3GPP TSG CN Plenary Meeting #17 **NP-020427**
4 - 6 September 2002, Biarritz, FRANCE

Source: **CN5 (OSA)**

Title: **Rel-5 CRs 29.198-01 OSA API Part 1: Overview**

Agenda item: **8.2**

Document for: **APPROVAL**

Doc-1st-Level	Spec	CR	Rev	Phase	Subject	Cat	Version-Current	Doc-2nd-Level	Workitem
NP-020427	29.198-01	010	-	Rel-5	Addition to ObjectRef description in WSDL Mapping Rules	F	5.0.0	N5-020597	OSA2
NP-020427	29.198-01	011	-	Rel-5	Addition of sequence tag to Choice types	F	5.0.0	N5-020598	OSA2
NP-020427	29.198-01	012	-	Rel-5	Replace all occurrences of the xsd:anyURI type to xsd:string	F	5.0.0	N5-020599	OSA2
NP-020427	29.198-01	013	-	Rel-5	Correction to Namespace mapping in WSDL Mapping Rules	F	5.0.0	N5-020600	OSA2
NP-020427	29.198-01	014	-	Rel-5	Correction to xmlns:wsdl Namespace	F	5.0.0	N5-020601	OSA2
NP-020427	29.198-01	015	-	Rel-5	Prepend class name to <message> name.	F	5.0.0	N5-020602	OSA2
NP-020427	29.198-01	016	-	Rel-5	Correction to void return types in WSDL Mapping Rules	F	5.0.0	N5-020603	OSA2
NP-020427	29.198-01	017	-	Rel-5	Add missing CORBA realization rules in Part 1	F	5.0.0	N5-020738	OSA2
NP-020427	29.198-01	018	-	Rel-5	Add general introduction to the OSA APIs in Part 1	F	5.0.0	N5-020740	OSA2
NP-020427	29.198-01	019	-	Rel-5	Add references to ITU-T/ANSI for encoding of Carrier selection	F	5.0.0	N5-020758	OSA2

joint API group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) N5-020597
Meeting #19, Montreal, CANADA, 8 – 12 July 2002

CR-Form-v5

CHANGE REQUEST

⌘	**29.198-01**	**CR**	**010**	⌘ **rev**	**-**	⌘	Current version:	**5.0.0**	⌘

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

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

Title:	⌘	Addition to ObjectRef description in WSDL Mapping Rules		
Source:	⌘	CN5		
Work item code:	⌘	OSA2	***Date:*** ⌘	12/07/2002
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)*

Reason for change:	⌘	The description pertaining to the use of ObjectRef excludes the case where an ObjectRef may be a part of a structured type.
Summary of change:	⌘	The cases where ObjectRef may be used will be updated to included the case where a reference to an interface may be included within a structured type.
Consequences if not approved:	⌘	If not explicitly stated, mapping to WSDL may be misinterpreted.

Clauses affected:	⌘	Annex B.3	
Other specs affected:	⌘	□ Other core specifications □ Test specifications □ O&M Specifications	⌘
Other comments:	⌘		

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

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

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

B.3 Object References

Object references are used to identify an particular remote object instance. Object references are used in two ways:

1. Passed as a parameter within a method to a remote object or passed as an attribute of a structured type parameter within a method to the remote object.
2. Included within a message to identify the object for which the message is intended.

Within the context of SOAP, an object reference can be represented as a URL appended with a String. The String suffix identifies the particular remote object instance in the context of the URL.

An object reference will be represented by the new type ObjectRef. The ObjectRef type is defined within osa.wsdl as:

```
<xsd:simpleType name=’Objectref’>
   <xsd:restriction base=’xsd:anyURI’ />
</xsd:simpleType>
```

When an object reference is passed as a parameter, the parameter type is defined as a reference to an interface. When an object reference is an attribute of a structured type, that attribute is defined as a reference to an interface. Each interface will have a corresponding reference type associated with it. The interface reference will be defined as:

```
<xsd:simpleType name=’InterfaceNameRef’>
   <xsd:restriction base=’osaxsd:ObjectRef’ />
</xsd:simpleType>
```

where ***InterfaceName*** is the name of the particular interface.

When an object reference is used to identify the intended recipient of a message, then the object reference is included in the SOAP Header element as an ObjectRefHeader. The ObjectRefHeader is defined in the osa.wsdl document as follows:

```
<message name='ObjectRefHeader'>
      <part name='header' element='osaxsd:ObjectRef' />
</message>
```

Within each method, the ObjectRefHeader is bound to the message within the wsdl soap:header element of the input message of the binding element. For example:

```
<binding name='IpAccountManagerBinding' type='am:IpAccountManager'>
     <soap:binding style='rpc' transport='http://schemas.xmlsoap.org/soap/http' />
     <operation name='createNotification'>
        <soap:operation
soapAction='http://www.csapi.org/am/IpAccountManager#createNotification' />
        <input>
           <soap:body
                encodingStyle='http://schemas/xmlsoap.org/soap/encoding/'
                namespace = 'http://www.csapi.org/am.wsdl'
                use='encoded' />
           <soap:header
                message='osaxsd:ObjectRefHeader' part='header' />
            </input>
```

joint API group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) **N5-020598**
Meeting #19, Montreal, CANADA, 8 – 12 July 2002

CR-Form-v5

CHANGE REQUEST

⌘ **29.198-01 CR 011** ⌘ **rev** **-** ⌘ Current version: **5.0.0** ⌘

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

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

Title:	⌘	Addition of sequence tag to Choice types		
Source:	⌘	CN5		
Work item code:	⌘	OSA2	***Date:*** ⌘	12/07/2002
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)*

Reason for change:	⌘	The mapping for Tagged Choice of Data Elements is missing the <xsd:sequence> tag which violates XML Schema rules.
Summary of change:	⌘	The <xsd:sequence> tag will be inserted in between the <xsd:complexType> and <xsd:element> tags.
Consequences if not approved:	⌘	Code is improperly generated. SOAP interoperability issues would occur.

Clauses affected:	⌘	Annex B.4
Other specs affected:	⌘	□ Other core specifications ⌘ □ Test specifications □ O&M Specifications
Other comments:	⌘	

How to create CRs using this form:

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

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

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

<<TaggedChoiceOfDataElements>>

The UML TaggedChoiceOfDataElements data type contains the following attributes:

- Name
- SwitchType
- Roles
    - Name
    - Type

This type would then map to the following XML Schema construct:

```
<xsd:complexType name=”Name”>
   <xsd:sequence>
      <xsd:element name=”SwitchName” type=”SwitchType” />
      <xsd:choice>
        <xsd:element name=”Role-Name” type=”Role-Type” />
        <xsd:element name=”Role-Name” type=”Role-Type” />
        …
        <xsd:element name=”Role-Name” type=”Role-Type” />
      </xsd:choice>
   </xsd:sequence>
</complexType>
```

joint API group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) **N5-020599**
Meeting #19, Montreal, CANADA, 8 – 12 July 2002

CR-Form-v5

CHANGE REQUEST

⌘	**29.198-01**	**CR**	**012**	⌘ **rev**	**-**	⌘	Current version:	**5.0.0** ⌘

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

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

Title:	⌘	Replace all occurrences of the xsd:anyURI type to xsd:string		
Source:	⌘	CN5		
Work item code:	⌘	OSA2	***Date:*** ⌘	12/07/2002
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)*

Reason for change:	⌘	Not every SOAP implementation supports the anyURI type.
Summary of change:	⌘	Any occurrence of the xsd:anyURI type will be replaced by xsd:string
Consequences if not approved:	⌘	Interoperability issues may arise.

Clauses affected:	⌘	Annex B.3	
Other specs affected:	⌘	□ Other core specifications □ Test specifications □ O&M Specifications	⌘
Other comments:	⌘		

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

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

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

B.3 Object References

Object references are used to identify an particular remote object instance. Object references are used in two ways:

1. Passed as a parameter within a method to a remote object.
2. Included within a message to identify the object for which the message is intended.

Within the context of SOAP, an object reference can be represented as a URL appended with a String. The String suffix identifies the particular remote object instance in the context of the URL.

An object reference will be represented by the new type ObjectRef. The ObjectRef type is defined within osa.wsdl as:

```
<xsd:simpleType name=’Objectref’>
   <xsd:restriction base=’xsd:~~anyURI’~~ string’ />
</xsd:simpleType>
```

When an object reference is passed as a parameter, the parameter type is defined as a reference to an interface. Each interface will have a corresponding reference type associated with it. The interface reference will be defined as:

```
<xsd:simpleType name=’InterfaceNameRef’>
   <xsd:restriction base=’osaxsd:ObjectRef’ />
</xsd:simpleType>
```

where ***InterfaceName*** is the name of the particular interface.

When an object reference is used to identify the intended recipient of a message, then the object reference is included in the SOAP Header element as an ObjectRefHeader. The ObjectRefHeader is defined in the osa.wsdl document as follows:

```
<message name='ObjectRefHeader'>
      <part name='header' element='osaxsd:ObjectRef' />
</message>
```

Within each method, the ObjectRefHeader is bound to the message within the wsdl soap:header element of the input message of the binding element. For example:

```
<binding name='IpAccountManagerBinding' type='am:IpAccountManager'>
      <soap:binding style='rpc' transport='http://schemas.xmlsoap.org/soap/http' />
      <operation name='createNotification'>
         <soap:operation
soapAction='http://www.csapi.org/am/IpAccountManager#createNotification' />
         <input>
            <soap:body
                  encodingStyle='http://schemas/xmlsoap.org/soap/encoding/'
                  namespace = 'http://www.csapi.org/am.wsdl'
                  use='encoded' />
            <soap:header
                  message='osaxsd:ObjectRefHeader' part='header' />
             </input>
```

joint API group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) **N5-020600**
Meeting #19, Montreal, CANADA, 8 – 12 July 2002

CR-Form-v5

CHANGE REQUEST

⌘	**29.198-01**	**CR**	**013**	⌘ **rev**	**-**	⌘	Current version:	**5.0.0**	⌘

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

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

Title:	⌘	Correction to Namespace mapping in WSDL Mapping Rules		
Source:	⌘	CN5		
Work item code:	⌘	OSA2	***Date:*** ⌘	12/07/2002
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)*

Reason for change:	⌘	The namespace in the WSDL 'binding' is incorrectly specified.
Summary of change:	⌘	The namespace attributes used in the binding element will be changed from Module-Name.wsdl to Module-Name/wsdl
Consequences if not approved:	⌘	SOAP messages will be incorrect.

Clauses affected:	⌘	Annex B.5	
Other specs affected:	⌘	□ Other core specifications □ Test specifications □ O&M Specifications	⌘
Other comments:	⌘		

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

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

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

Mapping of Interface Class to WSDL *portType* and *binding* elements

A UML Interface Class contains the following attributes:

- Name
- Associated module (i.e. component)
- Operations
- Name
- Parameters
- Name
    - Exceptions
        - Name

This type would then map to the following WSDL portType element:

```
<portType name="Name">
   <operation
     name="Operation-Name"
     <input message="Operation-Name"/>
     <output message="Operation-NameResponse"/>
     <fault message="Operation–Exception– Name" />
     …
     <fault message="Operation–Exception–Name" />
   </operation>
   …
   <operation
     name="Operation-Name"
     <input message="Operation-Name"/>
     <output message="Operation-NameResponse"/>
     <fault name="Operation-Exception-Name" message="Operation–Exception–Name"
/>
     …
     <fault message="Operation–Exception–Name" />
   </operation>
</portType>
```

This type would also then map into the following WSDL binding element:

```
<binding
   name="Interface-NameBinding"
   type="Interface-Name">
   <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>

   <operation name="Operation-Name">
     <soap:operation soapAction="http://www.csapi.org/am/Name#Operation-Name"/>
     <input>
       <soap:body
         encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
         namespace="http://www.csapi.org/Module-Name~~.~~/wsdl"
```

```
           use="encoded"/>
              <soap:header message="osaxsd:ObjRefHeader" part="header" />
      </input>
      <output>
        <soap:body
           encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
           namespace=" http://www.csapi.org/Module-Name~~.~~/wsdl "
           use="encoded"/>
      </output>
       <fault>
              <soap:fault name="Operation-Exception-Name"
           encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
           namespace="http://www.csapi.org/Module-Name~~.~~/wsdl"
           use="encoded"/>
       </fault>

       … additional fault elements

    </operation>

    … additional operation elements

  </binding>
```

joint API group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) **N5-020601**
Meeting #19, Montreal, CANADA, 8 – 12 July 2002

CR-Form-v5

CHANGE REQUEST

⌘ **29.198-01 CR 014** ⌘ **rev** **-** ⌘ Current version: **5.0.0** ⌘

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

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

Title:	⌘ Correction to xmlns:wsdl Namespace		
Source:	⌘ CN5		
Work item code:	⌘ OSA2	***Date:*** ⌘	12/07/2002
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)*

Reason for change: ⌘	The xmlns:wsdl is incorrect.
Summary of change: ⌘	The namespace is changed from: xmlns:wsdl = 'http://http://schemas.xmlsoap.org/wsdl/' to: xmlns:wsdl = 'http://schemas.xmlsoap.org/wsdl/'
Consequences if not approved: ⌘	SOAP messages may be incorrect. Code generation tools will not generate the proper code. Interoperability issues may arise.

Clauses affected: ⌘	Annex B.2	
Other specs affected: ⌘	□ Other core specifications □ Test specifications □ O&M Specifications	⌘
Other comments: ⌘		

How to create CRs using this form:

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

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

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

B.2 Proposed Namespaces for the OSA WSDL

Namespaces are an important part of an XML Schema. They are used to qualify the source of a particular XML element.

There are several XML/SOAP/WSDL related Namespaces which are used within each of the WSDL documents. The Namespace Prefix and the Namespace are noted below.

xmlns:wsdl = ‘~~http://~~http://schemas.xmlsoap.org/wsdl/’

xmlns:soap=’http://schemas.xmlsoap.org/wsdl/soap/’

xmlns:SOAP-ENC=’http://schemas.xmlsoap.org/soap/encoding/’

xmlns:xsd:=’http://www.w3c.org/2001/XMLSchema’

There are also OSA specific namespaces which are used within the OSA WSDL documents. The OSA related namespaces present within each WSDL document depends on the WSDL document and which WSDL documents it imports. The guidelines used to derive these namespaces are:

- The root namespace for the OSA WSDL and XML schemas is http://www.csapi.org/
- There is one document generated for each component (Module) within the Analysis UML model. The document will have the name of the UML component with the extension ‘.wsdl’ For each wsdl document generated the following additional namespaces will be included:
    - xmlns:<component name>=’http://www.csapi.org/<component name>/wsdl’
    - xmlns:<component name>xsd=’http://www.csapi.org/<component name>/schema’
    - For each OSA wsdl document which is referenced by an import statement within the current wsdl document then the following additional namespaces will be included.
        - xmlns:<imported component name>=’http://www.csapi.org/<imported component name>/wsdl’
        - xmlns:<imported component name>xsd=’http://www.csapi.org/<imported component name>/schema’
- Attributes which require a QName value shall use the appropriate Namespace Prefix (as defined in the definitions element of the wsdl file) to qualify the element being referenced.

The namespaces are defined within the ‘definitions’ element of a wsdl document. For example, the definitions element of the am.wsdl document would look like:

```
<definitions
   name='am'
   targetNamespace='http://www.csapi.org/am/wsdl'
   xmlns='http://schemas.xmlsoap.org/wsdl/'
   xmlns:wsdl='http://schemas.xmlsoap.org/wsdl/'
   xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/'
   xmlns:SOAP-ENC='http://schemas.xmlsoap.org/soap/encoding/'
   xmlns:xsd='http://www.w3.org/2001/XMLSchema'
   xmlns:am='http://www.csapi.org/am/wsdl'
   xmlns:amxsd='http://www.csapi.org/am/schema'
   xmlns:osa='http://www.csapi.org/osa/wsdl'
   xmlns:osaxsd='http://www.csapi.org/osa/schema'>

<import namespace='http://www.csapi.org/osa/wsdl'
           location='osa.wsdl' />
```

joint API group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) **N5-020602**
Meeting #19, Montreal, CANADA, 8 – 12 July 2002

CR-Form-v5

CHANGE REQUEST

⌘	**29.198-01**	**CR**	**015**	⌘ **rev**	**-**	⌘	Current version:	**5.0.0**	⌘

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

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

Title:	⌘	Prepend class name to <message> name.		
Source:	⌘	CN5		
Work item code:	⌘	OSA2	***Date:*** ⌘	12/07/2002
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)*

Reason for change:	⌘	A wsdl file may contain duplicate messages if interfaces contain duplicate operations.
Summary of change:	⌘	Each name attribute of the message elements will be prepended with the name of the interface to which the message belongs. This will uniquely identify each message element within the wsdl file.
Consequences if not approved:	⌘	Code is improperly generated.

Clauses affected:	⌘	Annex B.5		
Other specs affected:	⌘	□ Other core specifications □ Test specifications □ O&M Specifications	⌘	
Other comments:	⌘			

How to create CRs using this form:

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

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

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

Mapping of Operations to WSDL *message* element

A UML Operation contains the following attributes:

- Interface
- Name
- Module Name
- Return Type
- Parameter
  - Name
  - Type

This type would then map to the following XML Schema construct:

```
<message name="Interface_Name">
   <part
     name="Parameter-Name"
     type="Parameter-Type"/>
   …
   <part
     name="Parameter-Name"
     type="Parameter-Type"/>
</message>

<message name="Interface_NameResponse">
   <part name="return" type="ReturnType"/>
</message>


Note: If the ReturnType is void, then no ‘type’ attribute would be included in
the Response message.
```

joint API group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) **N5-020603**
Meeting #19, Montreal, CANADA, 8 – 12 July 2002

CR-Form-v5

CHANGE REQUEST

⌘	**29.198-01**	**CR**	**016**	⌘ **rev**	**-**	⌘	Current version:	**5.0.0**	⌘

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

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

Title:	⌘	Correction to void return types in WSDL Mapping Rules		
Source:	⌘	CN5		
Work item code:	⌘	OSA2	***Date:*** ⌘	12/07/2002
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)*

Reason for change:	⌘	The message elements represent void return types are incorrectly specified.
Summary of change:	⌘	The <part> element of the <message> elements which represent void return types will be removed.
Consequences if not approved:	⌘	SOAP messages may be incorrect. Code generation tools will not generate the proper code.

Clauses affected:	⌘	Annex B.5		
Other specs affected:	⌘	□ Other core specifications □ Test specifications □ O&M Specifications	⌘	
Other comments:	⌘			

How to create CRs using this form:

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

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

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

Mapping of Operations to WSDL *message* element

A UML Operation contains the following attributes:

- Name
- Module Name
- Return Type
- Parameter
- Name
- Type

This type would then map to the following XML Schema construct:

```
<message name="Name">
   <part
     name="Parameter-Name"
     type="Parameter-Type"/>
   …
   <part
     name="Parameter-Name"
     type="Parameter-Type"/>
</message>

<message name="NameResponse">
   <part name="return" type="ReturnType"/>
</message>
```

Note: If the ReturnType is void, then no ‘~~type’ attribute~~’part’ element would be included in the Response message. (i.e. <message name=”***Name***Response” />

joint API group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) **N5-020738**
Meeting #19, Montreal, CANADA, 8 – 12 July 2002

CR-Form-v5

CHANGE REQUEST

⌘	**29.198-01**	**CR**	**017**	⌘ **rev**	**-**	⌘	Current version:	**5.0.0** ⌘

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

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

Title:	⌘	Add missing CORBA realization rules in Part 1		
Source:	⌘	CN5		
Work item code:	⌘	OSA2	***Date:*** ⌘	12/07/2002
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)*

Reason for change:	⌘	The CORBA realization rules are unspecified.
Summary of change:	⌘	Additional CORBA realization rules are added.
Consequences if not approved:	⌘	Interoperability problems.

Clauses affected:	⌘	Annex A.	
Other specs affected:	⌘	□ Other core specifications □ Test specifications □ O&M Specifications	⌘
Other comments:	⌘		

How to create CRs using this form:

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

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

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

Annex A (normative):
OMG IDL

A.1 Tools and Languages

The Object Management Group's (OMG) [15] Interface Definition Language (IDL) is used as a means to programmatically define the interfaces. IDL files are either generated manually from class diagrams or by using a UML tool. In the case IDLs are manually written and/or being corrected manually, correctness has been verified using a CORBA2 (orbos/97-02-25) compliant IDL compiler, e.g. [13].

A.2 Namespace

The used namespace in CORBA IDL is org.csapi.

A.3 Object References

In CORBA IDL it is not needed to explicitly indicate a reference to an object. Where the specifications explicitly indicate a reference to an object by adding "Ref" to the object type, this addition is removed when mapped to the IDL.

Example 1: struct TpMultiPartyCallIdentifier {
    IpMultiPartyCall CallReference;
    TpSessionID CallSessionID;
};

A.~~2~~4 ~~Strings and Collections~~Mapping of Datatypes

A.4.1 Basic Datatypes

In IDL, the data type *String* is typedefed (see Note below) from the CORBA primitive *string*. This CORBA primitive is made up of a length and a variable array of byte.

NOTE: A *typedef* is a type definition declaration in IDL.

TpBoolean maps to a CORBA boolean, TpInt32 to a CORBA long, TpFloat to a CORBA float, and TpOctet to a CORBA octet.

A.4.2 Constants

All constants are mapped to a CORBA const of type TpInt32.

Example 2: const TpInt32 P_TASK_REFUSED = 14;

A.4.3 Collections

In OMG IDL, ~~this~~ collections (Numbered Set and Numbered List) map~~s~~ to a sequence of the data type. A CORBA sequence is implicitly made of a length and a variable array of elements of the same type.

Example ~~1~~3: typedef sequence<TpSessionID> TpSessionIDSet;

Collection types can be implemented (for example, in C++) as a structure containing an integer for the *number* part, and an array for the *data* part.

Example ~~2~~4: The TpAddressSet data type may be defined in C++ as:

```
typedef struct {
    short       number;
    TpAddress   address [];
} TpAddressSet;
```

The array "address" is allocated dynamically with the exact number of required TpAddress elements based on "number".

A.4.4 Sequences

In OMG IDL sequences map to a CORBA Struct.

```
Example 5:  struct TpAddress {
                TpAddressPlan Plan;
                TpString AddrString;
                TpString Name;
                TpAddressPresentation Presentation;
                TpAddressScreening Screening;
                TpString SubAddressString;
            };
```

A.4.5 Enumerations

In OMG IDL enumerations map to a CORBA enum.

```
Example 6:     enum TpAddressScreening {
                   P_ADDRESS_SCREENING_UNDEFINED ,
                   P_ADDRESS_SCREENING_USER_VERIFIED_PASSED,
                   P_ADDRESS_SCREENING_USER_NOT_VERIFIED,
                   P_ADDRESS_SCREENING_USER_VERIFIED_FAILED ,
                   P_ADDRESS_SCREENING_NETWORK
               };
```

A.4.6 Choices

A choice maps to a CORBA union. For entries that do not have a corresponding type (defined as NULL in the specification) no union entry is generated. These entries are grouped in the default clause where NULL is replaced by short and the entry name (Undefined) by the name Dummy. When there are no NULL entries, the default clause is not generated.

```
Example 7:  union TpCallAdditionalErrorInfo switch (TpCallErrorType) {
                case P_CALL_ERROR_INVALID_ADDRESS: TpAddressError CallErrorInvalidAddress;
                default: short Dummy;
            };

Example 8:  union TpCallChargeOrder switch(TpCallChargeOrderCategory) {
                case P_CALL_CHARGE_TRANSPARENT: TpOctetSet TransparentCharge;
                case P_CALL_CHARGE_PREDEFINED_SET: TpInt32 ChargePlan;
            };
```

A.5 Use of NULL

CORBA allows the value NULL to be used for object references only. When the specification mentiones NULL as possible value of a struct, it means that each object reference in the struct shall be set to NULL. NULL does not apply to other datatypes then object references.

A.6 Exceptions

The TpCommonExceptions is mapped to a CORBA exception containing a data item of type TpInt32 to indicate the type of general exception and extra information of type TpString.

Example 9: exception TpCommonExceptions {
```
    TpInt32 ExceptionType;
    TpString ExtraInformation;
};
```

All other exceptions are also mapped to CORBA exceptions but containing a data item of type TpString to indicate additional information.

Example 10: exception P_INVALID_ASSIGNMENT_ID {
```
    TpString ExtraInformation;
};
```

joint API group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) **N5-020740**
Meeting #19, Montreal, CANADA, 8 – 12 July 2002

CR-Form-v5

CHANGE REQUEST

⌘	**29.198-01**	**CR**	**018**	⌘ **rev**	**-**	⌘	Current version:	**5.0.0** ⌘

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

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

Title:	⌘	Add general introduction to the OSA APIs in Part 1		
Source:	⌘	CN5		
Work item code:	⌘	OSA2	***Date:*** ⌘	12/07/2002
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)*

Reason for change:	⌘	An introduction into the OSA APIs is missing in the specification.
Summary of change:	⌘	Introduction to OSA APIs added to Part 1.
Consequences if not approved:	⌘	Interoperability problems because application developers might not fully understand how to program against the OSA APIs.

Clauses affected:	⌘	7	
Other specs affected:	⌘	□ Other core specifications □ Test specifications □ O&M Specifications	⌘
Other comments:	⌘		

How to create CRs using this form:

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

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

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

7. Introduction to Parlay/OSA APIs

This section contains the general rules that were followed by the design of the Parlay/OSA APIs and advice for how to use them. Note however that exceptions to these "rules" may exist and that examples are not exhaustive.

7.1 Interface Types

In the Parlay/OSA specifications different types of interfaces are distinguished:

– Application side (callback) interfaces. This type of interface needs to be implemented by an application (client) and the name of such an interface is prefixed with "IpApp".

– Interfaces of an SCF that are used by the Framework. The name of this type of server interface is prefixed with "IpSvc".

– Application side interfaces and SCF interfaces that are shared. The name of this type of interface is prefixed with "IpClient"

– Interfaces of the Framework that are used by an SCF. The name of this type of server interface is prefixed with "IpFw".

The name of all other interfaces of the Framework and SCFs that are used by an application, is prefixed with "Ip".

7.2 Service Factory

For each application that uses an SCF, a separate object is created to handle all communication to the application. This object is referred to as the Service Manager. The pattern used is often referred to as the Factory Pattern. The Service Manager creates any new objects in the SCF. The Service Manager and all the objects created by it are referred to as "service instance".

Once an application is granted access to an SCF, the Framework requests the SCF to create a new Service Manager. The reference to this Service Manager is provided to the application. From this moment onwards the application can start using the SCF.

7.3 Use of Sessions

A session is a series of interactions between two communication end points that occur during the span of a single connection. An example is all operations to set-up, control, and tear down a (multi-party) call. A session is identified by a Session ID. This ID is unique within the scope of a service instance and can be related to session numbers used in the network.

7.4 Interfaces and Sessions

Some interfaces have a one-to-one relation with a session. For every session there is a separate interface instance. In this case, this instance of an interface represents the session. All methods invoked on such an interface operate on the same session. These interfaces make no use of Session IDs.

Other interfaces can represent multiple sessions. The underlying implementation can then either create an instance per session or it can handle multiple sessions per instance (e.g., to combat extensive resource usage). When a method on such an interface is invoked it requires a Session ID to uniquely identify the session to which it applies.

7.5 Callback Interfaces

Some Parlay/OSA interfaces require an application to register a callback interface. This interface resides on the client (application) side and is used by the server (service) to report events, results, and errors. An application shall register its callback interface as soon as the corresponding server side interface is created.

7.6 Setting Callbacks

Two methods are available in every service interface that can be used for setting the callback interface: setCallback() and setCallbackWithSessionID(). Interfaces that do not use sessions shall (obviously) only implement setCallback(). An invocation of setCallbackWithSessionID() on such interfaces shall result in an exception (P_TASK_REFUSED).

Interfaces that use sessions shall only implement setCallbackWithSessionID(). An invocation of setCallback() on such interfaces shall result in an exception (P_TASK_REFUSED). This regardless of whether an interface instance actually implements multiple sessions or not.

7.7 Synchronous versus Asynchronous Methods

Two types of methods exist in Parlay/OSA interfaces. When a method does not require the SCS to contact other nodes in the network it is implemented as a synchronous method. When the method returns, the result (if applicable) of the operation is provided to the application. When an error occurs, an exception is thrown. Examples of synchronous methods are methods to retrieve data that is available in the SCS and methods that create an object.

In other cases, a method requires the SCS to contact other nodes in the network. There can be a delay between the moment a message is sent into the network and the moment that the result is received or an error is detected. To prevent that the application is blocked or that an application has to “guess” whether there is a problem in the SCS, these types of methods are made asynchronous.

An asynchronous method of an interface can be recognized by the fact that its name ends with “Req” (from request) and that in the corresponding callback interface two methods are included with the same name but ending with “Res” (from result) and “Err” (from error) instead. When no error has occurred, the “Res” method will be invoked when the result is available. In case an error has been detected, the “Err” method is invoked. Problems that can be detected by the SCS itself (for instance illegal parameter values) will result in exceptions being thrown when the “Req” method is called. After a “Req” method has returned, only errors shall be reported.

Because it is possible that multiple requests can be done in parallel (invoking multiple times a “Req” method without having received a result or error) a mechanism is needed to link requests with responses. Therefore, the “Req” method returns an Assignment ID and the “Res” and “Err” methods have this Assignment ID as input parameter. For session based interfaces the Session ID can be used also.

Some “Req” methods can result in multiple “Res” methods being invoked. However, the corresponding “Err” method will never be invoked more than once.

Note that methods on client side interfaces shall never raise an exception unless this is explicitly described in the specification.

Some methods switch on/off reports (for instance triggered location reports). These methods are of a different kind and do not follow the pattern that is described in this section.

A deadlock is a potential danger when using asynchronous methods, especially in single threaded systems. It can occur that client and server are waiting for each other for a task to be completed. It is considered good practice to build in mechanisms to prevent dealock from occuring, for instance by using multiple threads or using time-outs on remote method calls.

7.8 Out Parameters

Methods used in Parlay/OSA interfaces only have input parameters. Any result can only be reported by a return value. If multiple values need to be returned, a datatype is required that consists of a sequence of values. A value of this datatype is then returned by a method. This approach has been chosen because not all middleware solutions are (or may be) capable of dealing with (multiple) output parameters.

7.9 Exception Hierarchy

Exceptions are organized in an exception hierarchy. For the general exceptions and for each service type an abstract exception class is defined. Advantage for an application programmer is that (s)he does not need to catch all the specific exceptions, but may catch only the abstract exceptions.

Note however that the exception hierarchy is only available when the applicable Parlay/OSA realisation supports this. Java does, but CORBA and WSDL/SOAP do not.

7.10 Common Exceptions

Exception TpCommonExceptions can be thrown by any method. It is an aggregate of a number of general problems. To prevent that each method's signature requires all these exceptions they are all included in a single exception class.

The following rules apply on when what type of general exception shall be thrown:

- P_RESOURCES_UNAVAILABLE is thrown when a physical resource in the network is not available.
- P_INVALID_STATE is thrown when a method is called that is not allowed in the state that the Parlay/OSA state machines are in.
- P_TASK_CANCELLED is thrown in case of a temporary problem.
- P_TASK_NO_CALLBACK_ADDRESS_SET is thrown when no callback address has been set.
- P_METHOD_NOT_SUPPORTED is thrown when the application initiates methods that are either not according to the Service Level Agreement or not supported in the SCS.
- P_TASK_REFUSED is thrown in case of a problem that is not temporary and when none of the other common or dedicated exceptions apply.

Note that methods on application side callback interfaces shall never raise an exception unless explicitly stated in the specification.

7.11 Use of NULL

The Parlay/OSA specifications contain references to the NULL value to indicate the absence of a certain parameter. An example where this is used is for specifying NULL as a callback reference.

A parameter description for parameters of any datatype can indicate that NULL is a possible value. The realisation of NULL can differ per technology. A NULL value for a sequence in CORBA means that all its members shall be NULL while in Java the whole structure could be NULL.

Note that it always shall be stated in the specification when a NULL value can be expected.

7.12 Notification Handling

Several Parlay/OSA SCFs provide a mechanism for creating and receiving notifications. A notification is the reporting of an event occurring in the network or SCS. Examples of notifications are answer, busy, and on hook events.

This section describes the general mechanism of notification handling. Note that it might not apply (exactly) to every API.

There are two types of notifications. One that is created by an application and one that is controlled by the network. The first type normally is used when an ASP is responsible for service provisioning and has to create its own notifications in order to be able to serve subscribers. The second type is used when the network operator does service provisioning. The network operator creates the notifications and an application only needs to handle them.

Note that normally both mechanisms will not be used by one application. However, the Parlay/OSA interfaces do not prohibit this.

Another way to distinguish notifications is by monitor mode. Notifications can be requested in either NOTIFY or INTERRUPT mode. When requested in NOTIFY mode, the notifications is reported to the application but the SCS continues processing. For notifications requested in INTERRUPT mode, processing in the SCS is suspended when the notification is reported to the application. The application has to instruct the SCS explicitly (within a certain maximum time) how to proceed the processing. Note that not all SCFs support notifications in INTERRUPT mode.

When a notification is created and when an application registers for network controlled notifications a callback interface needs to be provided. This callback interface is used for reporting the notifications. There are however a few things that are worth mentioning here:

- Each time a (set of) notifications(s) is created, a callback is specified that is used for reporting the requested notifications. This callback interface may be the same, but may also differ. The assignment ID can be used to link a notification report to the creation of registration.
- Registering a callback for network controlled notifications needs to be done only once. The callback interface that is provided may be the same as the one used for creating a notification (note again that it is however not recommended to used both mechanisms in the same application).
- The callback specified when creating or registering for events overrules the callback set with setCallback() or setCallbackWithSessionID(). This means that this one will NOT be used for reporting notifications . It will however be used for all other methods that require the callback interface.
- Only if NULL is provided as callback interface reference, the callback interface that was set using setCallback() or setCallbackWithSessionID() is used for reporting notifications.
- It is possible to recreate a (set of) notification(s) or re-register for notifications. This is only useful when providing a different callback interface reference. In this case, the last provided interface is used for reporting notifications. The earlier provided callback interface is used as “backup” interface (this can be the one provided with setCallback() or setCallbackWithSessionID() if NULL was provided initially). Notifications are reported on this interface when calls to the most recent provided callback interface fail (object providing the interface is crashed or overloaded). When re-creating or re-registering, the same assignment ID is returned.

joint API group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) N5-020758
Meeting #19, Montreal, CANADA, 8 – 12 July 2002

CR-Form-v5

CHANGE REQUEST

⌘	**29.198-01**	**CR**	**019**	⌘ **rev**	**-**	⌘	Current version:	**5.0.0** ⌘

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

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

Title:	⌘	Add references to ITU-T/ANSI for encoding of Carrier selection		
Source:	⌘	CN5		
Work item code:	⌘	OSA2	***Date:*** ⌘	19/07/2002
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)*

Reason for change:	⌘	Addition of missing references
Summary of change:	⌘	Addition of references for encoding of Carrier selection
Consequences if not approved:	⌘	Incomplete list of references

Clauses affected:	⌘	2
Other specs affected:	⌘	□ Other core specifications ⌘ □ Test specifications □ O&M Specifications
Other comments:	⌘	Companion Rel-5 CR 29.198-04-3 (N5-020759)

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

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

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

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 22.127: "Service Requirement for the Open Services Access (OSA)".

[3] 3GPP TS 23.127: "Virtual Home Environment / Open Service Access (OSA)".

[4] 3GPP TS 23.078: "Customised Applications for Mobile network Enhanced Logic (CAMEL); Stage 2".

[5] 3GPP TS 22.101: "Service Aspects; Service Principles".

[6] World Wide Web Consortium "Composite Capability/Preference Profiles (CC/PP): A user side framework for content negotiation" (http://www.w3.org/TR/NOTE-CCPP/).

[7] 3GPP TS 29.002: "Mobile Application Part (MAP) specification".

[8] 3GPP TS 29.078: "Customised Applications for Mobile network Enhanced Logic (CAMEL); CAMEL Application Part (CAP) specification".

[9] Wireless Application Protocol (WAP), Version 2.0: "User Agent Profiling Specification" (WAP-248) (http://www.wapforum.org/what/technical.htm).

[10] Wireless Application Protocol (WAP), Version 2.0: "WAP Service Indication Specification" (WAP-167) (http://www.wapforum.org/what/technical.htm).

[11] Wireless Application Protocol (WAP), Version 2.0: "Push Architectural Overview" (WAP-250) (http://www.wapforum.org/what/technical.htm).

[12] Wireless Application Protocol (WAP), Version 2.0: "Wireless Application Protocol Architecture Specification" (WAP-210) (http://www.wapforum.org/what/technical.htm).

[13] IDL to Java Compiler (http://www.javasoft.com/products/jdk/idl/index.html).

[14] UML Unified Modelling Language (http://www.omg.org/uml).

[15] Object Management Group (http://www.omg.org).

[16] 3GPP TS 22.002: "Circuit Bearer Services (BS) supported by a Public Land Mobile Network (PLMN)".

[17] 3GPP TS 22.003: "Circuit Teleservices supported by a Public Land Mobile Network (PLMN)".

[18] 3GPP TS 24.002: "GSM - UMTS Public Land Mobile Network (PLMN) Access Reference Configuration".

[19] ITU-T Q.763: "Signalling System No. 7 – ISDN user part formats and codes".

[20] ITU-T Q.931: "ISDN user-network interface layer 3 specification for basic call control".

[21] ISO 8601: "Data elements and interchange formats -- Information interchange -- Representation of dates and times".

[22] ISO 4217: "Codes for the representation of currencies and funds".

[23] 3GPP TS 22.121: "Service aspects; The Virtual Home Environment; Stage 1".

[24] “The Parlay Group homepage” (http://www.parlay.org)

[25] “JAIN Community homepage” (http://www.java.sun.com/products/jain)

[26] 3GPP TS 23.057: "Mobile Execution Environment (MExE); Functional Description; Stage 2".

[27] “JSR Overview” (http://jcp.org/jsr/overview/index.en.jsp)

[28] “Java 2 SDK, Standard Edition” (http://java.sun.com/j2se/1.4/docs/relnotes/features.html)

[29] “Java Community Process” (http://jcp.org/)

[30] "World Wide Web Consortium homepage" (http://www.w3c.org)

[31] 3GPP TS 23.271 "Functional stage 2 description of location services (3GPP TS 23.271)"

[32] ITU-T Q.763, Signalling System No. 7 - ISDN user part formats and codes

[33] ANSI T1.113, Signalling System No. 7 (SS7) - Integrated Services Digital Network (ISDN) User Part