3GPP TSG-CN Meeting #24 **NP-040254**
02 – 04 June 2004, Seoul, KOREA

Source: **CN5 (OSA)**

Title: **3 Rel-4 CRs 29.198-03 OSA API Part 3: Framework (Correct Address Range service property type)**

Agenda item: **7.10 (OSA Enhancements [OSA1])**

Document for: **APPROVAL**

Doc-1st-	Spec	CR	Rev	Phase	Subject	Cat	Version	Doc-2nd-	Workite
NP-040254	29.198-03	110	-	Rel-4	Correct the service property type used for address ranges	F	4.8.0	N5-040249	OSA1
NP-040254	29.198-03	111	-	Rel-5	Correct the service property type used for address ranges	A	5.6.0	N5-040250	OSA1
NP-040254	29.198-03	112	-	Rel-6	Correct the service property type used for address ranges	A	6.0.1	N5-040251	OSA1

CR-Form-v7

CHANGE REQUEST

⌘ **29.198-03** **CR** **110** ⌘ **rev** **-** ⌘ Current version: **4.8.0** ⌘

*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 service property type used for address ranges
Source:	⌘	CN5 Ultan Mulligan, ETSI PTCC
Work item code:	⌘	OSA1
Date:	⌘	14/05/2004
Category:	⌘	**F**
Release:	⌘	*REL-4*

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 Service Property Type ADDRESSRANGE_SET is used to identify the sets of address ranges for which an application can request notifications. At present it is simply defined as a set of addresses, with wildcards permitted. The Address Plan within which these addresses are defined is missing. There is no way to correlate the values of a service property of type ADDRESSRANGE_SET with the values of a service property identifying the address plans supported by an SCF. This is a particular problem when more than one address plan is supported by an SCF, and has resulted in interoperability issues, where different interpretations have been placed on the contents of service properties of this type.
Summary of change:	⌘	Introduce a new service property type XML_ADDRESS_RANGE_SET which is defined as a sequence of values of TpAddressRange, and therefore contains all the information necessary to uniquely identify address ranges, including the address plan. The service property type is formatted in XML. This is because most other formatting possibilities, using , ; : = etc. as delimiters, could cause confusion with various address types, where similar delimiters are also used. Also, use of XML is compatible with the basic type of all service properties: they are passed as strings. Deprecate the existing ADDRESSRANGE_SET service property type, as it is replaced by XML_ADDRESS_RANGE_SET. This ensures the correction is backwards compatible.
Consequences if not approved:	⌘	The interoperability problems encountered will continue, with different vendors adopting their own interpretation of the meaning of this service property type. These interoperability problems impact the interface between the Framework and an Application, and the interface between the Framework and the SCF.

Clauses affected:	⌘	9.1

Other specs affected:	⌘	Y	N		⌘	
		X		Other core specifications		29.198-04, -05, -08, -11
			X	Test specifications		
			X	O&M Specifications		

Other comments:	⌘	Mirror CRs to this CR exist for Rel-5 and Rel-6 in N5-040250 and N5-040251 respectively. Related Rel-4 CRs to TS 29.198-4, 5, 8, 11 are in N5-040252, N5-040257, N5-040260 and N5-040263

9 Service Properties

9.1 Service Property Types

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

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

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

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

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

Property type name	Description	Example value (array of strings)	Interpretation of example value
BOOLEAN_SET	set of Booleans	{"FALSE"}	The set of Booleans consisting of the Boolean "false".
INTEGER_SET	set of integers	{"1", "2", "5", "7"}	The set of integers consisting of the integers 1, 2, 5 and 7.
STRING_SET	set of strings	{"Sophia", "Rijen"}	The set of strings consisting of the string “Sophia" and the string "Rijen"
ADDRESSRANGE_SET (Deprecated)	set of address ranges	{"123??*", "*.ericsson.se"}	The set of address ranges consisting of ranges 123??* and *.ericsson.se.
INTEGER_INTERVAL	interval of integers	{"5", "100"}	The integers that are between or equal to 5 and 100.
STRING_INTERVAL	interval of strings	{"Rijen", "Sophia"}	The strings that are between or equal to the strings "Rijen" and "Sophia", in lexicographical order.
INTEGER_INTEGER_MAP	map from integers to integers	{"1", "10", "2", "20", "3", "30"}	The map that maps 1 to 10, 2 to 20 and 3 to 30.
XML_ADDRESS_RANGE_SET	set of values of TpAddressRange. Values of TpAddressRange are described using XML. An XML schema is provided below for this purpose.	{"<AddressRangeSet><AddressRange><Plan>P_ADDRESS_PLAN_E164</Plan><AddrString>123*</AddrString></AddressRange><AddressRange><Plan>P_ADDRESS_PLAN_E164</Plan><AddrString>234*</AddrString></AddressRange></AddressRangeSet>"}	Any addresses starting with 123 or starting with 456 in the E.164 Address Plan

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

The value of XML_ADDRESS_RANGE_SET should comply with the following XML Schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="AddressRangeSet">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="AddressRange" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Plan" type="xs:string" default="P_ADDRESS_PLAN_ANY"/>
                            <xs:element name="AddrString" type="xs:string"/>
                            <xs:element name="Name" type="xs:string" minOccurs="0"/>
                            <xs:element name="SubAddressString" type="xs:string" minOccurs="0"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

An example usage could be:

```
{"<?xml version="1.0" encoding="UTF-8"?>
<AddressRangeSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="xml_address_range_set.xsd">
    <AddressRange>
        <Plan>P_ADDRESS_PLAN_E164</Plan>
        <AddrString>789*</AddrString>
    </AddressRange>
    <AddressRange>
        <Plan>P_ADDRESS_PLAN_ANY</Plan>
        <AddrString>123*</AddrString>
    </AddressRange>
    <AddressRange>
        <Plan>P_ADDRESS_PLAN_SIP</Plan>
        <AddrString>&lt;sip:*@parlay.org&gt;</AddrString>
        <Name/>
    </AddressRange>
</AddressRangeSet>"}
```

Note that the final address range corresponds to any sip address @parlay.org, i.e. <sip:*@parlay.org>.

9.2 General Service Properties

Each service instance has the following general properties:

- Service Name
- Service Version
- Service Instance ID
- Service Instance Description
- Product Name
- Product Version

- Supported Interfaces
- Operation Set

9.2.1 Service Name

This property contains the name of the service, e.g. “UserLocation”, “UserLocationCamel”, “UserLocationEmergency” or “UserStatus”.

9.2.2 Service Version

This property contains the version of the APIs, to which the service is compliant, e.g. “2.1".

9.2.3 Service Instance ID

This property uniquely identifies a specific instance of the service. The Framework generates this property.

9.2.4 Service Instance Description

This property contains a textual description of the service.

9.2.5 Product Name

This property contains the name of the product that provides the service, e.g. “Find It”, “Locate.com”.

9.2.6 Product Version

This property contains the version of the product that provides the service, e.g. “3.1.11”.

9.2.7 Supported Interfaces

This property contains a list of strings with interface names that the service supports, e.g. “IpUserLocation”, “IpUserStatus”.

9.2.8 Operation Set

Property	Type	Description
P_OPERATION_SET	STRING_SET	Specifies set of the operations the SCS supports. The notation to be used is : {“Interface1.operation1”,”Interface1.operation2”, “Interface2.operation1”}, e.g.: {“IpCall.createCall”,”IpCall.routeReq”}.

Annex B (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
Sep 2002	CN_17	NP-020423	045	--	Correction on use of NULL in Framework API	4.5.0	4.6.0
Mar 2003	CN_19	NP-030019	060	--	Correction of status of methods to interfaces in clause 7.3	4.6.0	4.7.0
Mar 2003	CN_19	NP-030019	061	--	Correction of status of methods to interfaces in clause 8.3	4.6.0	4.7.0
Mar 2003	CN_19	NP-030019	062	--	Correction to Initial Access Sequence Diagram	4.6.0	4.7.0
Mar 2003	CN_19	NP-030019	064	--	Enable creation/destruction of load level notifications at the request of Framework	4.6.0	4.7.0
Mar 2003	CN_19	NP-030019	066	--	Correction of Sequence for Framework – Service load management	4.6.0	4.7.0
Mar 2003	CN_19	NP-030019	072	--	Correction of status of methods to interfaces in clause 6.3	4.6.0	4.7.0
Mar 2003	CN_19	NP-030019	073	--	Add Initial Load Notification report for Framework Integrity Management Load Notification model	4.6.0	4.7.0
Jun 2003	CN_20	NP-030237	078	--	Correction to TpEncryptionCapability to correct support for Triple-DES	4.7.0	4.8.0
Jun 2003	CN_20	NP-030237	080	--	Correction of the Framework Service Instance Lifecycle Manager Sequence Diagram	4.7.0	4.8.0
Jun 2003	CN_20	NP-030237	082	--	Correction of the use of TpDomainID in Framework initiateAuthentication method	4.7.0	4.8.0

CR-Form-v7

CHANGE REQUEST

⌘ 29.198-03 CR 111 ⌘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 service property type used for address ranges
Source:	⌘	CN5 Ultan Mulligan, ETSI PTCC
Work item code:	⌘	OSA1
Date:	⌘	14/05/2004
Category:	⌘	A
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 Service Property Type ADDRESSRANGE_SET is used to identify the sets of address ranges for which an application can request notifications. At present it is simply defined as a set of addresses, with wildcards permitted. The Address Plan within which these addresses are defined is missing. There is no way to correlate the values of a service property of type ADDRESSRANGE_SET with the values of a service property identifying the address plans supported by an SCF. This is a particular problem when more than one address plan is supported by an SCF, and has resulted in interoperability issues, where different interpretations have been placed on the contents of service properties of this type.
Summary of change:	⌘	Introduce a new service property type XML_ADDRESS_RANGE_SET which is defined as a sequence of values of TpAddressRange, and therefore contains all the information necessary to uniquely identify address ranges, including the address plan. The service property type is formatted in XML. This is because most other formatting possibilities, using , ; : = etc. as delimiters, could cause confusion with various address types, where similar delimiters are also used. Also, use of XML is compatible with the basic type of all service properties: they are passed as strings. Deprecate the existing ADDRESSRANGE_SET service property type, as it is replaced by XML_ADDRESS_RANGE_SET. This ensures the correction is backwards compatible.
Consequences if not approved:	⌘	The interoperability problems encountered will continue, with different vendors adopting their own interpretation of the meaning of this service property type. These interoperability problems impact the interface between the Framework and an Application, and the interface between the Framework and the SCF.

Clauses affected:	⌘	9.1

Other specs affected:	⌘	Y	N		⌘	
		X		Other core specifications		TS 29.198-04-2, -04-3, -05, -08, -11
			X	Test specifications		
			X	O&M Specifications		

Other comments:	⌘	This is a mirror CR to the Rel-4 CR in N5-040249. Related Rel-5 CRs to TS 29.198-4-2, -4-3, 5, 8, 11 are in N5-040253, N5-040255, N5-040258, N5-040261 and N5-040264

9 Service Properties

9.1 Service Property Types

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

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

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

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

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

Service Property type name	Description	Example value (array of strings)	Interpretation of example value
BOOLEAN_SET	set of Booleans	{"FALSE"}	The set of Booleans consisting of the Boolean "false".
INTEGER_SET	set of integers	{"1", "2", "5", "7"}	The set of integers consisting of the integers 1, 2, 5 and 7.
STRING_SET	set of strings	{"Sophia", "Rijen"}	The set of strings consisting of the string “Sophia" and the string "Rijen"
ADDRESSRANGE_SET (Deprecated)	set of address ranges	{"123??*", "*.ericsson.se"}	The set of address ranges consisting of ranges 123??* and *.ericsson.se.
INTEGER_INTERVAL	interval of integers	{"5", "100"}	The integers that are between or equal to 5 and 100.
STRING_INTERVAL	interval of strings	{"Rijen", "Sophia"}	The strings that are between or equal to the strings "Rijen" and "Sophia", in lexicographical order.
INTEGER_INTEGER_MAP	map from integers to integers	{"1", "10", "2", "20", "3", "30"}	The map that maps 1 to 10, 2 to 20 and 3 to 30.
XML_ADDRESS_RANGE_SET	set of values of TpAddressRange. Values of TpAddressRange are described using XML. An XML schema is provided below for this purpose.	{"<AddressRangeSet><AddressRange><Plan>P_ADDRESS_PLAN_E164</Plan><AddrString>123*</AddrString></AddressRange><AddressRange><Plan>P_ADDRESS_PLAN_E164</Plan><AddrString>234*</AddrString></AddressRange></AddressRangeSet>"}	Any addresses starting with 123 or starting with 456 in the E.164 Address Plan

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

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

The value of XML_ADDRESS_RANGE_SET should comply with the following XML Schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="AddressRangeSet">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="AddressRange" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Plan" type="xs:string" default="P_ADDRESS_PLAN_ANY"/>
                            <xs:element name="AddrString" type="xs:string"/>
                            <xs:element name="Name" type="xs:string" minOccurs="0"/>
                            <xs:element name="SubAddressString" type="xs:string" minOccurs="0"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
```

```
    </xs:element>
</xs:schema>
```

An example usage could be:

```
{"<?xml version="1.0" encoding="UTF-8"?>
<AddressRangeSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="xml_address_range_set.xsd">
    <AddressRange>
        <Plan>P_ADDRESS_PLAN_E164</Plan>
        <AddrString>789*</AddrString>
    </AddressRange>
    <AddressRange>
        <Plan>P_ADDRESS_PLAN_ANY</Plan>
        <AddrString>123*</AddrString>
    </AddressRange>
    <AddressRange>
        <Plan>P_ADDRESS_PLAN_SIP</Plan>
        <AddrString>&lt;sip:*@parlay.org&gt;</AddrString>
        <Name/>
    </AddressRange>
</AddressRangeSet>"}
```

Note that the final address range corresponds to any sip address @parlay.org, i.e. <sip:*@parlay.org>.

9.2 General Service Properties

Each service instance has the following general properties:

- Service Name
- Service Version
- Service Instance ID
- Service Instance Description
- Product Name
- Product Version
- Supported Interfaces
- Operation Set

The following sections describe these general service properties in more detail. The values for the mode are defined in the type TpServiceTypePropertyMode.

9.2.1 Service Name

Property	Type	Mode	Description
P_SERVICE_NAME	STRING_SET	MANDATORY_ READONLY	This property contains the name of the service, e.g. “UserLocation”, “UserLocationCamel”, “UserLocationEmergency” or “UserStatus”.

9.2.2 Service Version

Property	Type	Mode	Description
P_SERVICE_VERSION	STRING_SET	MANDATORY	This property contains the version of the APIs, to which the service is compliant. It is a set of strings as specified in the TpVersion type.

9.2.3 Service ID

Property	Type	Mode	Description
P_SERVICE_ID	STRING_INTERVAL	READONLY	This property uniquely identifies a specific service. Note that the Framework generates this property value when the Service Supplier registers the service. This property should not be confused with the serviceInstanceID generated by the Framework when a Client Application signs a Service Agreement to obtain the Service Manager

9.2.4 Service Description

Property	Type	Mode	Description
P_SERVICE_DESCRIPTION	STRING_SET	MANDATORY_ READONLY	This property contains a textual description of the service. It should not be interpreted as a description of a Service Instance (as identified by a serviceInstanceID generated by the Framework when a Client Application signs a Service Agreement to obtain the Service Manager).

9.2.5 Product Name

Property	Type	Mode	Description
P_PRODUCT_NAME	STRING_SET	READONLY	This property contains the name of the product that provides the service, e.g. “Find It”, “Locate.com”.

9.2.6 Product Version

Property	Type	Mode	Description
P_PRODUCT_VERSION	STRING_SET	READONLY	This property contains the version of the product that provides the service, e.g. “3.1.11”.

9.2.7 <<deprecated>> Supported Interfaces

This property contains a list of strings with interface names that the service supports, e.g. “IpUserLocation”, “IpUserStatus”. This property is deprecated and will be removed in a future version of the specification.

9.2.8 Operation Set

Property	Type	Mode	Description
P_OPERATION_SET	STRING_SET	MANDATORY	Specifies set of the operations the SCS supports. The notation to be used is : {“Interface1.operation1”,”Interface1.operation 2”, “Interface2.operation1”}, e.g.: {“IpCall.createCall”,”IpCall.routeReq”}.

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** **112** ⌘ **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 service property type used for address ranges
Source:	⌘	CN5 Ultan Mulligan, ETSI PTCC
Work item code:	⌘	OSA1
Date:	⌘	14/05/2004
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 Service Property Type ADDRESSRANGE_SET is used to identify the sets of address ranges for which an application can request notifications. At present it is simply defined as a set of addresses, with wildcards permitted. The Address Plan within which these addresses are defined is missing. There is no way to correlate the values of a service property of type ADDRESSRANGE_SET with the values of a service property identifying the address plans supported by an SCF. This is a particular problem when more than one address plan is supported by an SCF, and has resulted in interoperability issues, where different interpretations have been placed on the contents of service properties of this type.
Summary of change:	⌘	Introduce a new service property type XML_ADDRESS_RANGE_SET which is defined as a sequence of values of TpAddressRange, and therefore contains all the information necessary to uniquely identify address ranges, including the address plan. The service property type is formatted in XML. This is because most other formatting possibilities, using , ; : = etc. as delimiters, could cause confusion with various address types, where similar delimiters are also used. Also, use of XML is compatible with the basic type of all service properties: they are passed as strings. Deprecate the existing ADDRESSRANGE_SET service property type, as it is replaced by XML_ADDRESS_RANGE_SET. This ensures the correction is backwards compatible.
Consequences if not approved:	⌘	The interoperability problems encountered will continue, with different vendors adopting their own interpretation of the meaning of this service property type. These interoperability problems impact the interface between the Framework and an Application, and the interface between the Framework and the SCF.

Clauses affected:	⌘	9.1

Other specs affected:	⌘	Y	N		⌘	
		X		Other core specifications		TS 29.198-04, -05, -08, -11, 12
			X	Test specifications		
			X	O&M Specifications		

Other comments:	⌘	This is a mirror CR to the Rel-4 CR in N5-040249. Related Rel-6 CRs to TS 29.198-4-2, -4-3, 5, 8, 11 are in N5-040254, N5-040256, N5-040259, N5-040262 and N5-040265

9 Service Properties

9.1 Service Super and Sub Types

Service Properties are used at service registration to indicate the capabilities of an SCF. They are normally used as an indication for limitations an SCF has. These limitations can come from the way an SCF is implemented or from limitations in the network. The service type of an SCF defines which properties the supplier shall provide at registration of the SCF.

An application uses Service Properties at service discovery to find services that have the required capabilities. The Framework validates the requested properties with the registered properties and provides the application with a list of SCFs that comply to the application's request.

The capabilities of an SCF can be extended by providing service properties in addition to the ones defined in this standard. For this extended SCF, a dedicated sub-type of a service is defined. A sub-type of an SCF shall be fully compatible with the standard SCF, that is, an application shall be able to use the sub type as if it was the standard type. This implies that the interface to the SCF remains unchanged. Also SCF sub types can be further extended. This way a hierarchy of service types can be built with the standard type being the root.

An example of a sub type is a Multy Party Call Control service that allows the application to request a certain quality-of-service level. An additional service property is added for this.

9.2 Service Property Types

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

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

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

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

Service Property type name	Description	Example value (array of strings)	Interpretation of example value
BOOLEAN_SET	set of Booleans	{"FALSE"}	The set of Booleans consisting of the Boolean "false".
INTEGER_SET	set of integers	{"1", "2", "5", "7"}	The set of integers consisting of the integers 1, 2, 5 and 7.
STRING_SET	set of strings	{"Sophia", "Rijen"}	The set of strings consisting of the string “Sophia" and the string "Rijen"
ADDRESSRANGE_SET (Deprecated)	set of address ranges	{"123??*", "*.ericsson.se"}	The set of address ranges consisting of ranges 123??* and *.ericsson.se.
INTEGER_INTERVAL	interval of integers	{"5", "100"}	The integers that are between or equal to 5 and 100.
STRING_INTERVAL	interval of strings	{"Rijen", "Sophia"}	The strings that are between or equal to the strings "Rijen" and "Sophia", in lexicographical order.
INTEGER_INTEGER_MAP	map from integers to integers	{"1", "10", "2", "20", "3", "30"}	The map that maps 1 to 10, 2 to 20 and 3 to 30.
XML_ADDRESS_RANGE_SET	set of values of TpAddressRange. Values of TpAddressRange are described using XML. An XML schema is provided below for this purpose.	{"<AddressRangeSet><AddressRange><Plan>P_ADDRESS_PLAN_E164</Plan><AddrString>123*</AddrString></AddressRange><AddressRange><Plan>P_ADDRESS_PLAN_E164</Plan><AddrString>234*</AddrString></AddressRange></AddressRangeSet>"}	Any addresses starting with 123 or starting with 456 in the E.164 Address Plan

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

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

The value of XML_ADDRESS_RANGE_SET should comply with the following XML Schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="AddressRangeSet">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="AddressRange" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Plan" type="xs:string" default="P_ADDRESS_PLAN_ANY"/>
                            <xs:element name="AddrString" type="xs:string"/>
                            <xs:element name="Name" type="xs:string" minOccurs="0"/>
                            <xs:element name="SubAddressString" type="xs:string" minOccurs="0"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

An example usage could be:

```
{"<?xml version="1.0" encoding="UTF-8"?>
<AddressRangeSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="xml_address_range_set.xsd">
    <AddressRange>
        <Plan>P_ADDRESS_PLAN_E164</Plan>
        <AddrString>789*</AddrString>
    </AddressRange>
    <AddressRange>
        <Plan>P_ADDRESS_PLAN_ANY</Plan>
        <AddrString>123*</AddrString>
    </AddressRange>
    <AddressRange>
        <Plan>P_ADDRESS_PLAN_SIP</Plan>
        <AddrString>&lt;sip:*@parlay.org&gt;</AddrString>
        <Name/>
    </AddressRange>
</AddressRangeSet>"}
```

Note that the final address range corresponds to any sip address @parlay.org, i.e. <sip:*@parlay.org>.

9.3 General Service Properties

Each service instance has the following general properties:

- Service Name
- Service Version
- Service ID
- Service Description
- Product Name
- Product Version
- Supported Interfaces
- Operation Set
- Compatible Service
- Backward Compatibility Level
- Migration Required
- Data Migrated
- Migration Date and Time

The following sections describe these general service properties in more detail. The values for the mode are defined in the type TpServiceTypePropertyMode.

9.3.1 Service Name

Property	Type	Mode	Description
P_SERVICE_NAME	STRING_SET	MANDATORY_ READONLY	This property contains the name of the service, e.g. “UserLocation”, “UserLocationCamel”, “UserLocationEmergency” or “UserStatus”.

9.3.2 Service Version

Property	Type	Mode	Description
P_SERVICE_VERSION	STRING_SET	MANDATORY	This property contains the version of the APIs, to which the service is compliant. It is a set of strings as specified in the TpVersion type.

9.3.3 Service ID

Property	Type	Mode	Description
P_SERVICE_ID	STRING_INTERVAL	READONLY	This property uniquely identifies a specific service. Note that the Framework generates this property value when the Service Supplier registers the service. This property should not be confused with the serviceInstanceID generated by the Framework when a Client Application signs a Service Agreement to obtain the Service Manager

9.3.4 Service Description

Property	Type	Mode	Description
P_SERVICE_DESCRIPTION	STRING_SET	MANDATORY_ READONLY	This property contains a textual description of the service. It should not be interpreted as a description of a Service Instance (as identified by a serviceInstanceID generated by the Framework when a Client Application signs a Service Agreement to obtain the Service Manager).

9.3.5 Product Name

Property	Type	Mode	Description
P_PRODUCT_NAME	STRING_SET	READONLY	This property contains the name of the product that provides the service, e.g. “Find It”, “Locate.com”.

9.3.6 Product Version

Property	Type	Mode	Description
P_PRODUCT_VERSION	STRING_SET	READONLY	This property contains the version of the product that provides the service, e.g. “3.1.11”.

9.3.7 <<deprecated>> Supported Interfaces

This property contains a list of strings with interface names that the service supports, e.g. “IpUserLocation”, “IpUserStatus”. This property is deprecated and will be removed in a future version of the specification.

9.3.8 Operation Set

Property	Type	Mode	Description
P_OPERATION_SET	STRING_SET	MANDATORY	Specifies set of the operations the SCS supports. The notation to be used is : {“Interface1.operation1”,”Interface1.operation 2”, “Interface2.operation1”}, e.g.: {“IpCall.createCall”,”IpCall.routeReq”}.

9.3.9 Compatible Service

Property	Type	Mode	Description
P_COMPATIBLE_WITH_SERVICE	STRING_SET	READONLY	Specifies the Set of Services, identified by their ServiceIDs, with which this new service is compatible. This property should at least be accompanied with the properties P_BACKWARD_COMPATIBILITY_LEVEL, P_MIGRATION_REQUIRED. Note that the new Service can be compatible with more than one Service that is currently registered to the Framework. Therefore this Property is a SET, as well as all related properties like Migration Required, Data Migrated, etc. For all these properties the order of the Services shall be identical.

9.3.10 Backward Compatibility Level

Property	Type	Mode	Description
P_BACKWARD_COMPATIBILITY_LEVEL	BOOLEAN_SET	READONLY	Specifies if the new service is completely backwards compatible with each service identified in the P_COMPATIBLE_WITH_SERVICE property: Value = TRUE: Service is completely backwards compatible Value = FALSE: SCS is not completely backwards compatible. This property requires the presence of P_COMPATIBLE_WITH_SERVICE property. Note that the new Service can be compatible with more than one Service that is currently registered to the Framework. Therefore this Property is a SET, as well as all related properties. For each service identified in P_COMPATIBLE_WITH_SERVICE, one value of this property shall be present in the value set of this property at service registration. For all these properties the order of the Services shall be identical.

9.3.11 Migration Required

Property	Type	Mode	Description
P_MIGRATION_REQUIRED	BOOLEAN_SET	READONLY	Specifies if the new service is replacing the service identified in the P_COMPATIBLE_WITH_SERVICE property: Value = TRUE: new service is replacing the existing one – migration is required before the date/time indicated in P_MIGRATION_DATE_AND_TIME property. Value = FALSE: new service is not replacing the existing one – migration not required, the existing service is retained. This property requires the presence of P_COMPATIBLE_WITH_SERVICE property. If the value set of P_MIGRATION_REQUIRED contains TRUE, P_DATA_MIGRATED and P_MIGRATION_DATE_AND_TIME properties shall also to be present. Note that the new Service can be compatible with more than one Service that is currently registered to the Framework. Therefore this Property is a SET, as well as all related properties. For each service identified in P_COMPATIBLE_WITH_SERVICE, one value of this property shall be present in the value set of this property at service registration. For all these properties the order of the Services shall be identical.

9.3.12 Data Migrated

Property	Type	Mode	Description
P_DATA_MIGRATED	BOOLEAN_SET	READONLY	Indicates if the data (e.g. notifications) from the existing service identified in the P_COMPATIBLE_WITH_SERVICE property is also available in this Service. Value = TRUE: all data is migrated Value = FALSE: no data is migrated This property requires the presence of P_COMPATIBLE_WITH_SERVICE and the P_MIGRATION_REQUIRED properties. Note that the new Service can be compatible with more than one Service that is currently registered to the Framework. Therefore this Property is a SET, as well as all related properties. For each service identified in P_COMPATIBLE_WITH_SERVICE, one value of this property shall be present in the value set of this property at service registration. For all these properties the order of the Services shall be identical.

9.3.13 Migration Date And Time

Property	Type	Mode	Description
P_MIGRATION_DATE_AND_TIME	STRING_SET	READONLY	This property contains the date and time, in the format described in TpDateAndTime, by which point applications shall have migrated from existing services to this new service. Migration to the new service requires the application to terminate the existing service agreement, and sign a new one. Failure to do this by the migration date and time indicated in this property may result in the service agreement being terminated by the Framework, since the service supplier may choose to unregister the service following this date and time. Only one value of TpDateAndTime is permitted to be present in this property at service registration. This property requires the presence of P_COMPATIBLE_WITH_SERVICE, P_MIGRATION_REQUIRED and P_DATA_MIGRATED properties. Note that the new Service can be compatible with more than one Service that is currently registered to the Framework. Therefore this Property is a SET, as well as all related properties. For each service identified in P_COMPATIBLE_WITH_SERVICE, one value of this property shall be present in the value set of this property at service registration. For all these properties the order of the Services shall be identical. For those services for which migration is not required (P_MIGRATION_REQUIRED set to FALSE), the corresponding value of this property shall be ignored.

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