normally take longer time to correct~~
APP_UNAVAILABLE_REMOTE_FAILURE	2	A remote failure to the application has been detected, e.g. a database is not working. A permanent failure. [1] ~~Normally take longer time to correct~~
APP_UNAVAILABLE_OVERLOADED	3	The Application is fully overloaded. A temporary problem.[2] ~~Often a temporary problem~~
APP_UNAVAILABLE_CLOSED	4	The Application has closed itself (e.g. to protect from fraud or malicious attack) . A permanent failure. [1] ~~Normally take longer time to correct~~
APP_UNAVAILABLE_NO_RESPONSE	5	The Framework has detected that the application has failed: e.g. non-response from an activity test, failure to return heartbeats. A permanent failure. [1]
APP_UNAVAILABLE_SW_UPGRADE	6	The Application is unavailable due to SW upgrade or other similar maintenance. A permanent failure. [1] ~~Often a temporary problem~~
APP_AVAILABLE	7	The Application has become available

[1] – The client application is unable (or does not wish) to continue using the service instance.

[2] - The "expected" recovery time could be defined within the SLA.

End of change in Clause 10.4.23

Change in Clause 10.4.27

10.4.27 TpFwAvailStatusReason

Defines the reason detailing the change in status of Framework availability.

Name	Value	Description
FRAMEWORK_UNAVAILABLE_UNDEFINED	0	Undefined. A permanent failure.[1]
FRAMEWORK_UNAVAILABLE_LOCAL_FAILURE	1	A local failure in the Framework has been detected. A permanent failure.[1] ~~Normally take longer time to correct~~
FRAMEWORK_UNAVAILABLE_REMOTE_FAILURE	2	A remote failure to the Framework has been detected, e.g. a database is not working. A permanent failure.[1] ~~Normally take longer time to correct~~
FRAMEWORK_UNAVAILABLE_OVERLOADED	3	The Framework is fully overloaded. A temporary problem.[2] ~~Often a temporary problem~~
FRAMEWORK_UNAVAILABLE_CLOSED	4	The Framework has closed itself (e.g. to protect from fraud or malicious attack) . A permanent failure.[1] ~~Normally take longer time to correct~~
FRAMEWORK_UNAVAILABLE_PROTOCOL_FAILURE	5	The Framework has detected that the protocol used between client and framework has failed. A permanent failure.[1]
FRAMEWORK_UNAVAILABLE_SW_UPGRADE	6	The Framework is unavailable due to SW upgrade or other similar maintenance. A permanent failure.[1] ~~Often a temporary problem~~
FRAMEWORK_AVAILABLE	7	The Framework has become available.

[1] - The Framework is unable (or does not wish) to continue using the client or service instance.

[2] - The 'expected' recovery time could be part of the Framework's local policies.

End of change in Clause 10.4.27

Annex E (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Mar 2001	CN_11	NP-010134	047	--	CR 29.198: for moving TS 29.198 from R99 to Rel 4 (N5-010158)	3.2.0	4.0.0
Jun 2001	CN_12	NP-010330	001	--	Corrections to OSA API Rel4	4.0.0	4.0.1
Sep 2001	CN_13	NP-010466	002	--	Changing references to JAIN	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	003	--	Update to the definitions of method svcUnavailableInd	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	004	--	Only one subject per method invocation for fault and load management	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	005	--	Fault management is missing some *Err methods	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	006	--	Method balance on Fault management interfaces	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	007	--	Change "TpString" into "TpOctetSets" in authentication and access	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	008	--	Replacement of register/unregisterLoadController	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	009	--	Redundant Framework Heartbeat Mechanism	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	010	--	Add a releaseInterface() method to IpAccess	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	011	--	Removal of serviceID from queryAppLoadReq()	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	012	--	Addition of listInterfaces() method	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	013	--	Introduction and use of new Service Instance ID	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	014	--	P_UNAUTHORISED_PARAMETER_VALUE thrown if non-accessible serviceID is provided	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	015	--	Introduction of Service Instance Lifecycle Management	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	016	--	Add support for multi-vendorship	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	017	--	Removal of P_SERVICE_ACCESS_TYPE	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	018	--	Confusing meaning of prescribedMethod	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	019	--	A client should only have one instance of a given service	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	020	--	Some methods on the IpApp interfaces should throw exceptions	4.1.0	4.2.0
Dec 2001	CN_14	NP-010596	021	--	Replace Out Parameters with Return Types	4.2.0	4.3.0
Dec 2001	CN_14	NP-010596	022	--	Correctionto Framework (FW)	4.2.0	4.3.0
Mar 2002	CN_15	NP-020105	023	--	Add P_INVALID_INTERFACE_TYPE exception to IpService.setCallback() and IpService.setCallbackWithSessionID()	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	024	--	Replace erroneous mention of P_OSA_ACCESS by the correct value P_OSA_AUTHENTICATION	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	025	--	Add missing inheritance in service agreement management interfaces	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	026	--	Include Operation Set as part of General Service Properties	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	027	--	Improved description of activityTestReq with respect to ServiceInstanceID	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	028	--	OSA Framework - Generate statistics records on behalf of another entity using genFaultStatsRecordReq	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	029	--	Update the interface names for alignment between 3GPP and ETSI/Parlay	4.3.0	4.4.0
Jun 2002	CN_16	NP-020179	030	--	Solving the problem in the OSA Framework with method appUnavailableInd() in a scenario with multiple service sessions per access session	4.4.0	4.5.0
Jun 2002	CN_16	NP-020179	031	--	Adding missing mandatory method (authenticationSucceeded) to sequence flow	4.4.0	4.5.0
Jun 2002	CN_16	NP-020186	032	--	Remove redundant data type definition TpServiceSpecString	4.5.0	5.0.0
Jun 2002	CN_16	NP-020181	033	--	Addition of support for Java API technology realisation	4.5.0	5.0.0
Jun 2002	CN_16	NP-020182	035	--	Addition of support for WSDL realisation	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	036	--	Clarify semantics of service properties of type BOOLEAN_SET	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	037	--	Addition of version management support to the Framework (29.198-03) in run-time	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	038	--	Enhancements on subscription management error information	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	039	--	Delete conflicting description of P_APPLICATION_NOT_ACTIVATED	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	040	--	Note added for P_SERVICE_INSTANCE Choice Element Name	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	041	--	Correcting the method descriptions for abortAuthentication and for initiateAuthentication	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	042	--	Correcting the description of heartbeat failure	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	043	--	Correcting erroneous FW<->Service instance sequence diagrams	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	044	--	Correcting the scope of TpFwID, which currently is giving it false limitations	4.5.0	5.0.0
Sep 2002	CN_17	NP-020428	046		Correction to description of TpServicePropertyTypeName	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	047		Remove undefined exception in registerService	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	048		Remove ServiceIDs from IpFwFaultManager.genFaultStatsRecordReq()	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	049		Correct appUnavailableInd and related methods	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	050		Remove unusable exception from IpFaultManager.appActivityTestRes()	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	051		Clarify the sequence of events in signing the service agreement	5.0.0	5.1.0

Sep 2002	CN_17	NP-020428	052		Correct use of electronic signatures	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	053		Addition of Sequence Diagrams for terminateAccess	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	054		Add indication what part of service agreement must be signed	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	055		Add text to clarify requirements on support of methods	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	056		Introduce types and modes for generic properties	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	057		Correction on use of NULL in Framework API	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	058		Add Negotiation of Authentication Mechanism for OSA level Authentication	5.0.0	5.1.0
Sep 2002	CN_17	NP-020395	058		Add text to clarify relationship between 3GPP and ETSI/Parlay OSA specifications	5.0.0	5.1.0
Mar 2003	CN_19	NP-030019	063	-	Correction to Initial Access Sequence Diagram	5.1.0	5.2.0
Mar 2003	CN_19	NP-030019	065	-	Enable creation/destruction of load level notifications at the request of Framework	5.1.0	5.2.0
Mar 2003	CN_19	NP-030019	067	-	Correction of Sequence for Framework – Service load management	5.1.0	5.2.0
Mar 2003	CN_19	NP-030019	074	-	Add Initial Load Notification report for Framework Integrity Management Load Notification model	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	068	--	Correction to Application's requirements for supporting methods	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	069	--	Correction of status of methods to interfaces in clause 7.3	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	070	--	Correction of status of methods to interfaces in clause 8.3	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	071	--	Correction of status of methods to interfaces in clause 6.3	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	075	--	Adding the appAvailStatusInd() and svcAvailStatusInd() methods	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	076	--	Remove race condition in signServiceAgreement	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	077	--	Change reference to deprecated method "authenticate" in TpAuthMechanism to "challenge"	5.1.0	5.2.0
Jun 2003	CN_20	NP-030237	079	--	Correction to TpEncryptionCapability to correct support for Triple-DES	5.2.0	5.3.0
Jun 2003	CN_20	NP-030237	081	--	Correction of the Framework Service Instance Lifecycle Manager Sequence Diagram	5.2.0	5.3.0
Jun 2003	CN_20	NP-030237	083	--	Correction of the use of TpDomainID in Framework initiateAuthentication method	5.2.0	5.3.0
Sep 2003	CN_21	NP-030352	085	--	Correction to Java Realisation Annex	5.3.0	5.4.0
Dec 2003	CN_22	NP-030549	086	--	Correction of the sequence diagram for Fault Management	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	087	--	Correction of State Transition Diagram for IpAccess	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	088	--	Correction of Correlation Behaviour in Load Management	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	089	--	Correction of Correlation Behaviour in Fault Management	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	090	--	Correction and Clarification of Framework Access Session Behaviour	5.4.0	5.5.0
Dec 2003	CN_22	NP-030553	091	--	Add OSA API support for 3GPP2 networks	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	092	--	Add description for service super and sub types	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	093	--	Add support for registration of additional service property types and modes	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	094	--	Improve User Interaction message management functions	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	095	--	Add new values for TpServiceTypeName for Policy Management	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	096	--	Allow for applications to re-obtain the reference to the service manager	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	097	--	Add support in OSA to inform applications about new SCSs and their level of Backward compatibility – Align with SA1's 22.127	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	098	--	Add “Extended User Status” as service type name - Align with 29.198-06	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	099	--	Add P_USER_BINDING to TpServiceTypeName	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	100	--	Modify Framework Availability Indication in Fault Management	5.5.0	6.0.0
Feb 2004	--	--	--	--	Added Java code attachment 2919803J2EE.zip which was delivered late by outside developers. See Annex C.	6.0.0	6.0.1

joint-API-group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) ***N5-040351***
Meeting #27, Miami, FL, USA, 10-14 May 2004

CR-Form-v7

CHANGE REQUEST

⌘ **29.198-03** **CR** **116** ⌘ **rev** **-** ⌘ Current version: **5.6.0** ⌘

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

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network **X**

Title:	⌘	Correct description for the use of selectSigningAlgorithm
Source:	⌘	CN5 Lucent Technologies
Work item code:	⌘	OSA2 ***Date:*** ⌘ 18/05/2004
Category:	⌘	**F** ***Release:*** ⌘ Rel-5

Use one 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 one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change:	⌘	The described usage of the selectSigningAlgorithm() method is ambiguous and requires correction.
Summary of change:	⌘	Additional clarifying text has been added to correct the description of selectSigningAlgorithm() and to each of the methods that use a digital signature.
Consequences if not approved:	⌘	A client application could use an incorrect signing algorithm with some methods resulting in operational failures.

Clauses affected:	⌘	6.3.1.6.6, 6.3.1.6.7, 6.3.1.6.8, 7.3.2.1.2, 7.3.2.2.1, 7.3.2.2.2

Other specs affected: ⌘	Y	N		⌘
		X	Other core specifications	
		X	Test specifications	
		X	O&M Specifications	

Other comments: ⌘ Mirror CR for Rel-6 in N5-040352

Change in Clause 6.3.1.6.6

6.3.1.6.6 Method <<new>> selectSigningAlgorithm()

The client uses this method to inform the Framework of the different signing algorithms it supports for use in all cases where digital signatures are required. The Framework will select one of the suggested algorithms. This method shall be the first method invoked by the client on IpAccess. The algorithm chosen as a result of the response to this method remains valid for an instance of IpAccess and until this method is re-invoked by the client.

Subsequent invocations of selectSigningAlgorithm() may change the signing algorithm used during the access session. However, once signServiceAgreement() has been called on the client by the framework, the signing algorithm currently selected must be used for the client's invocation of signServiceAgreement() on the Framework as well as for subsequent calls to terminateServiceAgreement(). Other operations requiring digital signatures will use the latest algorithm specified by selectSigningAlgorithm().

If an algorithm that is acceptable to the framework within the capability of the client cannot be found, the framework throws the P_NO_ACCEPTABLE_SIGNING_ALGORITHM exception.

Returns: selectedAlgorithm. This is the signing algorithm chosen by the Framework. The chosen algorithm shall be taken from the list proposed by the Client.

Parameters

signingAlgorithmCaps : in TpSigningAlgorithmCapabilityList

The list of signing algorithms supported by the client.

Returns

TpSigningAlgorithm

Raises

TpCommonExceptions, P_ACCESS_DENIED, P_NO_ACCEPTABLE_SIGNING_ALGORITHM

End of change in Clause 6.3.1.6.6

Change in Clause 6.3.1.6.7

6.3.1.6.7 Method <<new>> terminateAccess()

The terminateAccess method is used by the client to request that its access session with the framework is ended. After it is invoked, the client will no longer be authenticated with the framework. The client will not be able to use the references to any of the framework interfaces gained during the access session. Any calls to these interfaces will fail. Also, all remaining service instances created by the framework either directly in this access session or on behalf of the client during this access session shall be terminated.

Parameters

terminationText : in TpString

This is the termination text describes the reason for the termination of the access session.

digitalSignature : in TpOctetSet

This contains a CMS (Cryptographic Message Syntax) object (as defined in RFC 2630) with content type Signed-data. The signature is calculated and created as per section 5 of RFC 2630 using the latest signing algorithm selected with selectSigningAlgorithm(). The content is made of the termination text. The "external signature" construct shall not be used (i.e. the eContent field in the EncapsulatedContentInfo field shall be present and contain the termination text string). The signing-time attribute, as defined in section 11.3 of RFC 2630, shall also be used to provide replay

prevention. The client uses this to confirm its identity to the framework. The framework can check that the terminationText has been signed by the client. If a match is made, the access session is terminated, otherwise the P_INVALID_SIGNATURE exception will be thrown.

Raises

TpCommonExceptions, P_INVALID_SIGNATURE

End of change in Clause 6.3.1.6.7

Change in Clause 6.3.1.6.8

6.3.1.6.8 Method <<new>> relinquishInterface()

The client uses this method to release an instance of a framework interface that was obtained during this access session.

Parameters

interfaceName : in TpInterfaceName

This is the name of the framework interface which is being released. If the interfaceName is invalid, the framework throws the P_INVALID_INTERFACE_NAME exception. If the interface has not been given to the client during this access session, then the P_TASK_REFUSED exception will be thrown.

terminationText : in TpString

This is the termination text describes the reason for the release of the interface. This text is required simply because the digitalSignature parameter requires a terminationText to sign.

digitalSignature : in TpOctetSet

This contains a CMS (Cryptographic Message Syntax) object (as defined in RFC 2630) with content type Signed-data. The signature is calculated and created as per section 5 of RFC 2630 using the latest signing algorithm selected with selectSigningAlgorithm(). The content is made of the termination text. The "external signature" construct shall not be used (i.e. the eContent field in the EncapsulatedContentInfo field shall be present and contain the termination text string). The signing-time attribute, as defined in section 11.3 of RFC 2630, shall also be used to provide replay prevention. The client uses this to confirm its identity to the framework. The framework can check that the terminationText has been signed by the client. If a match is made, the interface is released, otherwise the P_INVALID_SIGNATURE exception will be thrown.

Raises

TpCommonExceptions, P_INVALID_SIGNATURE, P_INVALID_INTERFACE_NAME

End of change in Clause 6.3.1.6.8

Change in Clause 7.3.2.1.2

7.3.2.1.2 Method terminateServiceAgreement()

This method is used by the framework to terminate an agreement for the service.

Parameters

`serviceToken : in TpServiceToken`

This is the token passed back from the framework in a previous selectService() method call. This token is used to identify the service agreement to be terminated. If the serviceToken is invalid, or unknown to the client application, the P_INVALID_SERVICE_TOKEN exception will be thrown.

`terminationText : in TpString`

This is the termination text that describes the reason for the termination of the service agreement.

`digitalSignature : in TpOctetSet`

This contains a CMS (Cryptographic Message Syntax) object (as defined in RFC 2630) with content type Signed-data. The signature is calculated and created as per section 5 of RFC 2630 using the same signing algorithm as was used to initially sign the service agreement. The content is the termination text. The "external signature" construct shall not be used (i.e. the eContent field in the EncapsulatedContentInfo field shall be present and contain the termination text string). The signing-time attribute, as defined in section 11.3 of RFC 2630, shall also be used to provide replay prevention. The signing algorithm used is the same as the signing algorithm given when the service agreement was signed using signServiceAgreement(). The framework uses this to confirm its identity to the client application. The client application can check that the terminationText has been signed by the framework. If a match is made, the service agreement is terminated, otherwise the P_INVALID_SIGNATURE exception will be thrown.

Raises

`TpCommonExceptions, P_INVALID_SERVICE_TOKEN, P_INVALID_SIGNATURE`

End of change in Clause 7.3.2.1.2

Change in Clause 7.3.2.2.1

7.3.2.2.1 Method signServiceAgreement()

After the framework has called signServiceAgreement() on the application's IpAppServiceAgreementManagement interface, this method is used by the client application to request that the framework sign the service agreement, which allows the client application to use the service. A reference to the service manager interface of the service is returned to the client application. The service manager returned will be configured as per the service level agreement. If the framework uses 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. If the client application is not allowed to access the service, then an error code (P_SERVICE_ACCESS_DENIED) is returned. If the client application invokes this method before the processing (i.e. digital signature verification) the reponse of signServiceAgreement() on the application's IpAppServiceAgreementManagement interface completed, a TpCommonExceptions with ExceptionType P_INVALID_STATE may be raised to indicate that this method is currently unable to complete the method due to a race condition. In this case, the TpCommonExceptions with ExceptionType P_INVALID_STATE suggests the application to retry the method invocation after a reasonable amount of time has passed.

Returns <signatureAndServiceMgr> : This contains the digital signature of the framework for the service agreement, and a reference to the service manager interface of the service.

```
structure TpSignatureAndServiceMgr {
        digitalSignature:  TpOctetSet;
            serviceMgrInterface:  IpServiceRef;
                };
```

The digitalSignature contains a CMS (Cryptographic Message Syntax) object (as defined in RFC 2630) with content type Signed-data. The signature is calculated and created as per section 5 of RFC 2630. The content is the agreement text given by the client application. The "external signature" construct shall not be used (i.e. the eContent field in the

EncapsulatedContentInfo field shall be present and contain the agreement text string). The signing-time attribute, as defined in section 11.3 of RFC 2630, shall also be used to provide replay prevention.

The serviceMgrInterface is a reference to the service manager interface for the selected service.

Parameters

serviceToken : in TpServiceToken

This is the token returned by the framework in a call to the selectService() method. This token is used to identify the service instance requested by the client application. If the serviceToken is invalid, or has expired, an error code (P_INVALID_SERVICE_TOKEN) is returned.

agreementText : in TpString

This is the agreement text that is to be signed by the framework using the private key of the framework. If the agreementText is invalid, then an error code (P_INVALID_AGREEMENT_TEXT) is returned.

signingAlgorithm : in TpSigningAlgorithm

This is the algorithm used to compute the digital signature. It shall be identical to the one ~~chosen~~ used by the framework when invoking signServiceAgreement() on the client~~in response to IpAccess.selectSigningAlgorithm()~~. If the signingAlgorithm is not the ~~chosen~~ same one, is invalid, or unknown to the framework, an error code (P_INVALID_SIGNING_ALGORITHM) is returned. The list of possible algorithms is as specified in the TpSigningAlgorithm table. The identifier used in this parameter must correspond to the digestAlgorithm and signatureAlgorithm fields in the SignerInfo field in the digitalSignature (see below).

Returns

TpSignatureAndServiceMgr

Raises

TpCommonExceptions, P_ACCESS_DENIED, P_INVALID_AGREEMENT_TEXT, P_INVALID_SERVICE_TOKEN, P_INVALID_SIGNING_ALGORITHM, P_SERVICE_ACCESS_DENIED

End of change in Clause 7.3.2.2.1

Change in Clause 7.3.2.2.2

7.3.2.2.2 Method terminateServiceAgreement()

This method is used by the client application to terminate an agreement for the service.

Parameters

serviceToken : in TpServiceToken

This is the token passed back from the framework in a previous selectService() method call. This token is used to identify the service agreement to be terminated. If the serviceToken is invalid, or has expired, an error code (P_INVALID_SERVICE_TOKEN) is returned.

terminationText : in TpString

This is the termination text that describes the reason for the termination of the service agreement.

digitalSignature : in TpOctetSet

This contains a CMS (Cryptographic Message Syntax) object (as defined in RFC 2630) with content type Signed-data. The signature is calculated and created as per section 5 of RFC 2630 using the same signing algorithm as was used to initially sign the service agreement. The content is the termination text. The "external signature" construct shall not be used (i.e. the eContent field in the EncapsulatedContentInfo field shall be present and contain the termination text string). The signing-time attribute, as defined in section 11.3 of RFC 2630, shall also be used to provide replay prevention. The signing algorithm used is the same as the signing algorithm given when the service agreement was signed using signServiceAgreement(). The framework uses this to check that the terminationText has been signed by the client application. If a match is made, the service agreement is terminated, otherwise an error code (P_INVALID_SIGNATURE) is returned.

Raises

TpCommonExceptions, P_ACCESS_DENIED, P_INVALID_SERVICE_TOKEN, P_INVALID_SIGNATURE

End of change in Clause 7.3.2.2.2

Annex D (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Mar 2001	CN_11	NP-010134	047	--	CR 29.198: for moving TS 29.198 from R99 to Rel 4 (N5-010158)	3.2.0	4.0.0
Jun 2001	CN_12	NP-010330	001	--	Corrections to OSA API Rel4	4.0.0	4.0.1
Sep 2001	CN_13	NP-010466	002	--	Changing references to JAIN	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	003	--	Update to the definitions of method svcUnavailableInd	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	004	--	Only one subject per method invocation for fault and load management	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	005	--	Fault management is missing some *Err methods	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	006	--	Method balance on Fault management interfaces	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	007	--	Change "TpString" into "TpOctetSets" in authentication and access	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	008	--	Replacement of register/unregisterLoadController	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	009	--	Redundant Framework Heartbeat Mechanism	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	010	--	Add a releaseInterface() method to IpAccess	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	011	--	Removal of serviceID from queryAppLoadReq()	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	012	--	Addition of listInterfaces() method	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	013	--	Introduction and use of new Service Instance ID	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	014	--	P_UNAUTHORISED_PARAMETER_VALUE thrown if non-accessible serviceID is provided	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	015	--	Introduction of Service Instance Lifecycle Management	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	016	--	Add support for multi-vendorship	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	017	--	Removal of P_SERVICE_ACCESS_TYPE	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	018	--	Confusing meaning of prescribedMethod	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	019	--	A client should only have one instance of a given service	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	020	--	Some methods on the IpApp interfaces should throw exceptions	4.1.0	4.2.0
Dec 2001	CN_14	NP-010596	021	--	Replace Out Parameters with Return Types	4.2.0	4.3.0
Dec 2001	CN_14	NP-010596	022	--	Correctionto Framework (FW)	4.2.0	4.3.0
Mar 2002	CN_15	NP-020105	023	--	Add P_INVALID_INTERFACE_TYPE exception to IpService.setCallback() and IpService.setCallbackWithSessionID()	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	024	--	Replace erroneous mention of P_OSA_ACCESS by the correct value P_OSA_AUTHENTICATION	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	025	--	Add missing inheritance in service agreement management interfaces	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	026	--	Include Operation Set as part of General Service Properties	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	027	--	Improved description of activityTestReq with respect to ServiceInstanceID	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	028	--	OSA Framework - Generate statistics records on behalf of another entity using genFaultStatsRecordReq	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	029	--	Update the interface names for alignment between 3GPP and ETSI/Parlay	4.3.0	4.4.0
Jun 2002	CN_16	NP-020179	030	--	Solving the problem in the OSA Framework with method appUnavailableInd() in a scenario with multiple service sessions per access session	4.4.0	4.5.0
Jun 2002	CN_16	NP-020179	031	--	Adding missing mandatory method (authenticationSucceeded) to sequence flow	4.4.0	4.5.0
Jun 2002	CN_16	NP-020186	032	--	Remove redundant data type definition TpServiceSpecString	4.5.0	5.0.0
Jun 2002	CN_16	NP-020181	033	--	Addition of support for Java API technology realisation	4.5.0	5.0.0
Jun 2002	CN_16	NP-020182	035	--	Addition of support for WSDL realisation	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	036	--	Clarify semantics of service properties of type BOOLEAN_SET	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	037	--	Addition of version management support to the Framework (29.198-03) in run-time	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	038	--	Enhancements on subscription management error information	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	039	--	Delete conflicting description of P_APPLICATION_NOT_ACTIVATED	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	040	--	Note added for P_SERVICE_INSTANCE Choice Element Name	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	041	--	Correcting the method descriptions for abortAuthentication and for initiateAuthentication	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	042	--	Correcting the description of heartbeat failure	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	043	--	Correcting erroneous FW<->Service instance sequence diagrams	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	044	--	Correcting the scope of TpFwID, which currently is giving it false limitations	4.5.0	5.0.0
Sep 2002	CN_17	NP-020428	046		Correction to description of TpServicePropertyTypeName	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	047		Remove undefined exception in registerService	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	048		Remove ServiceIDs from IpFwFaultManager.genFaultStatsRecordReq()	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	049		Correct appUnavailableInd and related methods	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	050		Remove unusable exception from IpFaultManager.appActivityTestRes()	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	051		Clarify the sequence of events in signing the service agreement	5.0.0	5.1.0

Sep 2002	CN_17	NP-020428	052		Correct use of electronic signatures	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	053		Addition of Sequence Diagrams for terminateAccess	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	054		Add indication what part of service agreement must be signed	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	055		Add text to clarify requirements on support of methods	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	056		Introduce types and modes for generic properties	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	057		Correction on use of NULL in Framework API	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	058		Add Negotiation of Authentication Mechanism for OSA level Authentication	5.0.0	5.1.0
Sep 2002	CN_17	NP-020395	058		Add text to clarify relationship between 3GPP and ETSI/Parlay OSA specifications	5.0.0	5.1.0
Mar 2003	CN_19	NP-030019	063	-	Correction to Initial Access Sequence Diagram	5.1.0	5.2.0
Mar 2003	CN_19	NP-030019	065	-	Enable creation/destruction of load level notifications at the request of Framework	5.1.0	5.2.0
Mar 2003	CN_19	NP-030019	067	-	Correction of Sequence for Framework – Service load management	5.1.0	5.2.0
Mar 2003	CN_19	NP-030019	074	-	Add Initial Load Notification report for Framework Integrity Management Load Notification model	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	068	--	Correction to Application's requirements for supporting methods	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	069	--	Correction of status of methods to interfaces in clause 7.3	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	070	--	Correction of status of methods to interfaces in clause 8.3	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	071	--	Correction of status of methods to interfaces in clause 6.3	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	075	--	Adding the appAvailStatusInd() and svcAvailStatusInd() methods	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	076	--	Remove race condition in signServiceAgreement	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	077	--	Change reference to deprecated method "authenticate" in TpAuthMechanism to "challenge"	5.1.0	5.2.0
Jun 2003	CN_20	NP-030237	079	--	Correction to TpEncryptionCapability to correct support for Triple-DES	5.2.0	5.3.0
Jun 2003	CN_20	NP-030237	081	--	Correction of the Framework Service Instance Lifecycle Manager Sequence Diagram	5.2.0	5.3.0
Jun 2003	CN_20	NP-030237	083	--	Correction of the use of TpDomainID in Framework initiateAuthentication method	5.2.0	5.3.0
Sep 2003	CN_21	NP-030352	085	--	Correction to Java Realisation Annex	5.3.0	5.4.0
Dec 2003	CN_22	NP-030549	086	--	Correction of the sequence diagram for Fault Management	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	087	--	Correction of State Transition Diagram for IpAccess	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	088	--	Correction of Correlation Behaviour in Load Management	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	089	--	Correction of Correlation Behaviour in Fault Management	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	090	--	Correction and Clarification of Framework Access Session Behaviour	5.4.0	5.5.0
Apr 2004	CN_23bis	NP-040155	101	--	Correct Java Code to conform with Java Rulebook in TS 29.198-01 and to remove errors	5.5.0	5.6.0

joint-API-group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) ***N5-040352***
Meeting #27, Miami, FL, USA, 10-14 May 2004

CR-Form-v7

CHANGE REQUEST

⌘ **29.198-03** **CR** **117** ⌘ **rev** **-** ⌘ Current version: **6.0.1** ⌘

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

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network **X**

Title:	⌘	Correct description for the use of selectSigningAlgorithm
Source:	⌘	CN5 Lucent Technologies
Work item code:	⌘	OSA2 ***Date:*** ⌘ 18/05/2004
Category:	⌘	**A** ***Release:*** ⌘ Rel-6

Use one 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 one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change:	⌘	The described usage of the selectSigningAlgorithm() method is ambiguous and requires correction
Summary of change:	⌘	Additional clarifying text has been added to correct the description of selectSigningAlgorithm() and to each of the methods that use a digital signature.
Consequences if not approved:	⌘	A client application could use an incorrect signing algorithm with some methods resulting in operational failures.

Clauses affected:	⌘	6.3.1.6.6, 6.3.1.6.7, 6.3.1.6.8, 7.3.2.1.2, 7.3.2.2.1, 7.3.2.2.2

Other specs affected: ⌘	Y	N		⌘
		X	Other core specifications	
		X	Test specifications	
		X	O&M Specifications	

Other comments: ⌘ Mirror CR for Rel-5 in N5-040351

Change in Clause 6.3.1.6.6

6.3.1.6.6 Method <<new>> selectSigningAlgorithm()

The client uses this method to inform the Framework of the different signing algorithms it supports for use in all cases where digital signatures are required. The Framework will select one of the suggested algorithms. This method shall be the first method invoked by the client on IpAccess. The algorithm chosen as a result of the response to this method remains valid for an instance of IpAccess and until this method is re-invoked by the client.

Subsequent invocations of selectSigningAlgorithm() may change the signing algorithm used during the access session. However, once signServiceAgreement() has been called on the client by the framework, the signing algorithm currently selected must be used for the client's invocation of signServiceAgreement() on the Framework as well as for subsequent calls to terminateServiceAgreement(). Other operations requiring digital signatures will use the latest algorithm specified by selectSigningAlgorithm().

If an algorithm that is acceptable to the framework within the capability of the client cannot be found, the framework throws the P_NO_ACCEPTABLE_SIGNING_ALGORITHM exception.

Returns: selectedAlgorithm. This is the signing algorithm chosen by the Framework. The chosen algorithm shall be taken from the list proposed by the Client.

Parameters

signingAlgorithmCaps : in TpSigningAlgorithmCapabilityList

The list of signing algorithms supported by the client.

Returns

TpSigningAlgorithm

Raises

TpCommonExceptions, P_ACCESS_DENIED, P_NO_ACCEPTABLE_SIGNING_ALGORITHM

End of change in Clause 6.3.1.6.6

Change in Clause 6.3.1.6.7

6.3.1.6.7 Method <<new>> terminateAccess()

The terminateAccess method is used by the client to request that its access session with the framework is ended. After it is invoked, the client will no longer be authenticated with the framework. The client will not be able to use the references to any of the framework interfaces gained during the access session. Any calls to these interfaces will fail. Also, all remaining service instances created by the framework either directly in this access session or on behalf of the client during this access session shall be terminated.

Parameters

terminationText : in TpString

This is the termination text describes the reason for the termination of the access session.

digitalSignature : in TpOctetSet

This contains a CMS (Cryptographic Message Syntax) object (as defined in RFC 2630) with content type Signed-data. The signature is calculated and created as per section 5 of RFC 2630 using the latest signing algorithm selected with selectSigningAlgorithm(). The content is made of the termination text. The "external signature" construct shall not be used (i.e. the eContent field in the EncapsulatedContentInfo field shall be present and contain the termination text string). The signing-time attribute, as defined in section 11.3 of RFC 2630, shall also be used to provide replay

prevention. The client uses this to confirm its identity to the framework. The framework can check that the terminationText has been signed by the client. If a match is made, the access session is terminated, otherwise the P_INVALID_SIGNATURE exception will be thrown.

Raises

TpCommonExceptions, P_INVALID_SIGNATURE

End of change in Clause 6.3.1.6.7

Change in Clause 6.3.1.6.8

6.3.1.6.8 Method <<new>> relinquishInterface()

The client uses this method to release an instance of a framework interface that was obtained during this access session.

Parameters

interfaceName : in TpInterfaceName

This is the name of the framework interface which is being released. If the interfaceName is invalid, the framework throws the P_INVALID_INTERFACE_NAME exception. If the interface has not been given to the client during this access session, then the P_TASK_REFUSED exception will be thrown.

terminationText : in TpString

This is the termination text describes the reason for the release of the interface. This text is required simply because the digitalSignature parameter requires a terminationText to sign.

digitalSignature : in TpOctetSet

This contains a CMS (Cryptographic Message Syntax) object (as defined in RFC 2630) with content type Signed-data. The signature is calculated and created as per section 5 of RFC 2630 using the latest signing algorithm selected with selectSigningAlgorithm(). The content is made of the termination text. The "external signature" construct shall not be used (i.e. the eContent field in the EncapsulatedContentInfo field shall be present and contain the termination text string). The signing-time attribute, as defined in section 11.3 of RFC 2630, shall also be used to provide replay prevention. The client uses this to confirm its identity to the framework. The framework can check that the terminationText has been signed by the client. If a match is made, the interface is released, otherwise the P_INVALID_SIGNATURE exception will be thrown.

Raises

TpCommonExceptions, P_INVALID_SIGNATURE, P_INVALID_INTERFACE_NAME

End of change in Clause 6.3.1.6.8

Change in Clause 7.3.2.1.2

7.3.2.1.2 Method terminateServiceAgreement()

This method is used by the framework to terminate an agreement for the service.

Parameters

serviceToken : in TpServiceToken

This is the token passed back from the framework in a previous selectService() method call. This token is used to identify the service agreement to be terminated. If the serviceToken is invalid, or unknown to the client application, the P_INVALID_SERVICE_TOKEN exception will be thrown.

terminationText : in TpString

This is the termination text that describes the reason for the termination of the service agreement.

digitalSignature : in TpOctetSet

This contains a CMS (Cryptographic Message Syntax) object (as defined in RFC 2630) with content type Signed-data. The signature is calculated and created as per section 5 of RFC 2630 using the same signing algorithm as was used to initially sign the service agreement. The content is the termination text. The "external signature" construct shall not be used (i.e. the eContent field in the EncapsulatedContentInfo field shall be present and contain the termination text string). The signing-time attribute, as defined in section 11.3 of RFC 2630, shall also be used to provide replay prevention. The signing algorithm used is the same as the signing algorithm given when the service agreement was signed using signServiceAgreement(). The framework uses this to confirm its identity to the client application. The client application can check that the terminationText has been signed by the framework. If a match is made, the service agreement is terminated, otherwise the P_INVALID_SIGNATURE exception will be thrown.

Raises

TpCommonExceptions, P_INVALID_SERVICE_TOKEN, P_INVALID_SIGNATURE

End of change in Clause 7.3.2.1.2

Change in Clause 7.3.2.2.1

7.3.2.2.1 Method signServiceAgreement()

After the framework has called signServiceAgreement() on the application's IpAppServiceAgreementManagement interface, this method is used by the client application to request that the framework sign the service agreement, which allows the client application to use the service. A reference to the service manager interface of the service is returned to the client application. The service manager returned will be configured as per the service level agreement. If the framework uses 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. If the client application is not allowed to access the service, then an error code (P_SERVICE_ACCESS_DENIED) is returned. If the client application invokes this method before the processing (i.e. digital signature verification) the reponse of signServiceAgreement() on the application's IpAppServiceAgreementManagement interface completed, a TpCommonExceptions with ExceptionType P_INVALID_STATE may be raised to indicate that this method is currently unable to complete the method due to a race condition. In this case, the TpCommonExceptions with ExceptionType P_INVALID_STATE suggests the application to retry the method invocation after a reasonable amount of time has passed.

There must be only one service instance per client application. Therefore, in case the client attempts to select a service for which it has already signed a service agreement and this service agreement has not been terminated, a reference to the already existing service manager will be returned.

Returns <signatureAndServiceMgr> : This contains the digital signature of the framework for the service agreement, and a reference to the service manager interface of the service.

```
structure TpSignatureAndServiceMgr {
        digitalSignature: TpOctetSet;
        serviceMgrInterface: IpServiceRef;
        };
```

The digitalSignature contains a CMS (Cryptographic Message Syntax) object (as defined in RFC 2630) with content type Signed-data. The signature is calculated and created as per section 5 of RFC 2630. The content is the agreement text given by the client application. The "external signature" construct shall not be used (i.e. the eContent field in the EncapsulatedContentInfo field shall be present and contain the agreement text string). The signing-time attribute, as defined in section 11.3 of RFC 2630, shall also be used to provide replay prevention.

The serviceMgrInterface is a reference to the service manager interface for the selected service.

Parameters

`serviceToken : in TpServiceToken`

This is the token returned by the framework in a call to the selectService() method. This token is used to identify the service instance requested by the client application. If the serviceToken is invalid, or has expired, an error code (P_INVALID_SERVICE_TOKEN) is returned.

`agreementText : in TpString`

This is the agreement text that is to be signed by the framework using the private key of the framework. If the agreementText is invalid, then an error code (P_INVALID_AGREEMENT_TEXT) is returned.

`signingAlgorithm : in TpSigningAlgorithm`

This is the algorithm used to compute the digital signature. It shall be identical to the one ~~chosen~~ used by the framework when invoking signServiceAgreement() on the client~~in response to IpAccess.selectSigningAlgorithm()~~. If the signingAlgorithm is not the ~~chosen~~ same one, is invalid, or unknown to the framework, an error code (P_INVALID_SIGNING_ALGORITHM) is returned. The list of possible algorithms is as specified in the TpSigningAlgorithm table. The identifier used in this parameter must correspond to the digestAlgorithm and signatureAlgorithm fields in the SignerInfo field in the digitalSignature (see below).

Returns

`TpSignatureAndServiceMgr`

Raises

`TpCommonExceptions, P_ACCESS_DENIED, P_INVALID_AGREEMENT_TEXT, P_INVALID_SERVICE_TOKEN, P_INVALID_SIGNING_ALGORITHM, P_SERVICE_ACCESS_DENIED`

End of change in Clause 7.3.2.2.1

Change in Clause 7.3.2.2.2

7.3.2.2.2 Method terminateServiceAgreement()

This method is used by the client application to terminate an agreement for the service.

Parameters

`serviceToken : in TpServiceToken`

This is the token passed back from the framework in a previous selectService() method call. This token is used to identify the service agreement to be terminated. If the serviceToken is invalid, or has expired, an error code (P_INVALID_SERVICE_TOKEN) is returned.

`terminationText : in TpString`

This is the termination text that describes the reason for the termination of the service agreement.

digitalSignature : in TpOctetSet

This contains a CMS (Cryptographic Message Syntax) object (as defined in RFC 2630) with content type Signed-data. The signature is calculated and created as per section 5 of RFC 2630 using the same signing algorithm as was used to initially sign the service agreement. The content is the termination text. The "external signature" construct shall not be used (i.e. the eContent field in the EncapsulatedContentInfo field shall be present and contain the termination text string). The signing-time attribute, as defined in section 11.3 of RFC 2630, shall also be used to provide replay prevention. The signing algorithm used is the same as the signing algorithm given when the service agreement was signed using signServiceAgreement(). The framework uses this to check that the terminationText has been signed by the client application. If a match is made, the service agreement is terminated, otherwise an error code (P_INVALID_SIGNATURE) is returned.

Raises

TpCommonExceptions, P_ACCESS_DENIED, P_INVALID_SERVICE_TOKEN, P_INVALID_SIGNATURE

End of change in Clause 7.3.2.2.2

Annex E (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Mar 2001	CN_11	NP-010134	047	--	CR 29.198: for moving TS 29.198 from R99 to Rel 4 (N5-010158)	3.2.0	4.0.0
Jun 2001	CN_12	NP-010330	001	--	Corrections to OSA API Rel4	4.0.0	4.0.1
Sep 2001	CN_13	NP-010466	002	--	Changing references to JAIN	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	003	--	Update to the definitions of method svcUnavailableInd	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	004	--	Only one subject per method invocation for fault and load management	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	005	--	Fault management is missing some *Err methods	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	006	--	Method balance on Fault management interfaces	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	007	--	Change "TpString" into "TpOctetSets" in authentication and access	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	008	--	Replacement of register/unregisterLoadController	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	009	--	Redundant Framework Heartbeat Mechanism	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	010	--	Add a releaseInterface() method to IpAccess	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	011	--	Removal of serviceID from queryAppLoadReq()	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	012	--	Addition of listInterfaces() method	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	013	--	Introduction and use of new Service Instance ID	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	014	--	P_UNAUTHORISED_PARAMETER_VALUE thrown if non-accessible serviceID is provided	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	015	--	Introduction of Service Instance Lifecycle Management	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	016	--	Add support for multi-vendorship	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	017	--	Removal of P_SERVICE_ACCESS_TYPE	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	018	--	Confusing meaning of prescribedMethod	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	019	--	A client should only have one instance of a given service	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	020	--	Some methods on the IpApp interfaces should throw exceptions	4.1.0	4.2.0
Dec 2001	CN_14	NP-010596	021	--	Replace Out Parameters with Return Types	4.2.0	4.3.0
Dec 2001	CN_14	NP-010596	022	--	Correctionto Framework (FW)	4.2.0	4.3.0
Mar 2002	CN_15	NP-020105	023	--	Add P_INVALID_INTERFACE_TYPE exception to IpService.setCallback() and IpService.setCallbackWithSessionID()	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	024	--	Replace erroneous mention of P_OSA_ACCESS by the correct value P_OSA_AUTHENTICATION	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	025	--	Add missing inheritance in service agreement management interfaces	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	026	--	Include Operation Set as part of General Service Properties	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	027	--	Improved description of activityTestReq with respect to ServiceInstanceID	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	028	--	OSA Framework - Generate statistics records on behalf of another entity using genFaultStatsRecordReq	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	029	--	Update the interface names for alignment between 3GPP and ETSI/Parlay	4.3.0	4.4.0
Jun 2002	CN_16	NP-020179	030	--	Solving the problem in the OSA Framework with method appUnavailableInd() in a scenario with multiple service sessions per access session	4.4.0	4.5.0
Jun 2002	CN_16	NP-020179	031	--	Adding missing mandatory method (authenticationSucceeded) to sequence flow	4.4.0	4.5.0
Jun 2002	CN_16	NP-020186	032	--	Remove redundant data type definition TpServiceSpecString	4.5.0	5.0.0
Jun 2002	CN_16	NP-020181	033	--	Addition of support for Java API technology realisation	4.5.0	5.0.0
Jun 2002	CN_16	NP-020182	035	--	Addition of support for WSDL realisation	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	036	--	Clarify semantics of service properties of type BOOLEAN_SET	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	037	--	Addition of version management support to the Framework (29.198-03) in run-time	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	038	--	Enhancements on subscription management error information	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	039	--	Delete conflicting description of P_APPLICATION_NOT_ACTIVATED	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	040	--	Note added for P_SERVICE_INSTANCE Choice Element Name	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	041	--	Correcting the method descriptions for abortAuthentication and for initiateAuthentication	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	042	--	Correcting the description of heartbeat failure	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	043	--	Correcting erroneous FW<->Service instance sequence diagrams	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	044	--	Correcting the scope of TpFwID, which currently is giving it false limitations	4.5.0	5.0.0
Sep 2002	CN_17	NP-020428	046		Correction to description of TpServicePropertyTypeName	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	047		Remove undefined exception in registerService	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	048		Remove ServiceIDs from IpFwFaultManager.genFaultStatsRecordReq()	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	049		Correct appUnavailableInd and related methods	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	050		Remove unusable exception from IpFaultManager.appActivityTestRes()	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	051		Clarify the sequence of events in signing the service agreement	5.0.0	5.1.0

Sep 2002	CN_17	NP-020428	052		Correct use of electronic signatures	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	053		Addition of Sequence Diagrams for terminateAccess	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	054		Add indication what part of service agreement must be signed	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	055		Add text to clarify requirements on support of methods	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	056		Introduce types and modes for generic properties	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	057		Correction on use of NULL in Framework API	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	058		Add Negotiation of Authentication Mechanism for OSA level Authentication	5.0.0	5.1.0
Sep 2002	CN_17	NP-020395	058		Add text to clarify relationship between 3GPP and ETSI/Parlay OSA specifications	5.0.0	5.1.0
Mar 2003	CN_19	NP-030019	063	-	Correction to Initial Access Sequence Diagram	5.1.0	5.2.0
Mar 2003	CN_19	NP-030019	065	-	Enable creation/destruction of load level notifications at the request of Framework	5.1.0	5.2.0
Mar 2003	CN_19	NP-030019	067	-	Correction of Sequence for Framework – Service load management	5.1.0	5.2.0
Mar 2003	CN_19	NP-030019	074	-	Add Initial Load Notification report for Framework Integrity Management Load Notification model	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	068	--	Correction to Application's requirements for supporting methods	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	069	--	Correction of status of methods to interfaces in clause 7.3	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	070	--	Correction of status of methods to interfaces in clause 8.3	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	071	--	Correction of status of methods to interfaces in clause 6.3	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	075	--	Adding the appAvailStatusInd() and svcAvailStatusInd() methods	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	076	--	Remove race condition in signServiceAgreement	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	077	--	Change reference to deprecated method "authenticate" in TpAuthMechanism to "challenge"	5.1.0	5.2.0
Jun 2003	CN_20	NP-030237	079	--	Correction to TpEncryptionCapability to correct support for Triple-DES	5.2.0	5.3.0
Jun 2003	CN_20	NP-030237	081	--	Correction of the Framework Service Instance Lifecycle Manager Sequence Diagram	5.2.0	5.3.0
Jun 2003	CN_20	NP-030237	083	--	Correction of the use of TpDomainID in Framework initiateAuthentication method	5.2.0	5.3.0
Sep 2003	CN_21	NP-030352	085	--	Correction to Java Realisation Annex	5.3.0	5.4.0
Dec 2003	CN_22	NP-030549	086	--	Correction of the sequence diagram for Fault Management	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	087	--	Correction of State Transition Diagram for IpAccess	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	088	--	Correction of Correlation Behaviour in Load Management	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	089	--	Correction of Correlation Behaviour in Fault Management	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	090	--	Correction and Clarification of Framework Access Session Behaviour	5.4.0	5.5.0
Dec 2003	CN_22	NP-030553	091	--	Add OSA API support for 3GPP2 networks	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	092	--	Add description for service super and sub types	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	093	--	Add support for registration of additional service property types and modes	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	094	--	Improve User Interaction message management functions	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	095	--	Add new values for TpServiceTypeName for Policy Management	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	096	--	Allow for applications to re-obtain the reference to the service manager	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	097	--	Add support in OSA to inform applications about new SCSs and their level of Backward compatibility – Align with SA1's 22.127	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	098	--	Add “Extended User Status” as service type name - Align with 29.198-06	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	099	--	Add P_USER_BINDING to TpServiceTypeName	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	100	--	Modify Framework Availability Indication in Fault Management	5.5.0	6.0.0
Feb 2004	--	--	--	--	Added Java code attachment 2919803J2EE.zip which was delivered late by outside developers. See Annex C.	6.0.0	6.0.1

CR-Form-v7

CHANGE REQUEST

⌘ **29.198-03** **CR** **118** ⌘ **rev** **-** ⌘ Current version: **5.6.0** ⌘

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

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network **X**

Title:	⌘	Correct the description of the usage of CHAP within authentication
Source:	⌘	CN5 Lucent Technologies
Work item code:	⌘	OSA2
Date:	⌘	18/05/2004
Category:	⌘	**F**
Release:	⌘	Rel-5

Use one 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 one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change:	⌘	Correct the text defining the usage of the CHAP protocol during authentication within the challenge() methods to eliminate differing interpretations of its usage that have resulted in interoperability problems, which have been discovered in various interoperability testing events.
Summary of change:	⌘	Corrected description and clarifying steps on the usage of CHAP have been added to the challenge() method descriptions to define how it is used in authentication. The same description and steps have been applied to both challenge() methods.
Consequences if not approved:	⌘	Confusion and disagreement as to precisely how the CHAP exchange is used within authentication would remain, and there will be continuing interoperability problems between vendors resulting in inability to deliver services.

Clauses affected:	⌘	6.3.1.1.4, 6.3.1.5.6

Other specs affected: ⌘	Y	N	
		X	Other core specifications
		X	Test specifications
		X	O&M Specifications

⌘

Other comments:	⌘	Mirror CR for Rel-6 in N5-040354

Change in Clause 6.3.1.1.4

6.3.1.1.4 Method <<new>> challenge()

This method is used by the framework to authenticate the client. The client must respond with the correct responses to the challenges presented by the framework. The number of exchanges is dependent on the policies of each side. The authentication of the client is deemed successful when the authenticationSucceeded method is invoked by the Framework.

The invocation of this method may be interleaved with challenge() calls by the client on the IpAPILevelAuthentication interface. The client shall respond immediately to authentication challenges from the Framework, and not wait until the Framework has responded to any challenge the client may issue.

This method shall only be used when the method initiateAuthenticationWithVersion() is used on the IpInitial interface.

Returns <response> : This is the response of the client application to the challenge of the framework in the current sequence. The formatting and construction of this parameter shall be according to section 4.1 of RFC 1994. A complete CHAP Response packet shall be used to carry the response ~~string~~octet set. That octet set will be the result of applying the designated hashing algorithm, which is indicated via the client's invocation of selectAuthenticationMechanism(), to an octet set consisting of the concatenation of the CHAP Identifier, the shared "secret", and the supplied challenge value. The Name field of the CHAP Response packet must be present and contain a valid value in order for the CHAP Response to be valid. However, the Name field is not used in the authentication process.~~The Response packet shall make the contents of this returned parameter. The Name field of the CHAP Response packet shall be present but not contain any useful value.~~

Parameters

`challenge : in TpOctetSet`

The challenge presented by the framework to be responded to by the client. The challenge format used will be in accordance with the IETF PPP Authentication Protocols - Challenge Handshake Authentication Protocol (RFC 1994).

The challenge octet set must be formatted as a CHAP Challenge packet as defined in section 4.1 of RFC 1994. A complete and properly formatted CHAP Challenge packet must be used. The Name field of the CHAP Challenge packet must be present and contain a valid value in order for the CHAP Response to be valid. However, the Name field is not used in the authentication process.~~The formatting of the challenge value shall be according to section 4.1 of RFC 1994. A complete CHAP Request packet shall be used to carry the challenge value. The Name field of the CHAP Request packet shall be present but not contain any useful value.~~

Steps for constructing the challenge octet set:

1. Create a random challenge value (octet set). Per RFC 1994, this value must between 1 and 255 octets in length.
2. Construct a CHAP Challenge packet based on 4.1 of RFC 1994 with the octet set from the previous step passed in the Value field within the CHAP Challenge.

Returns

`TpOctetSet`

Steps for constructing the response octet set:

1. Extract the Identifier and Value fields from the CHAP Challenge packet passed in the challenge() method's challenge parameter
2. Build an octet set consisting of the concatenation of the Identifier, the "shared secret", and the Value from the CHAP Challenge
3. Compute the hash of the octet set resulting from the previous step using the designated hashing algorithm
4. Construct a complete CHAP Response packet with the resulting octet set from previous step as the CHAP Value

Steps for validating the response octet set:

1. Verify that the Identifier sent in the original CHAP Challenge matches the Identifier received in the CHAP Response. If it does not, authentication fails.

2. Build an octet set consisting of the concatenation of the original Identifier, the "shared secret", and the original challenge value

3. Compute the hash of the resulting octet set from the previous step using the designated hashing algorithm

4. Verify the octet set resulting from the previous step matches the octet set contained in the Value field of the CHAP Response. A match indicates successful authentication.

End of change in Clause 6.3.1.1.4

Change in Clause 6.3.1.5.6

6.3.1.5.6 Method <<new>> challenge()

This method is used by the client to authenticate the framework. The framework must respond with the correct responses to the challenges presented by the client. The domainID received in the initiateAuthenticationWithVersion() 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 authentication of the framework is deemed successful when the authenticationSucceeded method is invoked by the client.

The invocation of this method may be interleaved with challenge() calls by the framework on the client's APILevelAuthentication interface.

This method shall only be used when the IpAPILevelAuthentication interface is obtained by using initiateAuthenticationWithVersion() on the IpInitial interface.

Returns <response> : This is the response of the framework to the challenge of the client in the current sequence. The formatting and construction of this parameter shall be according to section 4.1 of RFC 1994. A complete CHAP Response packet shall be used to carry the response ~~string~~octet set. That octet set will be the result of applying the designated hashing algorithm, which is indicated via the client's invocation of selectAuthenticationMechanism(), to an octet set consisting of the concatenation of the CHAP Identifier, the shared "secret", and the supplied challenge value. The Name field of the CHAP Response packet must be present and contain a valid value in order for the CHAP Response to be valid. However, the Name field is not used in the authentication process.~~The Response packet shall make the contents of this returned parameter. The Name field of the CHAP Response packet shall be present but not contain any useful value.~~

Parameters

`challenge : in TpOctetSet`

The challenge presented by the client to be responded to by the framework. The challenge format used will be in accordance with the IETF PPP Authentication Protocols - Challenge Handshake Authentication Protocol (RFC 1994).

The challenge octet set must be formatted as a CHAP Challenge packet as defined in section 4.1 of RFC 1994. A complete and properly formatted CHAP Challenge packet must be used. The Name field of the CHAP Challenge packet must be present and contain a valid value in order for the CHAP Response to be valid. However, the Name field is not used in the authentication process.~~The formatting of the challenge value shall be according to section 4.1 of RFC 1994. A complete CHAP Request packet shall be used to carry the challenge value. The Name field of the CHAP Request packet shall be present but not contain any useful value.~~

Steps for constructing the challenge octet set:

1. Create a random challenge value (octet set). Per RFC 1994, this value must between 1 and 255 octets in length.

2. Construct a CHAP Challenge packet based on 4.1 of RFC 1994 with the octet set from the previous step passed in the Value field within the CHAP Challenge.

Returns

TpOctetSet

Steps for constructing the response octet set:

1. Extract the Identifier and Value fields from the CHAP Challenge packet passed in the challenge() method's challenge parameter
2. Build an octet set consisting of the concatenation of the Identifier, the "shared secret", and the Value from the CHAP Challenge
3. Compute the hash of the octet set resulting from the previous step using the designated hashing algorithm
4. Construct a complete CHAP Response packet with the resulting octet set from previous step as the CHAP Value

Steps for validating the response octet set:

1. Verify that the Identifier sent in the original CHAP Challenge matches the Identifier received in the CHAP Response. If it does not, authentication fails.
2. Build an octet set consisting of the concatenation of the original Identifier, the "shared secret", and the original challenge value
3. Compute the hash of the resulting octet set from the previous step using the designated hashing algorithm

Verify the octet set resulting from the previous step matches the octet set contained in the Value field of the CHAP Response. A match indicates successful authentication.

Raises

TpCommonExceptions, P_ACCESS_DENIED

End of change in Clause 6.3.1.5.6

Annex D (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Mar 2001	CN_11	NP-010134	047	--	CR 29.198: for moving TS 29.198 from R99 to Rel 4 (N5-010158)	3.2.0	4.0.0
Jun 2001	CN_12	NP-010330	001	--	Corrections to OSA API Rel4	4.0.0	4.0.1
Sep 2001	CN_13	NP-010466	002	--	Changing references to JAIN	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	003	--	Update to the definitions of method svcUnavailableInd	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	004	--	Only one subject per method invocation for fault and load management	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	005	--	Fault management is missing some *Err methods	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	006	--	Method balance on Fault management interfaces	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	007	--	Change "TpString" into "TpOctetSets" in authentication and access	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	008	--	Replacement of register/unregisterLoadController	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	009	--	Redundant Framework Heartbeat Mechanism	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	010	--	Add a releaseInterface() method to IpAccess	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	011	--	Removal of serviceID from queryAppLoadReq()	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	012	--	Addition of listInterfaces() method	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	013	--	Introduction and use of new Service Instance ID	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	014	--	P_UNAUTHORISED_PARAMETER_VALUE thrown if non-accessible serviceID is provided	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	015	--	Introduction of Service Instance Lifecycle Management	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	016	--	Add support for multi-vendorship	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	017	--	Removal of P_SERVICE_ACCESS_TYPE	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	018	--	Confusing meaning of prescribedMethod	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	019	--	A client should only have one instance of a given service	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	020	--	Some methods on the IpApp interfaces should throw exceptions	4.1.0	4.2.0
Dec 2001	CN_14	NP-010596	021	--	Replace Out Parameters with Return Types	4.2.0	4.3.0
Dec 2001	CN_14	NP-010596	022	--	Correctionto Framework (FW)	4.2.0	4.3.0
Mar 2002	CN_15	NP-020105	023	--	Add P_INVALID_INTERFACE_TYPE exception to IpService.setCallback() and IpService.setCallbackWithSessionID()	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	024	--	Replace erroneous mention of P_OSA_ACCESS by the correct value P_OSA_AUTHENTICATION	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	025	--	Add missing inheritance in service agreement management interfaces	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	026	--	Include Operation Set as part of General Service Properties	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	027	--	Improved description of activityTestReq with respect to ServiceInstanceID	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	028	--	OSA Framework - Generate statistics records on behalf of another entity using genFaultStatsRecordReq	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	029	--	Update the interface names for alignment between 3GPP and ETSI/Parlay	4.3.0	4.4.0
Jun 2002	CN_16	NP-020179	030	--	Solving the problem in the OSA Framework with method appUnavailableInd() in a scenario with multiple service sessions per access session	4.4.0	4.5.0
Jun 2002	CN_16	NP-020179	031	--	Adding missing mandatory method (authenticationSucceeded) to sequence flow	4.4.0	4.5.0
Jun 2002	CN_16	NP-020186	032	--	Remove redundant data type definition TpServiceSpecString	4.5.0	5.0.0
Jun 2002	CN_16	NP-020181	033	--	Addition of support for Java API technology realisation	4.5.0	5.0.0
Jun 2002	CN_16	NP-020182	035	--	Addition of support for WSDL realisation	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	036	--	Clarify semantics of service properties of type BOOLEAN_SET	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	037	--	Addition of version management support to the Framework (29.198-03) in run-time	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	038	--	Enhancements on subscription management error information	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	039	--	Delete conflicting description of P_APPLICATION_NOT_ACTIVATED	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	040	--	Note added for P_SERVICE_INSTANCE Choice Element Name	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	041	--	Correcting the method descriptions for abortAuthentication and for initiateAuthentication	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	042	--	Correcting the description of heartbeat failure	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	043	--	Correcting erroneous FW<->Service instance sequence diagrams	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	044	--	Correcting the scope of TpFwID, which currently is giving it false limitations	4.5.0	5.0.0
Sep 2002	CN_17	NP-020428	046		Correction to description of TpServicePropertyTypeName	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	047		Remove undefined exception in registerService	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	048		Remove ServiceIDs from IpFwFaultManager.genFaultStatsRecordReq()	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	049		Correct appUnavailableInd and related methods	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	050		Remove unusable exception from IpFaultManager.appActivityTestRes()	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	051		Clarify the sequence of events in signing the service agreement	5.0.0	5.1.0

Sep 2002	CN_17	NP-020428	052		Correct use of electronic signatures	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	053		Addition of Sequence Diagrams for terminateAccess	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	054		Add indication what part of service agreement must be signed	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	055		Add text to clarify requirements on support of methods	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	056		Introduce types and modes for generic properties	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	057		Correction on use of NULL in Framework API	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	058		Add Negotiation of Authentication Mechanism for OSA level Authentication	5.0.0	5.1.0
Sep 2002	CN_17	NP-020395	058		Add text to clarify relationship between 3GPP and ETSI/Parlay OSA specifications	5.0.0	5.1.0
Mar 2003	CN_19	NP-030019	063	-	Correction to Initial Access Sequence Diagram	5.1.0	5.2.0
Mar 2003	CN_19	NP-030019	065	-	Enable creation/destruction of load level notifications at the request of Framework	5.1.0	5.2.0
Mar 2003	CN_19	NP-030019	067	-	Correction of Sequence for Framework – Service load management	5.1.0	5.2.0
Mar 2003	CN_19	NP-030019	074	-	Add Initial Load Notification report for Framework Integrity Management Load Notification model	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	068	--	Correction to Application's requirements for supporting methods	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	069	--	Correction of status of methods to interfaces in clause 7.3	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	070	--	Correction of status of methods to interfaces in clause 8.3	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	071	--	Correction of status of methods to interfaces in clause 6.3	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	075	--	Adding the appAvailStatusInd() and svcAvailStatusInd() methods	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	076	--	Remove race condition in signServiceAgreement	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	077	--	Change reference to deprecated method "authenticate" in TpAuthMechanism to "challenge"	5.1.0	5.2.0
Jun 2003	CN_20	NP-030237	079	--	Correction to TpEncryptionCapability to correct support for Triple-DES	5.2.0	5.3.0
Jun 2003	CN_20	NP-030237	081	--	Correction of the Framework Service Instance Lifecycle Manager Sequence Diagram	5.2.0	5.3.0
Jun 2003	CN_20	NP-030237	083	--	Correction of the use of TpDomainID in Framework initiateAuthentication method	5.2.0	5.3.0
Sep 2003	CN_21	NP-030352	085	--	Correction to Java Realisation Annex	5.3.0	5.4.0
Dec 2003	CN_22	NP-030549	086	--	Correction of the sequence diagram for Fault Management	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	087	--	Correction of State Transition Diagram for IpAccess	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	088	--	Correction of Correlation Behaviour in Load Management	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	089	--	Correction of Correlation Behaviour in Fault Management	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	090	--	Correction and Clarification of Framework Access Session Behaviour	5.4.0	5.5.0
Apr 2004	CN_23bis	NP-040155	101	--	Correct Java Code to conform with Java Rulebook in TS 29.198-01 and to remove errors	5.5.0	5.6.0

joint-API-group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) ***N5-040354***
Meeting #27, Miami, FL, USA, 10-14 May 2004

CR-Form-v7

CHANGE REQUEST

⌘ **29.198-03** **CR** **119** ⌘ **rev** **-** ⌘ Current version: **6.0.1** ⌘

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

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network **X**

Title:	⌘	Correct the description of the usage of CHAP within authentication
Source:	⌘	CN5 Lucent Technologies
Work item code:	⌘	OSA2
Date:	⌘	18/05/2004
Category:	⌘	**A**
Release:	⌘	Rel-6

Use one 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 one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change:	⌘	Correct the text defining the usage of the CHAP protocol during authentication within the challenge() methods to eliminate differing interpretations of its usage that have resulted in interoperability problems, which have been discovered in various interoperability testing events.
Summary of change:	⌘	Corrected description and clarifying steps on the usage of CHAP have been added to the challenge() method descriptions to define how it is used in authentication. The same description and steps have been applied to both challenge() methods.
Consequences if not approved:	⌘	Confusion and disagreement as to precisely how the CHAP exchange is used within authentication would remain, and there will be continuing interoperability problems between vendors resulting in inability to deliver services.
Clauses affected:	⌘	6.3.1.1.4, 6.3.1.5.6

Other specs affected: ⌘

Y	N	
	X	Other core specifications
	X	Test specifications
	X	O&M Specifications

⌘

Other comments: ⌘ Mirror CR for Rel-5 in N5-040353

Change in Clause 6.3.1.1.4

6.3.1.1.4 Method <<new>> challenge()

This method is used by the framework to authenticate the client. The client must respond with the correct responses to the challenges presented by the framework. The number of exchanges is dependent on the policies of each side. The authentication of the client is deemed successful when the authenticationSucceeded method is invoked by the Framework.

The invocation of this method may be interleaved with challenge() calls by the client on the IpAPILevelAuthentication interface. The client shall respond immediately to authentication challenges from the Framework, and not wait until the Framework has responded to any challenge the client may issue.

This method shall only be used when the method initiateAuthenticationWithVersion() is used on the IpInitial interface.

Returns <response> : This is the response of the client application to the challenge of the framework in the current sequence. The formatting and construction of this parameter shall be according to section 4.1 of RFC 1994. A complete CHAP Response packet shall be used to carry the response ~~string~~octet set. That octet set will be the result of applying the designated hashing algorithm, which is indicated via the client's invocation of selectAuthenticationMechanism(), to an octet set consisting of the concatenation of the CHAP Identifier, the shared "secret", and the supplied challenge value. The Name field of the CHAP Response packet must be present and contain a valid value in order for the CHAP Response to be valid. However, the Name field is not used in the authentication process.~~The Response packet shall make the contents of this returned parameter. The Name field of the CHAP Response packet shall be present but not contain any useful value.~~

Parameters

`challenge : in TpOctetSet`

The challenge presented by the framework to be responded to by the client. The challenge format used will be in accordance with the IETF PPP Authentication Protocols - Challenge Handshake Authentication Protocol (RFC 1994).

The challenge octet set must be formatted as a CHAP Challenge packet as defined in section 4.1 of RFC 1994. A complete and properly formatted CHAP Challenge packet must be used. The Name field of the CHAP Challenge packet must be present and contain a valid value in order for the CHAP Response to be valid. However, the Name field is not used in the authentication process.~~The formatting of the challenge value shall be according to section 4.1 of RFC 1994. A complete CHAP Request packet shall be used to carry the challenge value. The Name field of the CHAP Request packet shall be present but not contain any useful value.~~

Steps for constructing the challenge octet set:

1. Create a random challenge value (octet set). Per RFC 1994, this value must between 1 and 255 octets in length.
2. Construct a CHAP Challenge packet based on 4.1 of RFC 1994 with the octet set from the previous step passed in the Value field within the CHAP Challenge.

Returns

`TpOctetSet`

Steps for constructing the response octet set:

1. Extract the Identifier and Value fields from the CHAP Challenge packet passed in the challenge() method's challenge parameter
2. Build an octet set consisting of the concatenation of the Identifier, the "shared secret", and the Value from the CHAP Challenge
3. Compute the hash of the octet set resulting from the previous step using the designated hashing algorithm
4. Construct a complete CHAP Response packet with the resulting octet set from previous step as the CHAP Value

Steps for validating the response octet set:

1. Verify that the Identifier sent in the original CHAP Challenge matches the Identifier received in the CHAP Response. If it does not, authentication fails.

2. Build an octet set consisting of the concatenation of the original Identifier, the "shared secret", and the original challenge value

3. Compute the hash of the resulting octet set from the previous step using the designated hashing algorithm

4. Verify the octet set resulting from the previous step matches the octet set contained in the Value field of the CHAP Response. A match indicates successful authentication.

End of change in Clause 6.3.1.1.4

Change in Clause 6.3.1.5.6

6.3.1.5.6 Method <<new>> challenge()

This method is used by the client to authenticate the framework. The framework must respond with the correct responses to the challenges presented by the client. The domainID received in the initiateAuthenticationWithVersion() 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 authentication of the framework is deemed successful when the authenticationSucceeded method is invoked by the client.

The invocation of this method may be interleaved with challenge() calls by the framework on the client's APILevelAuthentication interface.

This method shall only be used when the IpAPILevelAuthentication interface is obtained by using initiateAuthenticationWithVersion() on the IpInitial interface.

Returns <response> : This is the response of the framework to the challenge of the client in the current sequence. The formatting and construction of this parameter shall be according to section 4.1 of RFC 1994. A complete CHAP Response packet shall be used to carry the response ~~string~~octet set. That octet set will be the result of applying the designated hashing algorithm, which is indicated via the client's invocation of selectAuthenticationMechanism(), to an octet set consisting of the concatenation of the CHAP Identifier, the shared "secret", and the supplied challenge value. The Name field of the CHAP Response packet must be present and contain a valid value in order for the CHAP Response to be valid. However, the Name field is not used in the authentication process.~~The Response packet shall make the contents of this returned parameter. The Name field of the CHAP Response packet shall be present but not contain any useful value.~~

Parameters

`challenge : in TpOctetSet`

The challenge presented by the client to be responded to by the framework. The challenge format used will be in accordance with the IETF PPP Authentication Protocols - Challenge Handshake Authentication Protocol (RFC 1994).

The challenge octet set must be formatted as a CHAP Challenge packet as defined in section 4.1 of RFC 1994. A complete and properly formatted CHAP Challenge packet must be used. The Name field of the CHAP Challenge packet must be present and contain a valid value in order for the CHAP Response to be valid. However, the Name field is not used in the authentication process.~~The formatting of the challenge value shall be according to section 4.1 of RFC 1994. A complete CHAP Request packet shall be used to carry the challenge value. The Name field of the CHAP Request packet shall be present but not contain any useful value.~~

Steps for constructing the challenge octet set:

1. Create a random challenge value (octet set). Per RFC 1994, this value must between 1 and 255 octets in length.

2. Construct a CHAP Challenge packet based on 4.1 of RFC 1994 with the octet set from the previous step passed in the Value field within the CHAP Challenge.

Returns

TpOctetSet

Steps for constructing the response octet set:

1. Extract the Identifier and Value fields from the CHAP Challenge packet passed in the challenge() method's challenge parameter
2. Build an octet set consisting of the concatenation of the Identifier, the "shared secret", and the Value from the CHAP Challenge
3. Compute the hash of the octet set resulting from the previous step using the designated hashing algorithm
4. Construct a complete CHAP Response packet with the resulting octet set from previous step as the CHAP Value

Steps for validating the response octet set:

1. Verify that the Identifier sent in the original CHAP Challenge matches the Identifier received in the CHAP Response. If it does not, authentication fails.
2. Build an octet set consisting of the concatenation of the original Identifier, the "shared secret", and the original challenge value
3. Compute the hash of the resulting octet set from the previous step using the designated hashing algorithm

Verify the octet set resulting from the previous step matches the octet set contained in the Value field of the CHAP Response. A match indicates successful authentication.

Raises

TpCommonExceptions, P_ACCESS_DENIED

End of change in Clause 6.3.1.5.6

Annex E (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Mar 2001	CN_11	NP-010134	047	--	CR 29.198: for moving TS 29.198 from R99 to Rel 4 (N5-010158)	3.2.0	4.0.0
Jun 2001	CN_12	NP-010330	001	--	Corrections to OSA API Rel4	4.0.0	4.0.1
Sep 2001	CN_13	NP-010466	002	--	Changing references to JAIN	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	003	--	Update to the definitions of method svcUnavailableInd	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	004	--	Only one subject per method invocation for fault and load management	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	005	--	Fault management is missing some *Err methods	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	006	--	Method balance on Fault management interfaces	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	007	--	Change "TpString" into "TpOctetSets" in authentication and access	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	008	--	Replacement of register/unregisterLoadController	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	009	--	Redundant Framework Heartbeat Mechanism	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	010	--	Add a releaseInterface() method to IpAccess	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	011	--	Removal of serviceID from queryAppLoadReq()	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	012	--	Addition of listInterfaces() method	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	013	--	Introduction and use of new Service Instance ID	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	014	--	P_UNAUTHORISED_PARAMETER_VALUE thrown if non-accessible serviceID is provided	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	015	--	Introduction of Service Instance Lifecycle Management	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	016	--	Add support for multi-vendorship	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	017	--	Removal of P_SERVICE_ACCESS_TYPE	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	018	--	Confusing meaning of prescribedMethod	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	019	--	A client should only have one instance of a given service	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	020	--	Some methods on the IpApp interfaces should throw exceptions	4.1.0	4.2.0
Dec 2001	CN_14	NP-010596	021	--	Replace Out Parameters with Return Types	4.2.0	4.3.0
Dec 2001	CN_14	NP-010596	022	--	Correctionto Framework (FW)	4.2.0	4.3.0
Mar 2002	CN_15	NP-020105	023	--	Add P_INVALID_INTERFACE_TYPE exception to IpService.setCallback() and IpService.setCallbackWithSessionID()	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	024	--	Replace erroneous mention of P_OSA_ACCESS by the correct value P_OSA_AUTHENTICATION	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	025	--	Add missing inheritance in service agreement management interfaces	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	026	--	Include Operation Set as part of General Service Properties	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	027	--	Improved description of activityTestReq with respect to ServiceInstanceID	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	028	--	OSA Framework - Generate statistics records on behalf of another entity using genFaultStatsRecordReq	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	029	--	Update the interface names for alignment between 3GPP and ETSI/Parlay	4.3.0	4.4.0
Jun 2002	CN_16	NP-020179	030	--	Solving the problem in the OSA Framework with method appUnavailableInd() in a scenario with multiple service sessions per access session	4.4.0	4.5.0
Jun 2002	CN_16	NP-020179	031	--	Adding missing mandatory method (authenticationSucceeded) to sequence flow	4.4.0	4.5.0
Jun 2002	CN_16	NP-020186	032	--	Remove redundant data type definition TpServiceSpecString	4.5.0	5.0.0
Jun 2002	CN_16	NP-020181	033	--	Addition of support for Java API technology realisation	4.5.0	5.0.0
Jun 2002	CN_16	NP-020182	035	--	Addition of support for WSDL realisation	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	036	--	Clarify semantics of service properties of type BOOLEAN_SET	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	037	--	Addition of version management support to the Framework (29.198-03) in run-time	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	038	--	Enhancements on subscription management error information	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	039	--	Delete conflicting description of P_APPLICATION_NOT_ACTIVATED	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	040	--	Note added for P_SERVICE_INSTANCE Choice Element Name	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	041	--	Correcting the method descriptions for abortAuthentication and for initiateAuthentication	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	042	--	Correcting the description of heartbeat failure	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	043	--	Correcting erroneous FW<->Service instance sequence diagrams	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	044	--	Correcting the scope of TpFwID, which currently is giving it false limitations	4.5.0	5.0.0
Sep 2002	CN_17	NP-020428	046		Correction to description of TpServicePropertyTypeName	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	047		Remove undefined exception in registerService	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	048		Remove ServiceIDs from IpFwFaultManager.genFaultStatsRecordReq()	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	049		Correct appUnavailableInd and related methods	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	050		Remove unusable exception from IpFaultManager.appActivityTestRes()	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	051		Clarify the sequence of events in signing the service agreement	5.0.0	5.1.0

Sep 2002	CN_17	NP-020428	052		Correct use of electronic signatures	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	053		Addition of Sequence Diagrams for terminateAccess	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	054		Add indication what part of service agreement must be signed	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	055		Add text to clarify requirements on support of methods	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	056		Introduce types and modes for generic properties	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	057		Correction on use of NULL in Framework API	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	058		Add Negotiation of Authentication Mechanism for OSA level Authentication	5.0.0	5.1.0
Sep 2002	CN_17	NP-020395	058		Add text to clarify relationship between 3GPP and ETSI/Parlay OSA specifications	5.0.0	5.1.0
Mar 2003	CN_19	NP-030019	063	-	Correction to Initial Access Sequence Diagram	5.1.0	5.2.0
Mar 2003	CN_19	NP-030019	065	-	Enable creation/destruction of load level notifications at the request of Framework	5.1.0	5.2.0
Mar 2003	CN_19	NP-030019	067	-	Correction of Sequence for Framework – Service load management	5.1.0	5.2.0
Mar 2003	CN_19	NP-030019	074	-	Add Initial Load Notification report for Framework Integrity Management Load Notification model	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	068	--	Correction to Application's requirements for supporting methods	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	069	--	Correction of status of methods to interfaces in clause 7.3	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	070	--	Correction of status of methods to interfaces in clause 8.3	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	071	--	Correction of status of methods to interfaces in clause 6.3	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	075	--	Adding the appAvailStatusInd() and svcAvailStatusInd() methods	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	076	--	Remove race condition in signServiceAgreement	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	077	--	Change reference to deprecated method "authenticate" in TpAuthMechanism to "challenge"	5.1.0	5.2.0
Jun 2003	CN_20	NP-030237	079	--	Correction to TpEncryptionCapability to correct support for Triple-DES	5.2.0	5.3.0
Jun 2003	CN_20	NP-030237	081	--	Correction of the Framework Service Instance Lifecycle Manager Sequence Diagram	5.2.0	5.3.0
Jun 2003	CN_20	NP-030237	083	--	Correction of the use of TpDomainID in Framework initiateAuthentication method	5.2.0	5.3.0
Sep 2003	CN_21	NP-030352	085	--	Correction to Java Realisation Annex	5.3.0	5.4.0
Dec 2003	CN_22	NP-030549	086	--	Correction of the sequence diagram for Fault Management	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	087	--	Correction of State Transition Diagram for IpAccess	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	088	--	Correction of Correlation Behaviour in Load Management	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	089	--	Correction of Correlation Behaviour in Fault Management	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	090	--	Correction and Clarification of Framework Access Session Behaviour	5.4.0	5.5.0
Dec 2003	CN_22	NP-030553	091	--	Add OSA API support for 3GPP2 networks	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	092	--	Add description for service super and sub types	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	093	--	Add support for registration of additional service property types and modes	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	094	--	Improve User Interaction message management functions	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	095	--	Add new values for TpServiceTypeName for Policy Management	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	096	--	Allow for applications to re-obtain the reference to the service manager	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	097	--	Add support in OSA to inform applications about new SCSs and their level of Backward compatibility – Align with SA1's 22.127	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	098	--	Add “Extended User Status” as service type name - Align with 29.198-06	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	099	--	Add P_USER_BINDING to TpServiceTypeName	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	100	--	Modify Framework Availability Indication in Fault Management	5.5.0	6.0.0
Feb 2004	--	--	--	--	Added Java code attachment 2919803J2EE.zip which was delivered late by outside developers. See Annex C.	6.0.0	6.0.1

CR-Form-v7

CHANGE REQUEST

⌘ **29.198-03** **CR** **120** ⌘ **rev** **-** ⌘ Current version: **5.6.0** ⌘

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

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network **X**

Title:	⌘	Correct TpSignatureAndServiceMgr to align with description in signServiceAgreement
Source:	⌘	CN5 Lucent Technologies
Work item code:	⌘	OSA2 ***Date:*** ⌘ 18/05/2004
Category:	⌘	**F** ***Release:*** ⌘ Rel-5

Use one 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 one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change:	⌘	The TpSignatureAndServiceMgr type description is incorrect and does not align with the signServiceAgreement description causing a discrepancy over the contents of the digital signature.
Summary of change:	⌘	The TpSignatureAndServiceMgr description has been corrected to align with the description in signServiceAgreement().
Consequences if not approved:	⌘	The contents of the digital signature returned in signServiceAgreement may vary depending on which section of the specification is used resulting in operational failures.

Clauses affected:	⌘	10.3.10

Other specs affected: ⌘	Y	N		⌘
		X	Other core specifications	
		X	Test specifications	
		X	O&M Specifications	

Other comments:	⌘	Mirror CR for Rel-6 in N5-040356

Change in Clause 10.3.10

10.3.10 TpSignatureAndServiceMgr

This is a Sequence of Data Elements containing the digital signature of the Framework for the service agreement, and a reference to the SCF manager interface of the SCF.

Sequence Element Name	Sequence Element Type
DigitalSignature	TpOctetSet
ServiceMgrInterface	IpServiceRef

The digitalSignature contains a CMS (Cryptographic Message Syntax) object (as defined in RFC 2630) with content type Signed-data. The signature is calculated and created as per section 5 of RFC 2630. The content is the agreement text given by the client application. The "external signature" construct shall not be used (i.e. the eContent field in the EncapsulatedContentInfo field shall be present and contain the agreement text string). The signing-time attribute, as defined in section 11.3 of RFC 2630, shall also be used to provide replay prevention. ~~The digitalSignature is the signed version of a hash of the service token and agreement text given by the client application.~~

The ServiceMgrInterface is a reference to the SCF manager interface for the selected SCF.

End of change in Clause 10.3.10

Annex D (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Mar 2001	CN_11	NP-010134	047	--	CR 29.198: for moving TS 29.198 from R99 to Rel 4 (N5-010158)	3.2.0	4.0.0
Jun 2001	CN_12	NP-010330	001	--	Corrections to OSA API Rel4	4.0.0	4.0.1
Sep 2001	CN_13	NP-010466	002	--	Changing references to JAIN	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	003	--	Update to the definitions of method svcUnavailableInd	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	004	--	Only one subject per method invocation for fault and load management	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	005	--	Fault management is missing some *Err methods	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	006	--	Method balance on Fault management interfaces	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	007	--	Change "TpString" into "TpOctetSets" in authentication and access	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	008	--	Replacement of register/unregisterLoadController	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	009	--	Redundant Framework Heartbeat Mechanism	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	010	--	Add a releaseInterface() method to IpAccess	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	011	--	Removal of serviceID from queryAppLoadReq()	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	012	--	Addition of listInterfaces() method	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	013	--	Introduction and use of new Service Instance ID	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	014	--	P_UNAUTHORISED_PARAMETER_VALUE thrown if non-accessible serviceID is provided	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	015	--	Introduction of Service Instance Lifecycle Management	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	016	--	Add support for multi-vendorship	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	017	--	Removal of P_SERVICE_ACCESS_TYPE	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	018	--	Confusing meaning of prescribedMethod	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	019	--	A client should only have one instance of a given service	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	020	--	Some methods on the IpApp interfaces should throw exceptions	4.1.0	4.2.0
Dec 2001	CN_14	NP-010596	021	--	Replace Out Parameters with Return Types	4.2.0	4.3.0
Dec 2001	CN_14	NP-010596	022	--	Correctionto Framework (FW)	4.2.0	4.3.0
Mar 2002	CN_15	NP-020105	023	--	Add P_INVALID_INTERFACE_TYPE exception to IpService.setCallback() and IpService.setCallbackWithSessionID()	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	024	--	Replace erroneous mention of P_OSA_ACCESS by the correct value P_OSA_AUTHENTICATION	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	025	--	Add missing inheritance in service agreement management interfaces	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	026	--	Include Operation Set as part of General Service Properties	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	027	--	Improved description of activityTestReq with respect to ServiceInstanceID	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	028	--	OSA Framework - Generate statistics records on behalf of another entity using genFaultStatsRecordReq	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	029	--	Update the interface names for alignment between 3GPP and ETSI/Parlay	4.3.0	4.4.0
Jun 2002	CN_16	NP-020179	030	--	Solving the problem in the OSA Framework with method appUnavailableInd() in a scenario with multiple service sessions per access session	4.4.0	4.5.0
Jun 2002	CN_16	NP-020179	031	--	Adding missing mandatory method (authenticationSucceeded) to sequence flow	4.4.0	4.5.0
Jun 2002	CN_16	NP-020186	032	--	Remove redundant data type definition TpServiceSpecString	4.5.0	5.0.0
Jun 2002	CN_16	NP-020181	033	--	Addition of support for Java API technology realisation	4.5.0	5.0.0
Jun 2002	CN_16	NP-020182	035	--	Addition of support for WSDL realisation	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	036	--	Clarify semantics of service properties of type BOOLEAN_SET	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	037	--	Addition of version management support to the Framework (29.198-03) in run-time	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	038	--	Enhancements on subscription management error information	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	039	--	Delete conflicting description of P_APPLICATION_NOT_ACTIVATED	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	040	--	Note added for P_SERVICE_INSTANCE Choice Element Name	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	041	--	Correcting the method descriptions for abortAuthentication and for initiateAuthentication	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	042	--	Correcting the description of heartbeat failure	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	043	--	Correcting erroneous FW<->Service instance sequence diagrams	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	044	--	Correcting the scope of TpFwID, which currently is giving it false limitations	4.5.0	5.0.0
Sep 2002	CN_17	NP-020428	046		Correction to description of TpServicePropertyTypeName	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	047		Remove undefined exception in registerService	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	048		Remove ServiceIDs from IpFwFaultManager.genFaultStatsRecordReq()	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	049		Correct appUnavailableInd and related methods	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	050		Remove unusable exception from IpFaultManager.appActivityTestRes()	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	051		Clarify the sequence of events in signing the service agreement	5.0.0	5.1.0

Sep 2002	CN_17	NP-020428	052		Correct use of electronic signatures	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	053		Addition of Sequence Diagrams for terminateAccess	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	054		Add indication what part of service agreement must be signed	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	055		Add text to clarify requirements on support of methods	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	056		Introduce types and modes for generic properties	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	057		Correction on use of NULL in Framework API	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	058		Add Negotiation of Authentication Mechanism for OSA level Authentication	5.0.0	5.1.0
Sep 2002	CN_17	NP-020395	058		Add text to clarify relationship between 3GPP and ETSI/Parlay OSA specifications	5.0.0	5.1.0
Mar 2003	CN_19	NP-030019	063	-	Correction to Initial Access Sequence Diagram	5.1.0	5.2.0
Mar 2003	CN_19	NP-030019	065	-	Enable creation/destruction of load level notifications at the request of Framework	5.1.0	5.2.0
Mar 2003	CN_19	NP-030019	067	-	Correction of Sequence for Framework – Service load management	5.1.0	5.2.0
Mar 2003	CN_19	NP-030019	074	-	Add Initial Load Notification report for Framework Integrity Management Load Notification model	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	068	--	Correction to Application's requirements for supporting methods	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	069	--	Correction of status of methods to interfaces in clause 7.3	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	070	--	Correction of status of methods to interfaces in clause 8.3	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	071	--	Correction of status of methods to interfaces in clause 6.3	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	075	--	Adding the appAvailStatusInd() and svcAvailStatusInd() methods	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	076	--	Remove race condition in signServiceAgreement	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	077	--	Change reference to deprecated method "authenticate" in TpAuthMechanism to "challenge"	5.1.0	5.2.0
Jun 2003	CN_20	NP-030237	079	--	Correction to TpEncryptionCapability to correct support for Triple-DES	5.2.0	5.3.0
Jun 2003	CN_20	NP-030237	081	--	Correction of the Framework Service Instance Lifecycle Manager Sequence Diagram	5.2.0	5.3.0
Jun 2003	CN_20	NP-030237	083	--	Correction of the use of TpDomainID in Framework initiateAuthentication method	5.2.0	5.3.0
Sep 2003	CN_21	NP-030352	085	--	Correction to Java Realisation Annex	5.3.0	5.4.0
Dec 2003	CN_22	NP-030549	086	--	Correction of the sequence diagram for Fault Management	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	087	--	Correction of State Transition Diagram for IpAccess	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	088	--	Correction of Correlation Behaviour in Load Management	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	089	--	Correction of Correlation Behaviour in Fault Management	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	090	--	Correction and Clarification of Framework Access Session Behaviour	5.4.0	5.5.0
Apr 2004	CN_23bis	NP-040155	101	--	Correct Java Code to conform with Java Rulebook in TS 29.198-01 and to remove errors	5.5.0	5.6.0

CR-Form-v7

CHANGE REQUEST

⌘ **29.198-03** **CR** **121** ⌘ **rev** **-** ⌘ Current version: **6.0.1** ⌘

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

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network ☒

Title:	⌘	Correct TpSignatureAndServiceMgr to align with description in signServiceAgreement
Source:	⌘	CN5 Lucent Technologies
Work item code:	⌘	OSA2
Date:	⌘	18/05/2004
Category:	⌘	**A**
Release:	⌘	Rel-6

Use one 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 one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change:	⌘	The TpSignatureAndServiceMgr type description is incorrect and does not align with the signServiceAgreement description causing a discrepancy over the contents of the digital signature.
Summary of change:	⌘	The TpSignatureAndServiceMgr description has been corrected to align with the description in signServiceAgreement().
Consequences if not approved:	⌘	The contents of the digital signature returned in signServiceAgreement may vary depending on which section of the specification is used resulting in operational failures.

Clauses affected:	⌘	10.3.10

Other specs affected: ⌘	Y	N		⌘
		X	Other core specifications	
		X	Test specifications	
		X	O&M Specifications	

Other comments:	⌘	Mirror CR for Rel-5 in N5-040355

Change in Clause 10.3.10

10.3.10 TpSignatureAndServiceMgr

This is a Sequence of Data Elements containing the digital signature of the Framework for the service agreement, and a reference to the SCF manager interface of the SCF.

Sequence Element Name	Sequence Element Type
DigitalSignature	TpOctetSet
ServiceMgrInterface	IpServiceRef

The digitalSignature contains a CMS (Cryptographic Message Syntax) object (as defined in RFC 2630) with content type Signed-data. The signature is calculated and created as per section 5 of RFC 2630. The content is the agreement text given by the client application. The "external signature" construct shall not be used (i.e. the eContent field in the EncapsulatedContentInfo field shall be present and contain the agreement text string). The signing-time attribute, as defined in section 11.3 of RFC 2630, shall also be used to provide replay prevention. ~~The digitalSignature is the signed version of a hash of the service token and agreement text given by the client application.~~

The ServiceMgrInterface is a reference to the SCF manager interface for the selected SCF.

End of change in Clause 10.3.10

Annex E (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Mar 2001	CN_11	NP-010134	047	--	CR 29.198: for moving TS 29.198 from R99 to Rel 4 (N5-010158)	3.2.0	4.0.0
Jun 2001	CN_12	NP-010330	001	--	Corrections to OSA API Rel4	4.0.0	4.0.1
Sep 2001	CN_13	NP-010466	002	--	Changing references to JAIN	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	003	--	Update to the definitions of method svcUnavailableInd	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	004	--	Only one subject per method invocation for fault and load management	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	005	--	Fault management is missing some *Err methods	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	006	--	Method balance on Fault management interfaces	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	007	--	Change "TpString" into "TpOctetSets" in authentication and access	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	008	--	Replacement of register/unregisterLoadController	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	009	--	Redundant Framework Heartbeat Mechanism	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	010	--	Add a releaseInterface() method to IpAccess	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	011	--	Removal of serviceID from queryAppLoadReq()	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	012	--	Addition of listInterfaces() method	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	013	--	Introduction and use of new Service Instance ID	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	014	--	P_UNAUTHORISED_PARAMETER_VALUE thrown if non-accessible serviceID is provided	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	015	--	Introduction of Service Instance Lifecycle Management	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	016	--	Add support for multi-vendorship	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	017	--	Removal of P_SERVICE_ACCESS_TYPE	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	018	--	Confusing meaning of prescribedMethod	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	019	--	A client should only have one instance of a given service	4.1.0	4.2.0
Sep 2001	CN_13	NP-010466	020	--	Some methods on the IpApp interfaces should throw exceptions	4.1.0	4.2.0
Dec 2001	CN_14	NP-010596	021	--	Replace Out Parameters with Return Types	4.2.0	4.3.0
Dec 2001	CN_14	NP-010596	022	--	Correctionto Framework (FW)	4.2.0	4.3.0
Mar 2002	CN_15	NP-020105	023	--	Add P_INVALID_INTERFACE_TYPE exception to IpService.setCallback() and IpService.setCallbackWithSessionID()	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	024	--	Replace erroneous mention of P_OSA_ACCESS by the correct value P_OSA_AUTHENTICATION	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	025	--	Add missing inheritance in service agreement management interfaces	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	026	--	Include Operation Set as part of General Service Properties	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	027	--	Improved description of activityTestReq with respect to ServiceInstanceID	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	028	--	OSA Framework - Generate statistics records on behalf of another entity using genFaultStatsRecordReq	4.3.0	4.4.0
Mar 2002	CN_15	NP-020105	029	--	Update the interface names for alignment between 3GPP and ETSI/Parlay	4.3.0	4.4.0
Jun 2002	CN_16	NP-020179	030	--	Solving the problem in the OSA Framework with method appUnavailableInd() in a scenario with multiple service sessions per access session	4.4.0	4.5.0
Jun 2002	CN_16	NP-020179	031	--	Adding missing mandatory method (authenticationSucceeded) to sequence flow	4.4.0	4.5.0
Jun 2002	CN_16	NP-020186	032	--	Remove redundant data type definition TpServiceSpecString	4.5.0	5.0.0
Jun 2002	CN_16	NP-020181	033	--	Addition of support for Java API technology realisation	4.5.0	5.0.0
Jun 2002	CN_16	NP-020182	035	--	Addition of support for WSDL realisation	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	036	--	Clarify semantics of service properties of type BOOLEAN_SET	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	037	--	Addition of version management support to the Framework (29.198-03) in run-time	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	038	--	Enhancements on subscription management error information	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	039	--	Delete conflicting description of P_APPLICATION_NOT_ACTIVATED	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	040	--	Note added for P_SERVICE_INSTANCE Choice Element Name	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	041	--	Correcting the method descriptions for abortAuthentication and for initiateAuthentication	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	042	--	Correcting the description of heartbeat failure	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	043	--	Correcting erroneous FW<->Service instance sequence diagrams	4.5.0	5.0.0
Jun 2002	CN_16	NP-020186	044	--	Correcting the scope of TpFwID, which currently is giving it false limitations	4.5.0	5.0.0
Sep 2002	CN_17	NP-020428	046		Correction to description of TpServicePropertyTypeName	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	047		Remove undefined exception in registerService	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	048		Remove ServiceIDs from IpFwFaultManager.genFaultStatsRecordReq()	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	049		Correct appUnavailableInd and related methods	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	050		Remove unusable exception from IpFaultManager.appActivityTestRes()	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	051		Clarify the sequence of events in signing the service agreement	5.0.0	5.1.0

Sep 2002	CN_17	NP-020428	052		Correct use of electronic signatures	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	053		Addition of Sequence Diagrams for terminateAccess	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	054		Add indication what part of service agreement must be signed	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	055		Add text to clarify requirements on support of methods	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	056		Introduce types and modes for generic properties	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	057		Correction on use of NULL in Framework API	5.0.0	5.1.0
Sep 2002	CN_17	NP-020428	058		Add Negotiation of Authentication Mechanism for OSA level Authentication	5.0.0	5.1.0
Sep 2002	CN_17	NP-020395	058		Add text to clarify relationship between 3GPP and ETSI/Parlay OSA specifications	5.0.0	5.1.0
Mar 2003	CN_19	NP-030019	063	-	Correction to Initial Access Sequence Diagram	5.1.0	5.2.0
Mar 2003	CN_19	NP-030019	065	-	Enable creation/destruction of load level notifications at the request of Framework	5.1.0	5.2.0
Mar 2003	CN_19	NP-030019	067	-	Correction of Sequence for Framework – Service load management	5.1.0	5.2.0
Mar 2003	CN_19	NP-030019	074	-	Add Initial Load Notification report for Framework Integrity Management Load Notification model	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	068	--	Correction to Application's requirements for supporting methods	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	069	--	Correction of status of methods to interfaces in clause 7.3	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	070	--	Correction of status of methods to interfaces in clause 8.3	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	071	--	Correction of status of methods to interfaces in clause 6.3	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	075	--	Adding the appAvailStatusInd() and svcAvailStatusInd() methods	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	076	--	Remove race condition in signServiceAgreement	5.1.0	5.2.0
Mar 2003	CN_19	NP-030028	077	--	Change reference to deprecated method "authenticate" in TpAuthMechanism to "challenge"	5.1.0	5.2.0
Jun 2003	CN_20	NP-030237	079	--	Correction to TpEncryptionCapability to correct support for Triple-DES	5.2.0	5.3.0
Jun 2003	CN_20	NP-030237	081	--	Correction of the Framework Service Instance Lifecycle Manager Sequence Diagram	5.2.0	5.3.0
Jun 2003	CN_20	NP-030237	083	--	Correction of the use of TpDomainID in Framework initiateAuthentication method	5.2.0	5.3.0
Sep 2003	CN_21	NP-030352	085	--	Correction to Java Realisation Annex	5.3.0	5.4.0
Dec 2003	CN_22	NP-030549	086	--	Correction of the sequence diagram for Fault Management	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	087	--	Correction of State Transition Diagram for IpAccess	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	088	--	Correction of Correlation Behaviour in Load Management	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	089	--	Correction of Correlation Behaviour in Fault Management	5.4.0	5.5.0
Dec 2003	CN_22	NP-030549	090	--	Correction and Clarification of Framework Access Session Behaviour	5.4.0	5.5.0
Dec 2003	CN_22	NP-030553	091	--	Add OSA API support for 3GPP2 networks	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	092	--	Add description for service super and sub types	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	093	--	Add support for registration of additional service property types and modes	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	094	--	Improve User Interaction message management functions	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	095	--	Add new values for TpServiceTypeName for Policy Management	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	096	--	Allow for applications to re-obtain the reference to the service manager	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	097	--	Add support in OSA to inform applications about new SCSs and their level of Backward compatibility – Align with SA1's 22.127	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	098	--	Add “Extended User Status” as service type name - Align with 29.198-06	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	099	--	Add P_USER_BINDING to TpServiceTypeName	5.5.0	6.0.0
Dec 2003	CN_22	NP-030554	100	--	Modify Framework Availability Indication in Fault Management	5.5.0	6.0.0
Feb 2004	--	--	--	--	Added Java code attachment 2919803J2EE.zip which was delivered late by outside developers. See Annex C.	6.0.0	6.0.1