3GPP TSG CN Plenary Meeting #21 **NP-030352**
17 - 19 September 2003, Frankfurt am Main, GERMANY

Source: **CN5 (OSA)**

Title: **15 Rel-5 CRs 29.198-xy OSA API Parts 1-14: Correction to Java Realisation Annex**

Agenda item: **8.2**

Document for: **APPROVAL**

Doc-1st-Level	Spec	CR	Rev	Phase	Subject	Cat	Version-Current	Doc-2nd-Level
NP-030352	29.198-01	024	-	Rel-5	Correction to Java Realisation Annex	F	5.2.0	N5-030412
NP-030352	29.198-02	036	-	Rel-5	Correction to Java Realisation Annex	F	5.3.0	N5-030413
NP-030352	29.198-03	085	-	Rel-5	Correction to Java Realisation Annex	F	5.3.0	N5-030414
NP-030352	29.198-04-1	007	-	Rel-5	Correction to Java Realisation Annex	F	5.3.0	N5-030415
NP-030352	29.198-04-2	008	-	Rel-5	Correction to Java Realisation Annex	F	5.3.0	N5-030416
NP-030352	29.198-04-3	014	-	Rel-5	Correction to Java Realisation Annex	F	5.3.0	N5-030417
NP-030352	29.198-04-4	009	-	Rel-5	Correction to Java Realisation Annex	F	5.3.0	N5-030418
NP-030352	29.198-05	039	-	Rel-5	Correction to Java Realisation Annex	F	5.3.0	N5-030420
NP-030352	29.198-06	023	-	Rel-5	Correction to Java Realisation Annex	F	5.2.0	N5-030421
NP-030352	29.198-07	014	-	Rel-5	Correction to Java Realisation Annex	F	5.3.0	N5-030422
NP-030352	29.198-08	026	-	Rel-5	Correction to Java Realisation Annex	F	5.3.0	N5-030423
NP-030352	29.198-11	020	-	Rel-5	Correction to Java Realisation Annex	F	5.2.0	N5-030426
NP-030352	29.198-12	022	-	Rel-5	Correction to Java Realisation Annex	F	5.2.0	N5-030427
NP-030352	29.198-13	004	-	Rel-5	Correction to Java Realisation Annex	F	5.1.0	N5-030428
NP-030352	29.198-14	013	-	Rel-5	Correction to Java Realisation Annex	F	5.2.0	N5-030429

CR-Form-v7

CHANGE REQUEST

⌘ **29.198-01** **CR** **024** ⌘ **rev** **-** ⌘ Current version: **5.2.0** ⌘

*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:	⌘	Correction to Java Realisation Annex
Source:	⌘	CN5 AePONA – Eamonn Murray
Work item code:	⌘	OSA2
Date:	⌘	08/09/2003
Category:	⌘	**F**
Release:	⌘	*REL-5*

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

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

Reason for change:	⌘	Correction to Annex C of the current specification. The production of JavaDoc as part of the Java Realisation process, requires that the FULL API is considered in producing the Java doc so that references between the subparts of the specification can be resolved. Therefore the JavaDoc is to be included as part of the Overview of the API rather than as a discrete subpart of each part of the specification.
Summary of change:	⌘	Add reference to J2SE and J2EE Javadoc files.
Consequences if not approved:	⌘	The Javadoc component of the Java Realisation cannot be produced within each part of the specification and therefore the Java Realisation will be incomplete.

Clauses affected:	⌘	Annex C

		Y	N			
Other specs affected:	⌘	X		Other core specifications	⌘	Rel-5 29.198-xy
			X	Test specifications		
			X	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/specs/CR.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.

KEEP the History box of the TS to be changed (see end of the present document)

********** Start of Change # 1 **********

Annex C (informative): Java Realisation API

C.1 Java Realisation Overview

The Parlay/OSA UML specifications are defined in a technology neutral manner. This annex aims to deliver for Java, a developer API, provided as a realisation, supporting a Java API that represents the UML specifications.

C.1.1 J2SE API

The J2SE API supports a J2SE development environment that

- provides an abstraction of the Parlay/OSA APIs that provides a local API for J2SE developers
- supports a listener based API for SCFs and a callback API for the Framework
- uses local object references as correlation mechanisms as Java developers are familiar with object correlation
- is a local API without visibility to the underlying transport

C.1.2 J2EE API

The J2EE API supports a development environment which allows the creation of J2EE and Java RMI interfaces for both the server and client, ensuring consistent interfaces for interoperability. These interfaces may be used for Java RMI on either JRMP or IIOP (RMI/IIOP), allowing use in J2EE environments. The interfaces may also be used as a thin layer on other transports, similar to other Java technologies that provide a RMI programming interface.

The J2EE API is a suitable base for Java across Java platforms, allowing creation of implementations that:

- may be a thin layer on transport protocols
- may support J2EE remote interfaces
- may support J2EE local interfaces

The Java files created with the realisation will be made available with the Parlay/OSA specifications.

The remaining sections of this annex deal with the following areas:

- section C.2 covers the tools and languages used to produce and define the Java Realisation
- section C.3 covers the mappings that are common across both Java Realisation APIs
- section C.4 covers the mappings specific to the J2SE API
- section C.5 covers the mappings specific to the J2EE API

C 1.3 Javadoc

The Javadoc that accompanies the J2SE realisation of the Parlay/OSA API specification is provided as archive 2919801J2SE.ZIP that accompanies the present document.

The Javadoc that accompanies the J2EE realisation of the Parlay/OSA API specification is provided as archive 2919801J2EE.ZIP that accompanies the present document.

C.2 Tools and languages

The Java language is used as a means to programmatically define the interfaces. Java source files are generated automatically from UML. The Java source files are created in accordance with the mappings defined within this annex.

The generated Java source files are verified syntactically using Java compilers such as javac. The Java API comprises

- J2SE API designed to be compatible with the Java 2 SDK, Standard Edition, version 1.3 (http://java.sun.com/j2se/1.3/) or later and a
- J2EE API compatible with the Java 2 Enterprise Edition (http://java.sun.com/j2ee/).

The J2SE API, developed in accordance to the conventions defined in section C.3 and C.4 will enable:

- portable Java applications, as far as the Java API is concerned
- independence of distribution mechanism technology (e.g. CORBA,SOAP,RMI)

C.3 Generic Mappings (Elements common to J2SE and J2EE)

NOTE: All Java code examples given in this section are taken from the J2SE Java Realisation API. See the appropriate Java files for examples for J2EE classes.

C.3.1 Namespace

The UML namespace org.csapi is represented by the Java package org.csapi.jr.

Packages under the org.csapi.jr package will contain "se" packages for J2SE specific Java artefacts and "ee" and "eelocal~~.remote~~" packages for J2EE specific Java artefacts.

For example, the User Location Camel Service package structure would appear as follows:

org.csapi.jr.se.mm.ulc containing J2SE API Java artefacts

org.csapi.jr.eelocal.mm.ulc containing J2EE local API Java artefacts

org.csapi.jr.ee~~.remote~~.mm.ulc containing the J2EE remote/RMI API Java artefacts

C.3.2 Package Naming Conventions

UML packages will be represented by Java packages. The sub-namespaces below the root namespaces described above will follow the naming used for the UML namespaces.

C.3.3 Object References

In Java there is no need to explicitly indicate a reference to an object as in Java objects are passed by value and not by reference. Where the specifications explicitly indicate a reference to an object by adding "Ref" to the object type, this addition is removed in the Java realisation.

Example 1:

UML	Java Realisation
IpUserLocationCamelRef	UserLocationCamel
IpCallRef	Call

C.3.4 Element Naming

The UML element names that begin with an uppercase will follow the Java naming conventions of with a leading lower case letter and mixed case names. The UML elements are equivalent to Java field names.

Example 2:

UML	Java Realisation
AddressPlan	addressPlan

C.3.5 Element Naming Collisions

If an element name collides with a Java keyword, the element name will be prefixed with an underscore.

Example 3:

UML	Java Realisation
Final	_final

C.3.6 Data Type Definitions

C.3.6.1 Basic Data Types

Java does not support type definitions (typedefs); therefore types are unwound to their basic data types e.g.:

Example 4:

UML	Java Realisation
TpCallAlertingMechanism	int
TpAccessType	java.lang.String

The following mappings apply to the basic data types:

UML	Java Realisation
TpBoolean	boolean
TpInt32	int
TpInt64	long
TpFloat	float
TpOctet	byte
TpString	java.lang.String
TpLongString	java.lang.String
TpAny	java.lang.Object

C.3.6.2 Constants

Constants are associated with a type definition or as a standalone entity. In both cases, the constant itself will be defined as a '`public final static`' field using its name and value.

When defined associated with a type definition, an interface using the name of the type definition will be defined enclosing all constants associated with the type definition.

Standalone constants within a package are defined within a Java interface with the name 'Constants' within that package.

Example 5:

```
package org.csapi.jr.se;
```

```
public interface Constants {
        public static final int METHOD_NOT_SUPPORTED = 22;
        public static final int NO_CALLBACK_ADDRESS_SET = 17;
        public static final int RESOURCES_UNAVAILABLE = 13;
        public static final int TASK_CANCELLED = 15;
        public static final int TASK_REFUSED = 14;
        public static final int INVALID_STATE = 744;
}
```

Example 6:

```
package org.csapi.jr.se.cc;
public interface CallSuperviseReport {
        public static final int CALL_SUPERVISE_TIMEOUT = 1;
        public static final int CALL_SUPERVISE_CALL_ENDED = 2;
        public static final int CALL_SUPERVISE_TONE_APPLIED = 4;
}
```

C.3.6.3 NumberedSetsOfDataElements (Collections)

In Java, Numbered Set and Numbered List are realised as an array of the data type.

Example 7:

UML	Java Realisation
TpAddressSet	Address[]

C.3.6.4 SequenceOfDataElements (Structures)

Struct data types are represented in Java as public final classes that implement java.io.Serializable, and have:

- each data element made available as a private variable in the class
- a default constructor and a constructor for all values are provided
- accessor and mutator methods are given for each variable
- the first letter of each sequence element name is changed to lower case
- an equals method is provided determining the equality of objects by their content
- a hashCode method is provided supporting the rules for hashCode relative to equals

Example 8:

```
package org.csapi.jr.se;
public final class Address implements java.io.Serializable {
        private AddressPlan plan;
        private String addrString = “”;
        private String name = “”;
        private AddressPresentation presentation;
        private AddressScreening screening;
        private String subAddressString = “”;

        public Address () {
        }

        public Address (AddressPlan plan, String addrString,
                String name, AddressPresentation presentation,
                AddressScreening screening, String subAddressString) {
            this.plan = plan;
            this.addrString = addrString;
            this.name = name;
            this.presentation = presentation;
            this.screening = screening;
            this.subAddressString = subAddressString;
        }
```

```
        public TpAddressPlan getPlan () {
            return (plan);
        }

        public void setPlan (TpAddressPlan plan) {
            this.plan = plan;
        }

        public String getAddrString () {
            return (addrString);
        }

        public void setAddrString (String addrString) {
            this.addrString = addrString;
        }

        … other get and set methods …

        public boolean equals (Object object) {
        // equality logic
        }

        public int hashcode () {
        // hash code calculation
        }
}
```

C.3.6.5 NameValuePair (Enumerations)

NameValuePair data types are represented in Java as public final classes that implement java.io.Serializable, and have:

- two static final data members per name-value pair
- a value returning method, named getValue()
- a name returning method, named getValueText()
- an integer conversion method, named getObject()
- a private constructor
- hashCode and equals implementations

No default constructor is provided. One of the data members per name-value pair has the same name as the name-value pair name. The other has an underscore "_" prepended and is intended for use in switch statements. Values are assigned sequentially, starting with 0.

The getObject() method returns the name-value pair class with the specified value if the specified value corresponds to an element of the name-value pair data type. If the specified value is out of range, an InvalidEnumValueException exception is raised

Example 9:

```
package org.csapi.jr.se;
public final class AddressScreening implements java.io.Serializable {
        private int _value;
        private static int _size = 5;
        private static AddressScreening[] _array = new AddressScreening[_size];


        public static final int _ADDRESS_SCREENING_UNDEFINED = 0;
        public static final AddressScreening ADDRESS_SCREENING_UNDEFINED = new
AddressScreening(_ADDRESS_SCREENING_UNDEFINED);

        public static final int _ADDRESS_SCREENING_USER_VERIFIED_PASSED = 1;
        public static final AddressScreening ADDRESS_SCREENING_USER_VERIFIED_PASSED = new
AddressScreening(_ADDRESS_SCREENING_USER_VERIFIED_PASSED);

        public static final int _ADDRESS_SCREENING_USER_NOT_VERIFIED = 2;
        public static final AddressScreening ADDRESS_SCREENING_USER_NOT_VERIFIED = new
AddressScreening(_ADDRESS_SCREENING_USER_NOT_VERIFIED);
```

```
        public static final int _ADDRESS_SCREENING_USER_VERIFIED_FAILED = 3;
        public static final AddressScreening ADDRESS_SCREENING_USER_VERIFIED_FAILED = new
AddressScreening(_ADDRESS_SCREENING_USER_VERIFIED_FAILED);

        public static final int _ADDRESS_SCREENING_NETWORK =  4;
        public static final AddressScreening ADDRESS_SCREENING_NETWORK = new
AddressScreening(_ADDRESS_SCREENING_NETWORK);


        public int getValue() {
            return _value;
        }

        public String getValueText() {
            switch (_value) {
            case _ADDRESS_SCREENING_UNDEFINED:
                return "ADDRESS_SCREENING_UNDEFINED";
            case _ADDRESS_SCREENING_USER_VERIFIED_PASSED:
                return "ADDRESS_SCREENING_USER_VERIFIED_PASSED";
            case _ADDRESS_SCREENING_USER_NOT_VERIFIED:
                return "ADDRESS_SCREENING_USER_NOT_VERIFIED";
            case _ADDRESS_SCREENING_USER_VERIFIED_FAILED:
                return "ADDRESS_SCREENING_USER_VERIFIED_FAILED";
            case _ADDRESS_SCREENING_NETWORK:
                return "ADDRESS_SCREENING_NETWORK";
            default:
                return "ERROR";
            }
        }

        public boolean equals(Object o) {
        //equality logic
        }

        public int hashCode() {
        //hash code calculation
            return _value;
        }

        public static AddressScreening getObject(int value) throws
org.csapi.jr.se.InvalidEnumValueException {
            if(value >= 0 && value < _size) {
                return _array[value];
            } else {
                throw new org.csapi.jr.se.InvalidEnumValueException();
            }
        }

        private AddressScreening(int value) {
            this._value = value;
            this._array[this._value] = this;
        }

}
```

C.3.6.6 TaggedChoiceOfDataElements (Unions)

Union data types are represented in Java as public final classes that implement java.io.Serializable, and have:

- a default constructor
- a discriminator field
- a discriminator accessor method, named getDiscriminator()
- an accessor and modifier method for each data element, the names of which are derived from choice element name

Conflicting names should be resolved by prefixing the field name with an underscore for getDiscriminator if there is a name clash with the mapped data type name or any of the data element names.

Where choice element type and choice element name are “NULL” and “Undefined”, respectively, a Java Object set as null replaces the NULL. If multiple NULL/Undefined combinations occur in the tagged choice of data elements, the method, setUndefined, will receive the discriminator as a parameter and set _object to null.

Accessor methods shall raise an InvalidUnionAccessorException exception if the expected data element has not been set.

Example 10:

```
package org.csapi.jr.se;
public final class AoCOrder implements java.io.Serializable {
        private CallAoCOrderCategory _discriminator = null;
        private java.lang.Object _object;

        public AoCOrder() {
        }

        public CallAoCOrderCategory getDiscriminator() throws
org.csapi.jr.se.InvalidUnionAccessorException {
            if(_discriminator == null) {
                throw new org.csapi.jr.se.InvalidUnionAccessorException();
            }
            return _discriminator;
        }

        public org.csapi.jr.se.ChargeAdviceInfo getChargeAdviceInfo() throws
org.csapi.jr.se.InvalidUnionAccessorException {
            if (!(_discriminator.equals((CallAoCOrderCategory)
CallAoCOrderCategory.CHARGE_ADVICE_INFO))) {
                throw new org.csapi.jr.se.InvalidUnionAccessorException();
            }
            return ((org.csapi.jr.se.ChargeAdviceInfo) _object);
        }

        public void setChargeAdviceInfo(org.csapi.jr.se.ChargeAdviceInfo value) {
            _discriminator = (CallAoCOrderCategory) CallAoCOrderCategory.CHARGE_ADVICE_INFO;
            _object = value;
        }

        public org.csapi.jr.se.ChargePerTime getChargePerTime() throws
org.csapi.jr.se.InvalidUnionAccessorException {
            if (!(_discriminator.equals((CallAoCOrderCategory)
CallAoCOrderCategory.CHARGE_PER_TIME))) {
                throw new org.csapi.jr.se.InvalidUnionAccessorException();
            }
            return ((org.csapi.jr.se.ChargePerTime) _object);
        }

        public void setChargePerTime(org.csapi.jr.se.ChargePerTime value) {
            _discriminator = (CallAoCOrderCategory) CallAoCOrderCategory.CHARGE_PER_TIME;
            _object = value;
        }

        public java.lang.String getNetworkCharge() throws
org.csapi.jr.se.InvalidUnionAccessorException {
            if (!(_discriminator.equals((CallAoCOrderCategory)
CallAoCOrderCategory.CHARGE_NETWORK))) {
                throw new org.csapi.jr.se.InvalidUnionAccessorException();
            }
            return ((java.lang.String) _object);
        }

        public void setNetworkCharge(java.lang.String value) {
            _discriminator = (CallAoCOrderCategory) CallAoCOrderCategory.CHARGE_NETWORK;
            _object = value;
        }


}
```

C.3.6.7 Exceptions

An exception maps to a constructed exception, providing appropriate constructors and accessor methods for the data contained within the exception. Each exception is defined as a public class extending java.lang.Exception, and containing a private field for each information element contained within the exception.

A default constructor is provided, along with a constructor containing only an embedded exception, a constructor containing a list of the fields in the exception and a constructor that contains the fields plus an embedded exception.

An accessor method is provided for each field, and for the embedded exception.

The following Java Realisations apply to mapping of exceptions:

- PlatformException
- P_XXX_XXX Exceptions
- TpCommonExceptions
- TpCommonExceptions' associated exceptions
- Additional abstract exceptions
- InvalidUnionAccessorException
- InvalidEnumValueException

C.3.6.7.1 PlatformException

PlatformException exception handles local platform and communication problem exceptions.

Example 11:

```
package org.csapi.jr.se;
public class PlatformException extends java.lang.RuntimeException {
        private Throwable _cause;

        public PlatformException () {
            super();
        }

        public PlatformException (String message) {
            super(message);
        }

        public PlatformException (String message, Throwable cause) {
            super(message);
            _cause = cause;
        }

        public PlatformException (Throwable cause) {
            _cause = cause;
        }

        public Throwable getCause() {
            return _cause;
        }
}
```

C.3.6.7.2 P_XXX_XXX Exceptions

P_XXX_XXX exceptions follow the XxxXxxException naming pattern, and inherit from java.lang.Exception.

Example 12:

```
package org.csapi.jr.se;
public class InvalidInterfaceTypeException extends java.lang.Exception {
```

```
        private Throwable _cause;

        public InvalidInterfaceTypeException() {
            super();
        }

        public InvalidInterfaceTypeException(String message) {
            super(message);
        }

        public InvalidInterfaceTypeException(String message,Throwable cause) {
            super(message);
            _cause = cause;
        }

        public InvalidInterfaceTypeException(Throwable cause) {
            _cause = cause;
        }

        public Throwable getCause() {
            return _cause;
        }
}
```

C.3.6.7.3 TpCommonExceptions

The name for TpCommonExceptions exception is made singular, i.e. CommonException, and inherits from java.lang.Exception.

Example 13:

```
package org.csapi.jr.se;
public class CommonException extends java.lang.Exception {
        private Throwable _cause;
        private int exceptionType;
        private String extraInformation;


        public CommonException () {
            super();
        }

        public CommonException (String message) {
            super(message);
        }
        public CommonException (String message, Throwable cause) {
            super(message);
            _cause = cause;
        }

        public CommonException (Throwable cause) {
            _cause = cause;
        }


        public Throwable getCause() {
            return _cause;
        }

        public int getExceptionType() {
            return exceptionType;
        }

        public int setExceptionType() {
            return exceptionType;
        }

        public String getExtraInformation() {
            return extraInformation;
        }

        public String setExtraInformation() {
            return extraInformation;
        }
```

```
}
```

C.3.6.7.4 TpCommonException's associated exceptions

P_XXX_XXX exception types (constants) associated with TpCommonExceptions follow the XxxXxxException naming pattern and inherit from CommonException.

Example 14:

```
package org.csapi.jr.se;
public class ResourcesUnavailableException extends org.csapi.jr.se.CommonException {

        public ResourcesUnavailableException () {
            super();
        }

        public ResourcesUnavailableException (String message) {
            super(message);
        }

        public ResourcesUnavailableException (String message, Throwable cause) {
            super(message, cause);
        }

        public ResourcesUnavailableException (Throwable cause) {
            _cause = cause;
        }


}
```

C.3.6.7.5 Additional abstract exceptions

Additional abstract exceptions (See ETSI ES 202 915-2, Annex D) have been defined which are TpInvalidArgumentException, TpFrameworkException, TpMobilityException, TpDataSessionException, TpMessagingException, TpConnectivityException, TpAccountException, TpPAMException and TpPolicyException and are mapped as follows:

Example 15:

```
package org.csapi.jr.se;
public class InvalidArgumentException extends java.lang.Exception {
        private Throwable _cause;

        public InvalidArgumentException () {
            super();
        }

        public InvalidArgumentException (String message) {
            super(message);
        }

        public InvalidArgumentException (String message, Throwable cause) {
            super(message);
            _cause = cause;
        }

        public InvalidArgumentException (Throwable cause) {
            _cause = cause;
        }

        public Throwable getCause() {
            return _cause;
        }
}
```

C.3.6.7.6 InvalidUnionAccessorException

An additional exception, InvalidUnionAccessorException, is defined which indicates that the expected data element has not been set.

Example 16:

```
package org.csapi.jr.se;
public class InvalidUnionAccessorException extends org.csapi.jr.se.InvalidArgumentException {

        public InvalidUnionAccessorException (){
            super ();
        }

        public InvalidUnionAccessorException (String message){
            super (message);
        }

        public InvalidUnionAccessorException (String message, Throwable cause){
            super (message,cause);
        }

        public InvalidUnionAccessorException (Throwable cause) {
            _cause = cause;
        }

}
```

C.3.6.7.7 InvalidEnumValueException

An additional exception, InvalidEnumValueException, is defined which indicates that an enum data type was accessed with an invalid request value.

Example 17:

```
package org.csapi.jr.se;
public class InvalidEnumValueException extends org.csapi.jr.se.InvalidArgumentException {

        public InvalidEnumValueException () {
            super ();
        }

        public InvalidEnumValueExceptions (String message) {
            super (message);
        }

        public InvalidEnumValueException (String message, Throwable cause) {
            super (message,cause);
        }

        public InvalidEnumValueException (Throwable cause) {
            _cause = cause;
        }

}
```

C.4 J2SE Specific Conventions

The UML interfaces are represented by Java public interfaces; those interfaces that inherit from other interfaces are represented in Java as extending that interface. The Java realisations of OSA/Parlay SCFs use an Event Listener design pattern while the Framework uses the Callback pattern.

This annex provides the information on realisation of the Java developer API including:

- How Java APIs are realised from Parlay UML
- Where the listener pattern is used, new classes to be generated from the UML

- Changes required to data types and methods to support correlation using object references
- Use of hierarchical exceptions

C.4.1 Removal of "Tp" Prefix

The UML data types labelled with the prefix “Tp” are represented in Java without this prefix.

Example 18:

UML	Java Realisation
TpCallAppInfo	CallAppInfo

In the case of name collisions between data types and interfaces as with IpTerminalCapabilities and IpService the UML data types labelled with the prefix “Tp” are represented in Java with an alternative prefix “Type”.

Example 19:

UML	Java Realisation
IpTerminalCapabilities	TerminalCapabilities
TpTerminalCapabilities	TypeTerminalCapabilities

The above example is based in conjunction with C.4.3 Removal of "Ip" Prefix.

C.4.2 Constants

The UML constants labelled with the prefix “P_” are represented in Java without this prefix.

Example 20:

UML Constant	Java Constant
P_NO_CALLBACK_ADDRESS_SET	NO_CALLBACK_ADDRESS_SET

C.4.3 Removal of "Ip" prefix

The "Ip" prefix is removed in the Java realisation of UML interfaces.

Example 21:

UML	Java
IpCallControlManager	CallControlManager

C.4.4 Mapping of IpInterface

IpInterface interface is represented by the CsapiInterface interface. This is a ‘marker’ interface, in that it contains no methods, but provides a common interface for related interfaces to inherit from. All interfaces to be serializable; this can be done by CsapiInterface extending Serializable.

Example 22:

```
package org.csapi.jr.se;
        public interface CsapiInterface extends Serializable{
        }
```

C.4.5 Mapping of IpService

IpService interface is represented by the Java Service interface. This provides a common interface for related interfaces to inherit from.

Example 23:

Service Interface:

```
package org.csapi.jr.se;
public interface Service extends CsapiInterface {
        public final static int IN_SERVICE_STATE=0 ;
        public final static int  OUT_OF_SERVICE_STATE=1;

        void addServiceStateChangeListener(ServiceStateChangeListener listener)
        int getServiceState();
        void removeServiceStateChangeListener( ServiceStateChangeListener listener) ;
}
```

Listener interface:

```
package org.csapi.jr.se;
public interface ServiceStateChangeListener {
        void onOutOfService(OutOfServiceEvent event);
}
```

Event class:

```
package org.csapi.jr.se;
public class OutOfServiceEvent extends EventObject {}
```

C.4.6 Mapping of UML Operations

The UML operations are represented in Java as methods.

Exceptions that can be raised by UML operations are represented in Java with the throws clause and the Java Realisation of the UML Exceptions.

UML “in” parameters, represented by “in ” preceding the parameter type are represented in Java without this clause.

Example 24:

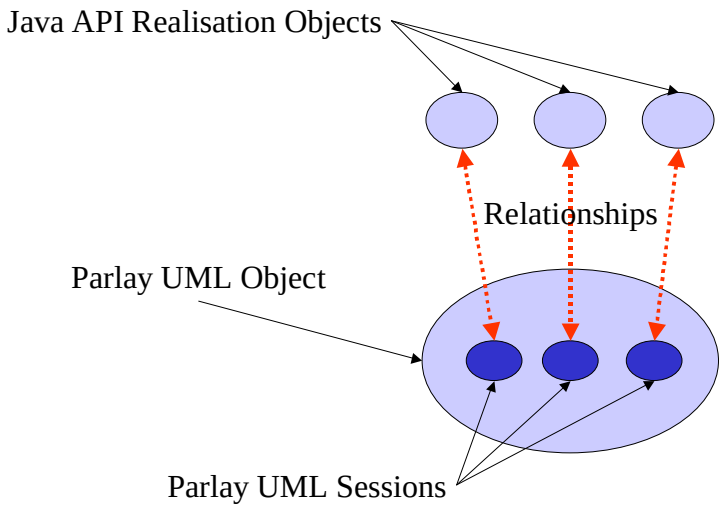
```
public void managerResumed ();

public CsapiInterface obtainInterface (InterfaceName interfaceName) throws
InvalidInterfaceNameException;

public Service createServiceManager (ClientAppID application, ServicePropertyList serviceProperties,
ServiceInstanceID serviceInstanceID);
```

The above example method signatures are based on generic mapping of interfaces, exceptions and data types.

C.4.7 Mapping of TpSessionID

The UML TpSessionID data types will be hidden in the J2SE APIs (and optionally supported by the underlying Java implementation). Consequently, the TpSessionIDSet data type and IpService.setCallbackWithSessionID() method are superfluous. Also, structures with only TpSessionID and interface references (e.g. TpCallIdentifier) are no longer necessary and references to these structures should be replaced by just the reference to the interface. For data types that contain TpSessionID the Java API Realisation object replaces theTpSessionID.

The following figure shows how Java API Realisation objects relate to Parlay UML objects and sessions. How this is realised in the adaptors is implementation dependent.



C.4.8 Mapping of TpAssignmentID to the creation of an Activity object.

The UML TpAssignmentID data types, which differentiate between multiple parallel asynchronous method invocations (activities) on the same ("parent") interface, are deleted and replaced with createXxx methods (one for each parallel asynchronous activity) that create ("child") activity interfaces. Where this would result in method names of the pattern createCreateXxx, this should be changed to method names with the pattern createXxx. Associated listeners would then remove the Create prefix from their name. These activity interfaces, in addition to possibly supporting other methods, will support one of the previously mentioned multiple parallel asynchronous method invocations. Hence, the Java API realisation creates multiple (activity) objects and invokes a single request per object rather than creating a single object and invoking multiple requests on that object, each request being differentiated using the TpAssignmentID value. The results of the asynchronous method invocation will be handled by the activity interface's listener interface. To create the activity interface, the original IpXxx interface (to be named Xxx) will replace its parallel supporting asynchronous method invocations, yyyYyyReq, with createYyyYyy methods that take no parameters but returns the activity interface, YyyYyy. Where this would result in method names of the pattern createCreateXxx, this should be changed to method names with the pattern createXxx. Associated listeners would then remove the Create prefix from their name. The activity interface will extend Activity interface (see next rule), have a simple FSM, the addYyyYyyListener, removeYyyYyyListener and the asynchronous method that previously supported a parallel capability (typically named yyyYyyReq, but also yyyYyyStop).

An Activity interface, packaged in org.csapi.jr.se, is added as a parent to all activity interfaces. An application may add listeners of type ActivityStateChangeListener to an Activity if it wishes be explicitly informed when the activity becomes invalid.

The YyyYyyListener activity listener interfaces will extend java.util.EventListener. The asynchronous methods of previously named IpAppXxx, typically labelled yyyYyyRes and yyyYyyErr but also yyyYyy, will be renamed onYyyYyyRes and onYyyYyyErr but also onYyyYyy. Each method will have an event parameter, typically labelled YyyYyyResEvent and YyyYyyErrEvent, but also YyyYyyEvent. Events will be classes that extend java.util.EventObject and contain a public constructor (with multiple parameters – one per class carried by the event) and a number of public getter methods (one per "gettable" class carried by the event). As a result of adding activity listener interfaces, this may cause the requirement for the original IpAppXxx to disappear, since the yyyYyyRes and yyyYyyErr methods will effectively be ported to the activity listener interfaces.

For data types that contain TpAssignmentID the activity object replaces the TpAssignmentID.

Example 25:

Activity Interface:

```
package org.csapi.jr.se;
public interface Activity {
        public final static int IDLE_STATE = 0;
        public final static int ACTIVE_STATE = 1;
```

```
        public final static int INVALID_STATE = 2;
        public int getState();
        public void addActivityStateChangeListener(ActivityStateChangeListener listener);
        public void removeActivityStateChageListener(ActivityStateChangeListener listener);
}
```

Activity Listener Interface and Event class:

```
package org.csapi.jr.se;
public interface ActivityStateChangeListener {
        onInvalidState (InvalidActivityEvent event)
}

public class InvalidActivityEvent extends EventObject {
}
```

Parent interface:

```
package org.csapi.jr.se.mmm.ul;
public interface UserLocation {
        public LocationReport createLocationReport();
        public ExtendedLocationReport createExtendedLocationReport();
        public PeriodicLocationReporting createPeriodicLocationReporting();
}
```

Child Interface:

```
package org.csapi.jr.se.mm.ul;
public interface LocationReport extends Activity {
        public void addLocationReportListener(LocationReportListener listener)
        public void removeLocationReportListener(LocationReportListener listener)
        public void locationReportReq(Address[] users) throws …
}
```

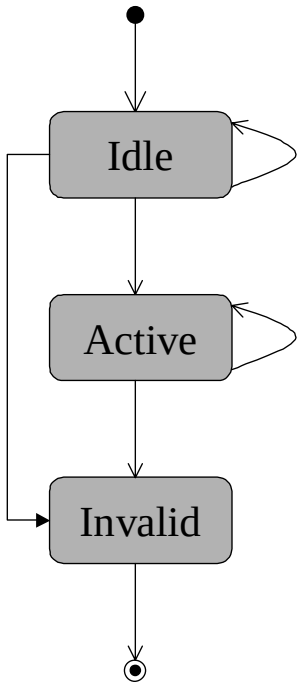
Listener Interface:

```
package org.csapi.jr.se.mm.ul;
public interface LocationReportListener extends CsapiInterface, java.util.EventListener {

        public void onLocationReportRes(LocationReportResEvent event);
        public void onLocationReportErr(LocationReportErrEvent event);
}
```

Event classes:

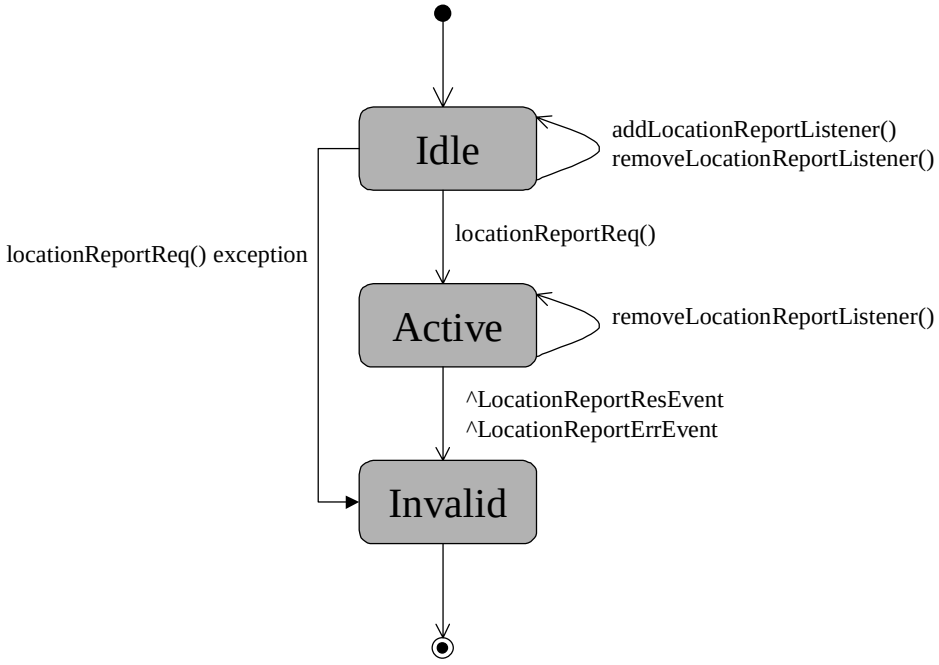
```
package org.csapi.jr.se.mmm.ul;
public class LocationReportResEvent extends java.util.EventObject{
        // with a public UserLocation[] constructor and a public getter
        // method for the parameter of the event
}

public classLocationReportErrEvent extends java.util.EventObject {
        // with a public MobilityError and MobilityDiagnostic constructor
        // and two public getter methods, one for each of the parameters
        // of the event
}
```

The Finite State Model for the Activity interface is given below:

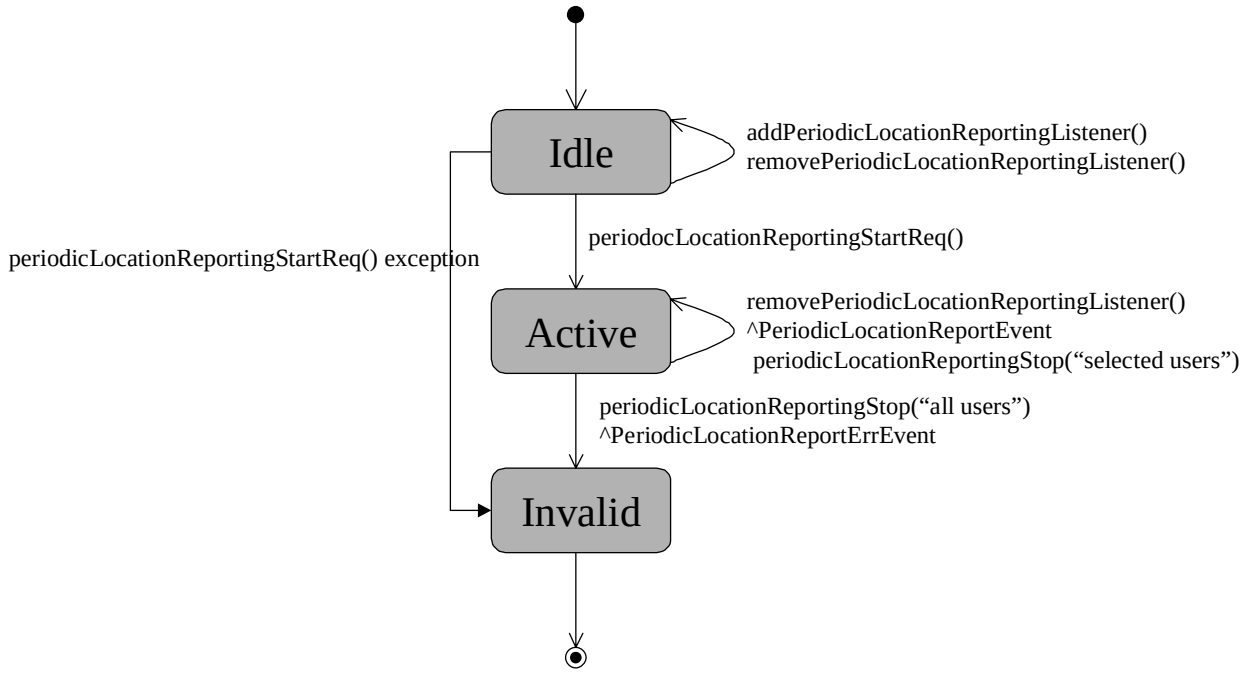


This interface specifies an activity, which might be provided by a service. An activity has three states: "idle", "active" and "invalid". The initial state is "idle" and here the listeners should be registered. It performs in the "active" state. It enters the "invalid" state when it has fulfilled its task or a fatal error occurred. In special cases state transition from "idle" to "invalid" is possible.

An example activity interface FSM is given below for a single activity request with a single response:



An example activity interface FSM is given below for a single activity request with repeating responses:



C.4.9 Callback Rule

The UML callback design pattern for non client-to-service interfaces (Parlay interface numbers 1, 3, 4, 5 and 6 [Fig 1]) is represented in Java with the callback design pattern. The UML callback design pattern for client-to-service interfaces (Parlay interface number 2 [Fig 1]) is represented in Java with the event listener design pattern.

The UML client-to-service interfaces (Parlay interface number 2) with the IpAppXxxx naming convention are represented in Java with the XxxxListener naming convention.

The IpService.setCallback method can be deleted; the interfaces that inherited the setCallback method now have associated addXxxxListener and removeXxxxListener methods. According to the *TpSessionID* mapping, IpService.setCallbackWithSessionID() method is deleted.

The XxxxListener listener interfaces will extend java.util.EventListener. The asynchronous methods of previously named IpAppXxxx, typically labelled yyyyYyyyRes and yyyyYyyyyErr but also yyyyYyyy, will be renamed onYyyyYyyyRes and onYyyyYyyyErr but also onYyyyYyyy. Each method will have an event parameter, typically labelled YyyyYyyyResEvent and YyyyYyyyErrEvent, but also YyyyYyyyEvent. Events will be classes that extend java.util.EventObject and contain a private constructor (with multiple parameters – one per class carried by the event) and a number of public getter methods (one per "gettable" class carried by the event). Events are read-only and serializable.

Example 26:

Listener Interface:

```
package org.csapi.jr.se.cc.mpccs;

MultiPartyCallListener extends CsapiInterface, java.util.EventListener{

public void onGetInfoRes(GetInfoResEvent event)
public void onGetInfoErr(GetInfoErrEvent event)
public void onSuperviseRes(SuperviseResEvent event)
public void onSuperviseErr(SuperviseErrEvent event)
public void onCallEnded(CallEndedEvent event)
public void onCreateAndRouteCallLegErr(CreateAndRouteCallLegErrEvent event)
}
```

MuliPartyCall Interface additional methods:

```
public void addMultiPartyCallListener(MultiPartyCallListener multiPartyCallListener);
```

```
public void removeMultiPartyCallListener(MultiPartyCallListener multiPartyCallListener);
```

C.4.10 Factory Rule

The following Factory class allows applications to obtain proprietary peer API objects. The term "peer" is Java nomenclature for a particular platform-specific implementation of a Java interface.

Example 27:

```
package org.csapi.jr.se.fw;
import org.csapi.jr.se.PeerUnavailableException;
import org.csapi.jr.se.InvalidArgumentException;
import org.csapi.jr.se.ResourcesUnavailableException;
import org.csapi.jr.se.fw.access.tsm.Initial;
import java.util.*;

public class InitialFactory {
        private static InitialFactory myFactory;
        private static String className = null;
        private static String lang     = "en";
        private static String cntry    = "US";

        private InitialFactory() {
        }

        public synchronized Initial createInitial(String initialPeerReference) throws
PeerUnavailableException, ResourcesUnavailableException , InvalidArgumentException {
            Locale currentLocale;
            ResourceBundle messages;
            String tryMessage;

            try {
                currentLocale = new Locale(lang, cntry);
                messages = ResourceBundle.getBundle("InitialFactoryBundle", currentLocale);

                // Validate all used values before using them later
                // avoiding error text exception to hide the real exception

                tryMessage = messages.getString("InitialPeerReferenceNull");
                tryMessage = messages.getString("InitialInstFailure");
                tryMessage = message.getString("DestroyInitialFailure");
            }
            catch (Exception e) {
                throw new ResourcesUnavailableException ("Localisation failed to be initialized");
            }

            if (initialPeerReference == null) {
                String errmsg = messages.getString("InitialPeerReferenceNull");
                throw new InvalidArgumentException (errmsg);
            }

            try {
                Class c = Class.forName (getImplementationClassName ());
                if(initialPeerReference.equals(“”)){
                    // Creates a new instance of the Object class
                    // using default constructor
                    return (Initial)c.newInstance ();
                }

                Class[] paramTypes = {initialPeerReference.getClass()};
                java.lang.reflect.Constructor ctor =
                c.getConstructor(paramTypes);
                Object[] params = {initialPeerReference};
                return (Initial) ctor.newInstance(params);
            } catch (Exception e) {
                String errmsg = messages.getString("InitialInstFailure");
                throw new PeerUnavailableException (errmsg);
            }
        }

        public synchronized static InitialFactory getInstance() {
            if (myFactory == null) {
                myFactory = new InitialFactory ();
            }
```

```
            return myFactory;
        }

        public String getImplementationClassName () {
            return className;
        }

        public static void setImplementationClassName (String className) {
            this.className = className;
        }

        public synchronized static void setLocale(String language, String country) {
            if (langauage == null) {
                lang = "en";
            }
            else {
                lang = language;
            }

            if (country == null) {
                cntry = "US";
            }
            else {
                cntry = country;
            }
        }

        public void destroyInitial(Initial initialInstance) {
            if (initialInstance == null) {
                return;
            }

            try {
                delete initialInstance;
            } catch (Exception e) {
                String errmsg = messages.getString("DestroyInitialFailure");
                throw new RuntimeException(errmsg);
            }
        }
}
```

C.4.11 J2SE Specific Exceptions

Exceptions in this section are only applicable within a J2SE environment.

C.4.11.1 PeerUnavailableException

PeerUnavailableException indicates failure to access an implementation of the Initial interface.

Example 28:

```
public class PeerUnavailableException extends java.lang.Exception {
        private Throwable _cause;
        public PeerUnavailableException () {
            super();
        }

        public PeerUnavailableException (String message) {
            super(message);
        }

        public PeerUnavailableException (String message, Throwable cause) {
            super(message);
            _cause = cause;
        }

        public PeerUnavailableException (Throwable cause) {
            _cause = cause;
        }

        public Throwable getCause() {
            return _cause;
        }
```

```
}
```

C.4.11.2 IllegalStateException

IllegalStateException exception signals that a method has been invoked at an illegal or inappropriate time.

Example 29:

```
package org.csapi.jr.se;
public class IllegalStateException extends Exception {

        private int _state;
        private Object _object;

        public IllegalStateException(Object object, int state) {
            super();
            _object = object;
            _state  = state;
        }

        public Illegal StateException(Object object, int state, String s) {
            super(s);
            _object = object;
            _state  = state;
        }

        public Object getObject() {
            return _object;
        }

        public int getState() {
            return _state;
        }
}
```

C.4.12 User Interaction Specific Rules

C.4.12.1 Interfaces representing UML IpUI and IpUICall Rule

The following mappings take account of the fact that when the TpAssignmentID rule is applied the Java interfaces representing UML IpUICall does not extend the Java interfaces representing UML IpUI.

Java UIGeneric replaces the UML IpUI. Methods common to both the Java UIGeneric and Java UICall are pulled up into a super-interface called UI. UML IpAppUI and IpAppUiCall interfaces are replaced by a UIListener interface.

C.4.12.2 Naming Collisions of GUI and CUI Activities Rule

Naming collisions that arise through GUI and CUI activities e.g. XXX, having the same name will be dealt with by prefixing the Call Related UI activity by “CallRelated”. Methods to create the activity will become createCallRelatedXXX().

C.5 J2EE Specific Conventions

J2EE supports both remote and local interfaces. ~~To support one set of data type definitions that will work with both remote and local interfaces, an inheritance approach is used where the remote interface is a super interface to the local interface, supporting Java language rules and accomplishing this goal. This is transparent to the application writer.~~

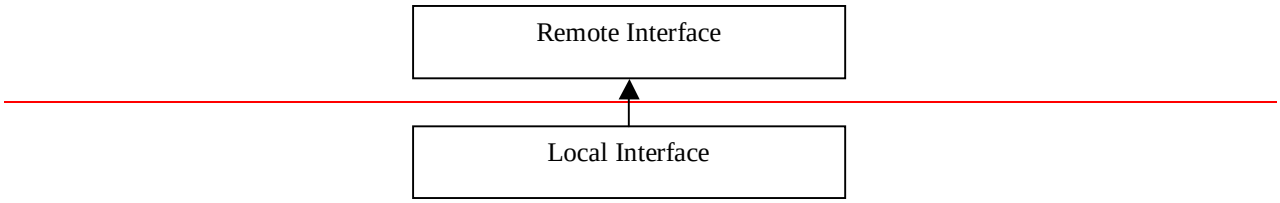


Figure :

C.5.1 Interface Fields in Data Types

A data type as a field must reference the interface as defined in the org.csapi.jr.ee package. This requirement addresses the Java language rule that allows exceptions to be removed in a sub-interface, but does not allow an exception to be added to a sub-interface.

Example 30:

```
package org.csapi.jr.ee;

public class TpMultiPartyCallIdentifier {
        org.csapi.jr.ee.remote.IpMultiPartyCall callReference;
        int callSessionID;
}
```

C.5.12 Serialization UID

All data types will have a serialVersionUID defined within its definition, as a static final long value.

Example 301:

```
package org.csapi.jr.ee;
public final class TpAddress implements java.io.Serializable {
        static final long serialVersionUID = 989898989898L;

        private TpAddressPlan plan;
        … remainder of class …
}
```

C.5.23 Remote Interface Definitions

C.5.23.1 IpInterface

This interface implements java.io.Serializable. Since it is the root interface for all other interfaces, this makes all defined interfaces serializable.

Example 312:

```
public interface IpInterfaceCall extends java.io.SerializableIpService
```

C.5.23.23 Methods for Remote Interfaces

A public method is defined within a remote interface for each method defined in the specification, with zero or one output specified as the return value, and all other parameters listed without any input marker. Each method will return java.rmi.RemoteException in addition to other exceptions, if any.

Example 323:

```
public void deassignCall (int callSessionID) throws java.rmi.RemoteException,
org.csapi.jr.ee.TpCommonException, org.csapi.jr.ee.InvalidSessionIdException;
```

C.5.34 Local Interface Definitions

~~C.5.4.1 Parent Interface~~

~~Each local interface extends its corresponding remote interface.~~

~~Example 34:~~

```
public interface IpCall extends org.csapi.jr.ee.remote.IpCall
```

~~C.5.4.2 Interface Inheritance~~

~~Interfaces in Java may extend each other using the 'extends' keyword. Where an interface is defined as inheriting from one or more other interfaces, it will declare the list of interfaces it inherits from in its interface declaration.~~

~~Example 35:~~

```
public interface IpCall extends org.csapi.jr.ee.remote.IpCall, IpService
```

C.5.34.13 Methods for Local Interfaces

A public method is defined within a local interface for each method defined in the specification, with zero or one output specified as the return value, and all other parameters listed without any input marker.

Example 336:

```
public void deassignCall (int callSessionID) throws org.csapi.jr.ee.TpCommonExceptions,
org.csapi.jr.ee.InvalidSessionIdException;
```

********** End of Change # 1 **********

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 (Correction to IDL namespace to align with that of ETSI and Parlay equivalent APIs: Change org.open_service_access root namespace to org.csapi) (N5-010267)	4.0.0	4.1.0
Sep 2001	CN_13	NP-010464	002	--	Changing references to JAIN	4.1.0	4.2.0
Dec 2001	CN_14	NP-010594	003	--	Replace Out Parameters with Return Types	4.2.0	4.3.0
Dec 2001	CN_14	NP-010594	004	--	Remove the perception that the OSA API only uses CORBA for its transport mechanism	4.2.0	4.3.0
Mar 2002	--	--	--	--	Editorial update (no CR) following Hong Kong CN5#16	4.3.0	4.3.1
Jun 2002	CN_16	NP-020181	005	--	Addition of support for Java API technology realisation	4.3.1	5.0.0
Jun 2002	CN_16	NP-020182	006	--	Addition of support for WSDL realisation	4.3.1	5.0.0
Jun 2002	CN_16	NP-020184	007	--	Adding the full naming convention for exceptions	4.3.1	5.0.0
Jun 2002	CN_16	NP-020184	008	--	Correction of References in OSA specifications	4.3.1	5.0.0
Jun 2002	CN_16	NP-020184	009	--	Addition of text describing the technology realisations of the Parlay/OSA specification	4.3.1	5.0.0
Sep 2002	CN_17	NP-020427	010	--	Addition to ObjectRef description in WSDL Mapping Rules	5.0.0	5.1.0
Sep 2002	CN_17	NP-020427	011	--	Addition of sequence tag to Choice types	5.0.0	5.1.0
Sep 2002	CN_17	NP-020427	012	--	Replace all occurrences of the xsd:anyURI type to xsd:string	5.0.0	5.1.0
Sep 2002	CN_17	NP-020427	013	--	Correction to Namespace mapping in WSDL Mapping Rules	5.0.0	5.1.0
Sep 2002	CN_17	NP-020427	014	--	Correction to xmlns:wsdl Namespace	5.0.0	5.1.0
Sep 2002	CN_17	NP-020427	015	--	Prepend class name to <message> name.	5.0.0	5.1.0
Sep 2002	CN_17	NP-020427	016	--	Correction to void return types in WSDL Mapping Rules	5.0.0	5.1.0
Sep 2002	CN_17	NP-020427	017	--	Add missing CORBA realization rules in Part 1	5.0.0	5.1.0
Sep 2002	CN_17	NP-020427	018	--	Add general introduction to the OSA APIs in Part 1	5.0.0	5.1.0
Sep 2002	CN_17	NP-020395	020	--	Add text to clarify relationship between 3GPP and ETSI/Parlay OSA specifications	5.0.0	5.1.0
Mar 2003	CN_19	--	--	--	Editorial update (no CR) following Bangkok CN5#22 (Introduction, Reference Titles)	5.1.0	5.1.1
Jun 2003	CN_20	NP-030298	022	1	Removal of un-used references	5.1.1	5.2.0
Jun 2003	CN_20	NP-030239	023	-	Correction to Java Realisation Annex	5.1.1	5.2.0

CR-Form-v7

CHANGE REQUEST

⌘ 29.198-02 CR 036 ⌘rev - ⌘ Current version: 5.3.0 ⌘

*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 ☒

Title:	⌘	Correction to Java Realisation Annex
Source:	⌘	CN5 AePONA – Eamonn Murray
Work item code:	⌘	OSA2
Date:	⌘	08/09/2003
Category:	⌘	**F**
Release:	⌘	*REL-5*

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

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

Reason for change:	⌘	Correction to Annex C of the current specification. The current Annex references Jain SPA. Jain SPA is no longer a supported activity or deliverable. Replace the current Annex with the Parlay Java Realisation as an informative Annex to the body of OSA API specfication deliverables.
Summary of change:	⌘	Replace the current Annex C that refers to Jain SPA as the informative Java Realisation with the Java Realisation rulebook produced by the Parlay Java Realisation Workgroup.
Consequences if not approved:	⌘	The API specification will reference a realisation that is no longer supported or valid.

Clauses affected:	⌘	Annex C			
Other specs affected:	⌘	**Y** / **N**		⌘	Rel-5 29.198-xy
		X /	Other core specifications		
		/ **X**	Test specifications		
		/ **X**	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/specs/CR.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.

KEEP the History box of the TS to be changed (see end of the present document)

********** Start of Change # 1 **********

~~# Annex C (informative):~~
~~# Java API Description of the Common Data definitions~~

~~The Java API representation of this specification can be obtained from the following URL:~~

~~- JAIN Common (http://jcp.org/jsr/detail/145.jsp)~~

~~Each JSR webpage contains a table identifying the relationships between the different versions of the Parlay, ETSI/OSA, 3GPP/OSA and JAIN SPA specifications. In addition, each JAIN SPA specification version indicates to which Parlay, ETSI/OSA and 3GPP/OSA specification versions it corresponds to.~~

Annex C (informative):
Java API Description of the Common Data definitions

The Java API realisation of this specification is produced in accordance with the Java Realisation rules defined in Part 1 of this specification. These rules aim to deliver for Java, a developer API, provided as a realisation, supporting a Java API that represents the UML specifications. The rules support the production of both J2SE and J2EE versions of the API from the common UML specifications.

The J2SE representation of this specification is provided as Java Code, contained in archive 2919802J2SE.ZIP that accompanies the present document.

The J2EE representation of this specification is provided as Java Code, contained in archive 2919802J2EE.ZIP that accompanies the present document.

********** End of Change # 1 **********

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 (Exception handling mechanism without ambiguity - Replace TpGeneralException and TpResultInfo with detailed exception classes which can be thrown for each method (N5-010261)	4.0.0	4.1.0
Jun 2001	CN_12	NP-010333	002	--	Introduction of TpOctet (In order to make sure that some data is sent over the "distributed wire" untouched a new data type is needed) (N5-010304)	4.0.0	4.1.0
Sep 2001	CN_13	NP-010465	003	--	Changing references to JAIN	4.1.0	4.2.0
Sep 2001	CN_13	NP-010465	004	--	Clarification of common exceptions	4.1.0	4.2.0
Sep 2001	CN_13	NP-010465	005	--	Invalid parameter value exception for SLA violation	4.1.0	4.2.0
Sep 2001	CN_13	NP-010465	006	--	Storing eventCriteria	4.1.0	4.2.0
Dec 2001	CN_14	NP-010595	007	--	Replace Out Parameters with Return Types	4.2.0	4.3.0
Dec 2001	CN_14	NP-010595	008	--	Correction to Common Data (CD)	4.2.0	4.3.0
Dec 2001	CN_14	NP-010595	009	--	Correction to values of TpAddressPlan	4.2.0	4.3.0
Mar 2002	CN_15	NP-020104	010	--	Ambiguous definition of TpAssignmentID	4.3.0	4.4.0
Mar 2002	CN_15	NP-020104	011	--	Data type alignment in the common data types	4.3.0	4.4.0
Jun 2002	CN_16	NP-020185	011	--	Allowing the use of tel URL in TpAddressPlan	4.4.0	5.0.0
Jun 2002	CN_16	NP-020185	012	--	Adding TpInt64 in order to aling with the new Rel-5 TS 29.198-14	4.4.0	5.0.0
Jun 2002	CN_16	NP-020185	013	--	Addition of undefined Data types: TpStringList and TpStringSet	4.4.0	5.0.0
Jun 2002	CN_16	NP-020181	014	--	Addition of support for Java API technology realisation	4.4.0	5.0.0
Jun 2002	CN_16	NP-020182	015	--	Addition of support for WSDL realisation	4.4.0	5.0.0
Jun 2002	CN_16	NP-020185	016	--	Deletion of P_SET_LENGTH_EXCEEDED	4.4.0	5.0.0
Jun 2002	CN_16	NP-020185	017	--	Removal of MIDL	4.4.0	5.0.0
Jun 2002	CN_16	NP-020185	018	--	Revise the scope of TpSessionID and TpAssignmentID	4.4.0	5.0.0
Jun 2002	CN_16	NP-020185	019	--	Deprecate P_ADDRESS_PLAN_MSMAIL	4.4.0	5.0.0
Jun 2002	CN_16	NP-020185	020	--	Addition of support for an Exception Hierarchy	4.4.0	5.0.0
Jun 2002	CN_16	NP-020185	021	--	Addition of type TpVersion in common data	4.4.0	5.0.0
Sep 2002	CN_17	NP-020395	022	--	Add text to clarify relationship between 3GPP and ETSI/Parlay OSA specifications	5.0.0	5.1.0
Oct 2002	--	--	--	--	Added the two missing attachments (osa.idl contained in archive 2919802IDL.ZIP) (osa.wsdl contained in archive 2919802WSDL.ZIP)	5.1.0	5.1.1
Mar 2003	CN_19	NP-030018	025	--	Clarification on uniqueness of assignmentID	5.1.1	5.2.0
Mar 2003	CN_19	NP-030018	027	--	Correction to P_INVALID_STATE value	5.1.1	5.2.0
Mar 2003	CN_19	NP-030018	029	--	Addition of Support of National Numbering Plans	5.1.1	5.2.0
Mar 2003	CN_19	NP-030027	030	--	Addition of Numbered List of Data Elements definition	5.1.1	5.2.0
Mar 2003	CN_19	NP-030027	031	--	Correction of Exception Hierarchy to align with Java Realisation	5.1.1	5.2.0
Mar 2003	CN_19	NP-030027	032	--	Promotion of TpDataSessionQosClass dat type definition to the Common Data Types	5.1.1	5.2.0
Jun 2003	CN_20	NP-030236	034	--	Correction of SIP Address wildcard rules	5.2.0	5.3.0
Jun 2003	CN_20	NP-030240	035	--	Add the type TpURN to Common Data Types	5.2.0	5.3.0

CR-Form-v7

CHANGE REQUEST

⌘ 29.198-03 CR 085 ⌘rev - ⌘ Current version: 5.3.0 ⌘

*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 ☒

Title:	⌘	Correction to Java Realisation Annex
Source:	⌘	CN5 AePONA – Eamonn Murray
Work item code:	⌘	OSA2
Date:	⌘	08/09/2003
Category:	⌘	**F**
Release:	⌘	*REL-5*

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

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

Reason for change:	⌘	Correction to Annex C of the current specification. The current Annex references Jain SPA. Jain SPA is no longer a supported activity or deliverable. Replace the current Annex with the Parlay Java Realisation as an informative Annex to the body of OSA API specfication deliverables.
Summary of change:	⌘	Replace the current Annex C that refers to Jain SPA as the informative Java Realisation with the Java Realisation rulebook produced by the Parlay Java Realisation Workgroup.
Consequences if not approved:	⌘	The API specification will reference a realisation that is no longer supported or valid.

Clauses affected:	⌘	Annex C			
Other specs affected:	⌘	**Y**	**N**		
		X		Other core specifications	⌘ Rel-5 29.198-xy
			X	Test specifications	
			X	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/specs/CR.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.

KEEP the History box of the TS to be changed (see end of the present document)

********** Start of Change # 1 **********

~~# Annex C (informative):~~
~~# Java API Description of the Framework~~

~~The Java API representation of this specification can be obtained from the following URLs:~~

~~- JAIN SPA Framework Access Session (http://jcp.org/jsr/detail/24.jsp)~~

~~- JAIN SPA Framework to Application (http://jcp.org/jsr/detail/119.jsp)~~

~~Each JSR webpage contains a table identifying the relationships between the different versions of the Parlay, ETSI/OSA, 3GPP/OSA and JAIN SPA specifications. In addition, each JAIN SPA specification version indicates to which Parlay, ETSI/OSA and 3GPP/OSA specification versions it corresponds to.~~

Annex C (informative):
Java API Description of the Framework

The Java API realisation of this specification is produced in accordance with the Java Realisation rules defined in Part 1 of this specification. These rules aim to deliver for Java, a developer API, provided as a realisation, supporting a Java API that represents the UML specifications. The rules support the production of both J2SE and J2EE versions of the API from the common UML specifications.

The J2SE representation of this specification is provided as Java, contained in archive 2919803J2SE.ZIP that accompanies the present document.

The J2EE representation of this specification is provided as Java, contained in archive 2919803J2EE.ZIP that accompanies the present document.

********** End of Change # 1 **********

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

CR-Form-v7

CHANGE REQUEST

⌘ 29.198-04-1 CR 007 ⌘rev - ⌘ Current version: 5.3.0 ⌘

*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 ☒

Title:	⌘	Correction to Java Realisation Annex
Source:	⌘	CN5 AePONA – Eamonn Murray
Work item code:	⌘	OSA2
Date:	⌘	08/09/2003
Category:	⌘	**F**
Release:	⌘	*REL-5*

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

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

Reason for change:	⌘	Correction to Annex C of the current specification. The current Annex references Jain SPA. Jain SPA is no longer a supported activity or deliverable. Replace the current Annex with the Parlay Java Realisation as an informative Annex to the body of OSA API specfication deliverables.
Summary of change:	⌘	Replace the current Annex C that refers to Jain SPA as the informative Java Realisation with the Java Realisation rulebook produced by the Parlay Java Realisation Workgroup.
Consequences if not approved:	⌘	The API specification will reference a realisation that is no longer supported or valid.

Clauses affected:	⌘	Annex C

		Y	N			
Other specs affected:	⌘	X		Other core specifications	⌘	Rel-5 29.198-xy
			X	Test specifications		
			X	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/specs/CR.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.

KEEP the History box of the TS to be changed (see end of the present document)

********** Start of Change # 1 **********

~~Annex C (informative):~~
~~Java API Description of the Call Control SCFs~~

~~The Java API representation of this specification can be obtained from the following URL:~~

~~- Java Call Control (http://jcp.org/jsr/detail/21.jsp)~~

~~Each JSR webpage contains a table identifying the relationships between the different versions of the Parlay, ETSI/OSA, 3GPP/OSA and JAIN SPA specifications. In addition, each JAIN SPA specification version indicates to which Parlay, ETSI/OSA and 3GPP/OSA specification versions it corresponds to.~~

Annex C (informative):
Java API Description of the Call Control SCFs

The Java API realisation of this specification is produced in accordance with the Java Realisation rules defined in Part 1 of this specification. These rules aim to deliver for Java, a developer API, provided as a realisation, supporting a Java API that represents the UML specifications. The rules support the production of both J2SE and J2EE versions of the API from the common UML specifications.

The J2SE representation of this specification is provided as Java Code, contained in archive 2919804-1J2SE.ZIP that accompanies the present document.

The J2EE representation of this specification is provided as Java Code, contained in archive 2919804-1J2EE.ZIP that accompanies the present document.

********** End of Change # 1 **********

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	1.0.0
June 2001	CN_12	NP-010327	--	--	Approved at TSG CN#12 and placed under Change Control	2.0.0	4.0.0
Sep 2001	CN_13	NP-010467	001	--	Changing references to JAIN	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	002	--	Correction of text descriptions for methods enableCallNotification and createNotification	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	003	--	Specify the behaviour when a call leg times out	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	004	--	Removal of Faulty state in MPCCS Call State Transition Diagram and method callFaultDetected in MPCCS in OSA R4	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	005	--	Missing TpCallAppInfoSet description in OSA R4	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	006	--	Redirecting a call leg vs. creating a call leg clarification in OSA R4	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	007	--	Introduction of MPCC Originating and Terminating Call Leg STDs for IpCallLeg	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	008	--	Corrections to SetChargePlan() Addition of PartyToCharge parmeter	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	009	--	Corrections to SetChargePlan()	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	010	--	Remove distinction between final- and intermediate-report	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	011	--	Inclusion of TpMediaType	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	012	--	Corrections to GCC STD	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	013	--	Introduction of sequence diagrams for MPCC services	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	014	--	The use of the REDIRECT event needs to be illustrated	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	015	--	Corrections to SetCallChargePlan()	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	016	--	Add one additional error indication	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	017	--	Corrections to Call Control – GCCS Exception handling	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	018	--	Corrections to Call Control – Errors in Exceptions	4.0.0	4.1.0
Dec 2001	CN_14	NP-010597	019	--	Replace Out Parameters with Return Types	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	020	--	Removal of time based charging property	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	021	--	Make attachMedia() and detachMedia() asynchronous	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	022	--	Correction of treatment datatype in superviseReq on call leg	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	023	--	Corrections to Call Control Data Types	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	024	--	Correction to Call Control (CC)	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	025	--	Amend the Generic Call Control introductory part	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	026	--	Correction in TpCallEventType	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	027	--	Addition of missing description of RouteErr()	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	028	--	Misleading description of createAndRouteCallLegErr()	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	029	--	Correction to values of TpCallNotificationType, TpCallLoadControlMechanismType	4.1.0	4.2.0
Dec 2001	CN_14	NP-010695	030	--	Correction of method getLastRedirectionAddress	4.1.0	4.2.0
Mar 2002	CN_15	NP-020106	031	--	Add P_INVALID_INTERFACE_TYPE exception to IpService.setCallback() and IpService.setCallbackWithSessionID()	4.2.0	4.3.0
Mar 2002	CN_15	NP-020106	032	--	Correction of Event Subscription/Notification Data Type	4.2.0	4.3.0
Mar 2002	CN_15	NP-020106	033	--	Correction of parameter name in IpCallLeg.routeReq() and in IpCallLeg.setAdviceOfCharge()	4.2.0	4.3.0
Mar 2002	CN_15	NP-020106	034	--	Clarification of ambiguous Event handling rules	4.2.0	4.3.0
Jun 2002	CN_16	NP-020180	035	--	Correction to TpCallChargePlan	4.3.0	4.4.0
Jun 2002	CN_16	NP-020180	036	--	Correction to CAMEL Service Property values	4.3.0	4.4.0
Jun 2002	CN_16	NP-020181	037	-	Addition of support for Java API technology realisation	4.4.0	5.0.0
Jun 2002	CN_16	NP-020182	038	-	Addition of support for WSDL realisation	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	039	-	Addition of support for Emergency Telecommunications Service	4.4.0	5.0.0
Jun 2002	CN_16	NP-020183	040	-	Addition of support for Network Controlled Notifications MPCC	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	041	-	Changes to getNotification()	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	042	-	Addition of P_UNSUPPORTED_MEDIA release cause to TpReleaseCause	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	043	-	Addition of CAMEL Phase 4 Service Property values	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	044	-	Addition of indication whether SCS supports initially multiple routeReqs in parallel	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	045	-	Explicit exception for continueProcessing when not in interrupted mode	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	046	-	Indication needed that supervision will be ended when call or callLeg is deassigned	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	047	-	Clarify ambiguous Supervision duration	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	048	-	Detach/Attach request illegal during pending Attach/Detach request	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	049	-	Correction of Multi-Party Call Control properties	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	050	-	Correcting the sequence diagram descriptions in GCC and MPCC	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	051	-	Correcting erroneous description of UI behaviour in call control	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	052	-	Correcting the descriptions of sequence diagrams that don't match the	4.4.0	5.0.0

					diagram		
Jun 2002	CN_16	NP-020187	053	-	Correcting erroneous references to GCC in MPCC	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	054	-	Addition of the Multi-media APIs to Call control SCF (29.198-4)	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	055	-	Updating Clause 4 for Release 5	4.4.0	5.0.0
Jun 2002	CN_16	NP-020188	056	-	Spliting of 29.198-04 into 4 separate TSs (sub-parts)	4.4.0	5.0.0
Sep 2002	CN_17	NP-020429	001	--	29.198-04-1 Add text to clarify requirements on support of methods	5.0.0	5.1.0
Sep 2002	CN_17	NP-020395	002		29.198-04-1 Add text to clarify relationship between 3GPP and ETSI/Parlay OSA specifications	5.0.0	5.1.0
Mar 2003	CN_19	NP-030029	003	--	Correction to Application's requirements for supporting methods	5.1.0	5.2.0
Mar 2003	CN_19	NP-030020	004	--	Correction to remove unused TpCallChargeOrder	5.1.0	5.2.0
Jun 2003	CN_20	NP-030242	005		Correction to Common Call Control Data	5.2.0	5.3.0

CR-Form-v7

CHANGE REQUEST

⌘ 29.198-04-2 CR 008 ⌘rev - ⌘ Current version: 5.3.0 ⌘

*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:	⌘	Correction to Java Realisation Annex
Source:	⌘	CN5 AePONA – Eamonn Murray
Work item code:	⌘	OSA2
Date:	⌘	08/09/2003
Category:	⌘	F
Release:	⌘	REL-5

Use one of the following categories:
- ***F*** *(correction)*
- ***A*** *(corresponds to a correction in an earlier release)*
- ***B*** *(addition of feature),*
- ***C*** *(functional modification of feature)*
- ***D*** *(editorial modification)*

Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

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

Reason for change:	⌘	Correction to Annex C of the current specification. The current Annex references Jain SPA. Jain SPA is no longer a supported activity or deliverable. Replace the current Annex with the Parlay Java Realisation as an informative Annex to the body of OSA API specfication deliverables.
Summary of change:	⌘	Replace the current Annex C that refers to Jain SPA as the informative Java Realisation with the Java Realisation rulebook produced by the Parlay Java Realisation Workgroup.
Consequences if not approved:	⌘	The API specification will reference a realisation that is no longer supported or valid.

Clauses affected:	⌘	Annex C

		Y	N			
Other specs affected:	⌘	X		Other core specifications	⌘	Rel-5 29.198-xy
			X	Test specifications		
			X	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/specs/CR.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.

KEEP the History box of the TS to be changed (see end of the present document)

********** Start of Change # 1 **********

~~Annex C (informative):~~
~~Java API Description of the Call Control SCFs~~

~~The Java API representation of this specification can be obtained from the following URL:~~

~~- Java Call Control (http://jcp.org/jsr/detail/21.jsp)~~

~~Each JSR webpage contains a table identifying the relationships between the different versions of the Parlay, ETSI/OSA, 3GPP/OSA and JAIN SPA specifications. In addition, each JAIN SPA specification version indicates to which Parlay, ETSI/OSA and 3GPP/OSA specification versions it corresponds to.~~

Annex C (informative):
Java API Description of the Call Control SCFs

The Java API realisation of this specification is produced in accordance with the Java Realisation rules defined in Part 1 of this specification. These rules aim to deliver for Java, a developer API, provided as a realisation, supporting a Java API that represents the UML specifications. The rules support the production of both J2SE and J2EE versions of the API from the common UML specifications.

The J2SE representation of this specification is provided as Java Code, contained in archive 2919804-2J2SE.ZIP that accompanies the present document.

The J2EE representation of this specification is provided as Java Code, contained in archive 2919804-2J2EE.ZIP that accompanies the present document.

********** End of Change # 1 **********

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	1.0.0
June 2001	CN_12	NP-010327	--	--	Approved at TSG CN#12 and placed under Change Control	2.0.0	4.0.0
Sep 2001	CN_13	NP-010467	001	--	Changing references to JAIN	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	002	--	Correction of text descriptions for methods enableCallNotification and createNotification	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	003	--	Specify the behaviour when a call leg times out	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	004	--	Removal of Faulty state in MPCCS Call State Transition Diagram and method callFaultDetected in MPCCS in OSA R4	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	005	--	Missing TpCallAppInfoSet description in OSA R4	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	006	--	Redirecting a call leg vs. creating a call leg clarification in OSA R4	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	007	--	Introduction of MPCC Originating and Terminating Call Leg STDs for IpCallLeg	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	008	--	Corrections to SetChargePlan() Addition of PartyToCharge parmeter	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	009	--	Corrections to SetChargePlan()	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	010	--	Remove distinction between final- and intermediate-report	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	011	--	Inclusion of TpMediaType	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	012	--	Corrections to GCC STD	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	013	--	Introduction of sequence diagrams for MPCC services	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	014	--	The use of the REDIRECT event needs to be illustrated	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	015	--	Corrections to SetCallChargePlan()	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	016	--	Add one additional error indication	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	017	--	Corrections to Call Control – GCCS Exception handling	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	018	--	Corrections to Call Control – Errors in Exceptions	4.0.0	4.1.0
Dec 2001	CN_14	NP-010597	019	--	Replace Out Parameters with Return Types	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	020	--	Removal of time based charging property	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	021	--	Make attachMedia() and detachMedia() asynchronous	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	022	--	Correction of treatment datatype in superviseReq on call leg	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	023	--	Corrections to Call Control Data Types	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	024	--	Correction to Call Control (CC)	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	025	--	Amend the Generic Call Control introductory part	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	026	--	Correction in TpCallEventType	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	027	--	Addition of missing description of RouteErr()	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	028	--	Misleading description of createAndRouteCallLegErr()	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	029	--	Correction to values of TpCallNotificationType, TpCallLoadControlMechanismType	4.1.0	4.2.0
Dec 2001	CN_14	NP-010695	030	--	Correction of method getLastRedirectionAddress	4.1.0	4.2.0
Mar 2002	CN_15	NP-020106	031	--	Add P_INVALID_INTERFACE_TYPE exception to IpService.setCallback() and IpService.setCallbackWithSessionID()	4.2.0	4.3.0
Mar 2002	CN_15	NP-020106	032	--	Correction of Event Subscription/Notification Data Type	4.2.0	4.3.0
Mar 2002	CN_15	NP-020106	033	--	Correction of parameter name in IpCallLeg.routeReq() and in IpCallLeg.setAdviceOfCharge()	4.2.0	4.3.0
Mar 2002	CN_15	NP-020106	034	--	Clarification of ambiguous Event handling rules	4.2.0	4.3.0
Jun 2002	CN_16	NP-020180	035	--	Correction to TpCallChargePlan	4.3.0	4.4.0
Jun 2002	CN_16	NP-020180	036	--	Correction to CAMEL Service Property values	4.3.0	4.4.0
Jun 2002	CN_16	NP-020181	037	-	Addition of support for Java API technology realisation	4.4.0	5.0.0
Jun 2002	CN_16	NP-020182	038	-	Addition of support for WSDL realisation	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	039	-	Addition of support for Emergency Telecommunications Service	4.4.0	5.0.0
Jun 2002	CN_16	NP-020183	040	-	Addition of support for Network Controlled Notifications MPCC	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	041	-	Changes to getNotification()	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	042	-	Addition of P_UNSUPPORTED_MEDIA release cause to TpReleaseCause	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	043	-	Addition of CAMEL Phase 4 Service Property values	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	044	-	Addition of indication whether SCS supports initially multiple routeReqs in parallel	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	045	-	Explicit exception for continueProcessing when not in interrupted mode	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	046	-	Indication needed that supervision will be ended when call or callLeg is deassigned	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	047	-	Clarify ambiguous Supervision duration	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	048	-	Detach/Attach request illegal during pending Attach/Detach request	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	049	-	Correction of Multi-Party Call Control properties	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	050	-	Correcting the sequence diagram descriptions in GCC and MPCC	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	051	-	Correcting erroneous description of UI behaviour in call control	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	052	-	Correcting the descriptions of sequence diagrams that don't match the	4.4.0	5.0.0

					diagram		
Jun 2002	CN_16	NP-020187	053	-	Correcting erroneous references to GCC in MPCC	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	054	-	Addition of the Multi-media APIs to Call control SCF (29.198-4)	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	055	-	Updating Clause 4 for Release 5	4.4.0	5.0.0
Jun 2002	CN_16	NP-020188	056	-	Spliting of 29.198-04 into 4 separate TSs (sub-parts)	4.4.0	5.0.0
Sep 2002	CN_17	NP-020430	001	--	29.198-04-2 Correction on use of NULL in Call Control API	5.0.0	5.1.0
Sep 2002	CN_17	NP-020395	002	--	Add text to clarify relationship between 3GPP and ETSI/Parlay OSA specifications	5.0.0	5.1.0
Mar 2003	CN_19	NP-030020	003	--	Correction of status of GCC methods	5.1.0	5.2.0
Mar 2003	CN_19	NP-030020	004	--	Correction to Prepaid Sequence Diagram	5.1.0	5.2.0
Mar 2003	CN_19	NP-030020	005	--	Correction to TpCallEventCriteriaResult in Generic Call Control	5.1.0	5.2.0
Jun 2003	CN_20	NP-030238	007	--	Correction of the description for callEventNotify & reportNotification	5.2.0	5.3.0

CR-Form-v7

CHANGE REQUEST

⌘ 29.198-04-3 CR 014 ⌘rev - ⌘ Current version: 5.3.0 ⌘

*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 ☒

Title:	⌘	Correction to Java Realisation Annex
Source:	⌘	CN5 AePONA – Eamonn Murray
Work item code:	⌘	OSA2
Date:	⌘	08/09/2003
Category:	⌘	**F**
Release:	⌘	*REL-5*

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

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

Reason for change:	⌘	Correction to Annex C of the current specification. The current Annex references Jain SPA. Jain SPA is no longer a supported activity or deliverable. Replace the current Annex with the Parlay Java Realisation as an informative Annex to the body of OSA API specfication deliverables.
Summary of change:	⌘	Replace the current Annex C that refers to Jain SPA as the informative Java Realisation with the Java Realisation rulebook produced by the Parlay Java Realisation Workgroup.
Consequences if not approved:	⌘	The API specification will reference a realisation that is no longer supported or valid.

Clauses affected:	⌘	Annex C

		Y	N			
Other specs affected:	⌘	X		Other core specifications	⌘	Rel-5 29.198-xy
			X	Test specifications		
			X	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/specs/CR.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.

KEEP the History box of the TS to be changed (see end of the present document)

********** Start of Change # 1 **********

~~Annex C (informative):~~
~~Java API Description of the Call Control SCFs~~

~~The Java API representation of this specification can be obtained from the following URL:~~

~~- Java Call Control (http://jcp.org/jsr/detail/21.jsp)~~

~~Each JSR webpage contains a table identifying the relationships between the different versions of the Parlay, ETSI/OSA, 3GPP/OSA and JAIN SPA specifications. In addition, each JAIN SPA specification version indicates to which Parlay, ETSI/OSA and 3GPP/OSA specification versions it corresponds to.~~

Annex C (informative):
Java API Description of the Call Control SCFs

The Java API realisation of this specification is produced in accordance with the Java Realisation rules defined in Part 1 of this specification. These rules aim to deliver for Java, a developer API, provided as a realisation, supporting a Java API that represents the UML specifications. The rules support the production of both J2SE and J2EE versions of the API from the common UML specifications.

The J2SE representation of this specification is provided as Java Code, contained in archive 2919804-3J2SE.ZIP that accompanies the present document.

The J2EE representation of this specification is provided as Java Code, contained in archive 2919804-3J2EE.ZIP that accompanies the present document.

********** End of Change # 1 **********

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	1.0.0
June 2001	CN_12	NP-010327	--	--	Approved at TSG CN#12 and placed under Change Control	2.0.0	4.0.0
Sep 2001	CN_13	NP-010467	001	--	Changing references to JAIN	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	002	--	Correction of text descriptions for methods enableCallNotification and createNotification	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	003	--	Specify the behaviour when a call leg times out	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	004	--	Removal of Faulty state in MPCCS Call State Transition Diagram and method callFaultDetected in MPCCS in OSA R4	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	005	--	Missing TpCallAppInfoSet description in OSA R4	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	006	--	Redirecting a call leg vs. creating a call leg clarification in OSA R4	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	007	--	Introduction of MPCC Originating and Terminating Call Leg STDs for IpCallLeg	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	008	--	Corrections to SetChargePlan() Addition of PartyToCharge parmeter	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	009	--	Corrections to SetChargePlan()	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	010	--	Remove distinction between final- and intermediate-report	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	011	--	Inclusion of TpMediaType	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	012	--	Corrections to GCC STD	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	013	--	Introduction of sequence diagrams for MPCC services	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	014	--	The use of the REDIRECT event needs to be illustrated	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	015	--	Corrections to SetCallChargePlan()	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	016	--	Add one additional error indication	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	017	--	Corrections to Call Control – GCCS Exception handling	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	018	--	Corrections to Call Control – Errors in Exceptions	4.0.0	4.1.0
Dec 2001	CN_14	NP-010597	019	--	Replace Out Parameters with Return Types	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	020	--	Removal of time based charging property	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	021	--	Make attachMedia() and detachMedia() asynchronous	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	022	--	Correction of treatment datatype in superviseReq on call leg	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	023	--	Corrections to Call Control Data Types	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	024	--	Correction to Call Control (CC)	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	025	--	Amend the Generic Call Control introductory part	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	026	--	Correction in TpCallEventType	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	027	--	Addition of missing description of RouteErr()	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	028	--	Misleading description of createAndRouteCallLegErr()	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	029	--	Correction to values of TpCallNotificationType, TpCallLoadControlMechanismType	4.1.0	4.2.0
Dec 2001	CN_14	NP-010695	030	--	Correction of method getLastRedirectionAddress	4.1.0	4.2.0
Mar 2002	CN_15	NP-020106	031	--	Add P_INVALID_INTERFACE_TYPE exception to IpService.setCallback() and IpService.setCallbackWithSessionID()	4.2.0	4.3.0
Mar 2002	CN_15	NP-020106	032	--	Correction of Event Subscription/Notification Data Type	4.2.0	4.3.0
Mar 2002	CN_15	NP-020106	033	--	Correction of parameter name in IpCallLeg.routeReq() and in IpCallLeg.setAdviceOfCharge()	4.2.0	4.3.0
Mar 2002	CN_15	NP-020106	034	--	Clarification of ambiguous Event handling rules	4.2.0	4.3.0
Jun 2002	CN_16	NP-020180	035	--	Correction to TpCallChargePlan	4.3.0	4.4.0
Jun 2002	CN_16	NP-020180	036	--	Correction to CAMEL Service Property values	4.3.0	4.4.0
Jun 2002	CN_16	NP-020181	037	--	Addition of support for Java API technology realisation	4.4.0	5.0.0
Jun 2002	CN_16	NP-020182	038	--	Addition of support for WSDL realisation	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	039	--	Addition of support for Emergency Telecommunications Service	4.4.0	5.0.0
Jun 2002	CN_16	NP-020183	040	--	Addition of support for Network Controlled Notifications MPCC	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	041	--	Changes to getNotification()	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	042	--	Addition of P_UNSUPPORTED_MEDIA release cause to TpReleaseCause	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	043	--	Addition of CAMEL Phase 4 Service Property values	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	044	--	Addition of indication whether SCS supports initially multiple routeReqs in parallel	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	045	--	Explicit exception for continueProcessing when not in interrupted mode	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	046	--	Indication needed that supervision will be ended when call or callLeg is deassigned	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	047	--	Clarify ambiguous Supervision duration	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	048	--	Detach/Attach request illegal during pending Attach/Detach request	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	049	--	Correction of Multi-Party Call Control properties	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	050	--	Correcting the sequence diagram descriptions in GCC and MPCC	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	051	--	Correcting erroneous description of UI behaviour in call control	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	052	--	Correcting the descriptions of sequence diagrams that don't match the	4.4.0	5.0.0

					diagram		
Jun 2002	CN_16	NP-020187	053	--	Correcting erroneous references to GCC in MPCC	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	054	--	Addition of the Multi-media APIs to Call control SCF (29.198-4)	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	055	--	Updating Clause 4 for Release 5	4.4.0	5.0.0
Jun 2002	CN_16	NP-020188	056	--	Spliting of 29.198-04 into 4 separate TSs (sub-parts)	4.4.0	5.0.0
Sep 2002	CN_17	NP-020431	001	--	29.198-04-3 Correction of error in Call Forward on Busy sequence diagram	5.0.0	5.1.0
Sep 2002	CN_17	NP-020431	002	--	Correct inconsistencies in IpCallLeg state transition diagrams	5.0.0	5.1.0
Sep 2002	CN_17	NP-020431	003	--	Clarification of the overlapping criteria definition and eventType mapping to IN TDPs	5.0.0	5.1.0
Sep 2002	CN_17	NP-020431	004	--	Add support for Carrier selection	5.0.0	5.1.0
Sep 2002	CN_17	NP-020431	005	--	Correction on use of NULL in Call Control API	5.0.0	5.1.0
Sep 2002	CN_17	NP-020395	006	--	Add text to clarify relationship between 3GPP and ETSI/Parlay OSA specifications	5.0.0	5.1.0
Mar 2003	CN_19	NP-030031	007	--	Correction of status of MPCC methods	5.1.0	5.2.0
Mar 2003	CN_19	NP-030031	008	--	Inconsistent description of use of secondary callback	5.1.0	5.2.0
Mar 2003	CN_19	NP-030020	009	--	Correction to TpReleaseCauseSet in Multi Party Call Control IDL	5.1.0	5.2.0
Mar 2003	CN_19	NP-030130	010	--	Correction of definition of the P_MAX_CALLLEGS_PER_CALL	5.1.0	5.2.0
Jun 2003	CN_20	NP-030238	011	--	Correction of the description for callEventNotify & reportNotification	5.2.0	5.3.0

CR-Form-v7

CHANGE REQUEST

⌘ 29.198-04-4 CR 009 ⌘rev - ⌘ Current version: 5.3.0 ⌘

*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 ☒

Title:	⌘	Correction to Java Realisation Annex
Source:	⌘	CN5 AePONA – Eamonn Murray
Work item code:	⌘	OSA2
Date:	⌘	08/09/2003
Category:	⌘	**F**
Release:	⌘	*REL-5*

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

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

Reason for change:	⌘	Correction to Annex C of the current specification. The current Annex references Jain SPA. Jain SPA is no longer a supported activity or deliverable. Replace the current Annex with the Parlay Java Realisation as an informative Annex to the body of OSA API specfication deliverables.
Summary of change:	⌘	Replace the current Annex C that refers to Jain SPA as the informative Java Realisation with the Java Realisation rulebook produced by the Parlay Java Realisation Workgroup.
Consequences if not approved:	⌘	The API specification will reference a realisation that is no longer supported or valid.

Clauses affected:	⌘	Annex C

		Y	N			
Other specs affected:	⌘	X		Other core specifications	⌘	Rel-5 29.198-xy
			X	Test specifications		
			X	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/specs/CR.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.

KEEP the History box of the TS to be changed (see end of the present document)

********** Start of Change # 1 **********

~~Annex C (informative):~~
~~Java API Description of the Call Control SCFs~~

~~The Java API representation of this specification can be obtained from the following URL:~~

~~- Java Call Control (http://jcp.org/jsr/detail/21.jsp)~~

~~Each JSR webpage contains a table identifying the relationships between the different versions of the Parlay, ETSI/OSA, 3GPP/OSA and JAIN SPA specifications. In addition, each JAIN SPA specification version indicates to which Parlay, ETSI/OSA and 3GPP/OSA specification versions it corresponds to.~~

Annex C (informative):
Java API Description of the Call Control SCFs

The Java API realisation of this specification is produced in accordance with the Java Realisation rules defined in Part 1 of this specification. These rules aim to deliver for Java, a developer API, provided as a realisation, supporting a Java API that represents the UML specifications. The rules support the production of both J2SE and J2EE versions of the API from the common UML specifications.

The J2SE representation of this specification is provided as Java Code, contained in archive 2919804-4J2SE.ZIP that accompanies the present document.

The J2EE representation of this specification is provided as Java Code, contained in archive 2919804-4J2EE.ZIP that accompanies the present document.

********** End of Change # 1 **********

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	1.0.0
June 2001	CN_12	NP-010327	--	--	Approved at TSG CN#12 and placed under Change Control	2.0.0	4.0.0
Sep 2001	CN_13	NP-010467	001	--	Changing references to JAIN	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	002	--	Correction of text descriptions for methods enableCallNotification and createNotification	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	003	--	Specify the behaviour when a call leg times out	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	004	--	Removal of Faulty state in MPCCS Call State Transition Diagram and method callFaultDetected in MPCCS in OSA R4	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	005	--	Missing TpCallAppInfoSet description in OSA R4	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	006	--	Redirecting a call leg vs. creating a call leg clarification in OSA R4	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	007	--	Introduction of MPCC Originating and Terminating Call Leg STDs for IpCallLeg	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	008	--	Corrections to SetChargePlan() Addition of PartyToCharge parmeter	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	009	--	Corrections to SetChargePlan()	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	010	--	Remove distinction between final- and intermediate-report	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	011	--	Inclusion of TpMediaType	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	012	--	Corrections to GCC STD	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	013	--	Introduction of sequence diagrams for MPCC services	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	014	--	The use of the REDIRECT event needs to be illustrated	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	015	--	Corrections to SetCallChargePlan()	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	016	--	Add one additional error indication	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	017	--	Corrections to Call Control – GCCS Exception handling	4.0.0	4.1.0
Sep 2001	CN_13	NP-010467	018	--	Corrections to Call Control – Errors in Exceptions	4.0.0	4.1.0
Dec 2001	CN_14	NP-010597	019	--	Replace Out Parameters with Return Types	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	020	--	Removal of time based charging property	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	021	--	Make attachMedia() and detachMedia() asynchronous	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	022	--	Correction of treatment datatype in superviseReq on call leg	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	023	--	Corrections to Call Control Data Types	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	024	--	Correction to Call Control (CC)	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	025	--	Amend the Generic Call Control introductory part	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	026	--	Correction in TpCallEventType	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	027	--	Addition of missing description of RouteErr()	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	028	--	Misleading description of createAndRouteCallLegErr()	4.1.0	4.2.0
Dec 2001	CN_14	NP-010597	029	--	Correction to values of TpCallNotificationType, TpCallLoadControlMechanismType	4.1.0	4.2.0
Dec 2001	CN_14	NP-010695	030	--	Correction of method getLastRedirectionAddress	4.1.0	4.2.0
Mar 2002	CN_15	NP-020106	031	--	Add P_INVALID_INTERFACE_TYPE exception to IpService.setCallback() and IpService.setCallbackWithSessionID()	4.2.0	4.3.0
Mar 2002	CN_15	NP-020106	032	--	Correction of Event Subscription/Notification Data Type	4.2.0	4.3.0
Mar 2002	CN_15	NP-020106	033	--	Correction of parameter name in IpCallLeg.routeReq() and in IpCallLeg.setAdviceOfCharge()	4.2.0	4.3.0
Mar 2002	CN_15	NP-020106	034	--	Clarification of ambiguous Event handling rules	4.2.0	4.3.0
Jun 2002	CN_16	NP-020180	035	--	Correction to TpCallChargePlan	4.3.0	4.4.0
Jun 2002	CN_16	NP-020180	036	--	Correction to CAMEL Service Property values	4.3.0	4.4.0
Jun 2002	CN_16	NP-020181	037	--	Addition of support for Java API technology realisation	4.4.0	5.0.0
Jun 2002	CN_16	NP-020182	038	--	Addition of support for WSDL realisation	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	039	--	Addition of support for Emergency Telecommunications Service	4.4.0	5.0.0
Jun 2002	CN_16	NP-020183	040	--	Addition of support for Network Controlled Notifications MPCC	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	041	--	Changes to getNotification()	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	042	--	Addition of P_UNSUPPORTED_MEDIA release cause to TpReleaseCause	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	043	--	Addition of CAMEL Phase 4 Service Property values	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	044	--	Addition of indication whether SCS supports initially multiple routeReqs in parallel	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	045	--	Explicit exception for continueProcessing when not in interrupted mode	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	046	--	Indication needed that supervision will be ended when call or callLeg is deassigned	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	047	--	Clarify ambiguous Supervision duration	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	048	--	Detach/Attach request illegal during pending Attach/Detach request	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	049	--	Correction of Multi-Party Call Control properties	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	050	--	Correcting the sequence diagram descriptions in GCC and MPCC	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	051	--	Correcting erroneous description of UI behaviour in call control	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	052	--	Correcting the descriptions of sequence diagrams that don't match the	4.4.0	5.0.0

					diagram		
Jun 2002	CN_16	NP-020187	053	--	Correcting erroneous references to GCC in MPCC	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	054	--	Addition of the Multi-media APIs to Call control SCF (29.198-4)	4.4.0	5.0.0
Jun 2002	CN_16	NP-020187	055	--	Updating Clause 4 for Release 5	4.4.0	5.0.0
Jun 2002	CN_16	NP-020188	056	--	Spliting of 29.198-04 into 4 separate TSs (sub-parts)	4.4.0	5.0.0
Sep 2002	CN_17	NP-020395	001	--	29.198-04-4 Add text to clarify relationship between 3GPP and ETSI/Parlay OSA specifications	5.0.0	5.1.0
Mar 2003	CN_19	NP-030032	002	--	Correction of status of MMCC methods	5.1.0	5.2.0
Mar 2003	CN_19	NP-030032	003	--	Correction of TpMediaStreamDataTypeRequest	5.1.0	5.2.0
Mar 2003	CN_19	NP-030032	004	--	Addition of missing TpMultiMediaCallIdentifierSet to data types	5.1.0	5.2.0
Jun 2003	CN_20	NP-030238	005	--	Correction of the description for callEventNotify & reportNotification	5.2.0	5.3.0
Jun 2003	CN_20	NP-030243	006	--	Correction to TpAudioCapabiltiesType and TpVideoCapabilitiesType to correctly indicate the required capabilities	5.2.0	5.3.0

CR-Form-v7

CHANGE REQUEST

⌘ 29.198-05 CR 039 ⌘rev - ⌘ Current version: 5.3.0 ⌘

*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 ☒

Title:	⌘	Correction to Java Realisation Annex
Source:	⌘	CN5 AePONA – Eamonn Murray
Work item code:	⌘	OSA2
Date:	⌘	08/09/2003
Category:	⌘	**F**
Release:	⌘	*REL-5*

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

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

Reason for change:	⌘	Correction to Annex C of the current specification. The current Annex references Jain SPA. Jain SPA is no longer a supported activity or deliverable. Replace the current Annex with the Parlay Java Realisation as an informative Annex to the body of OSA API specfication deliverables.
Summary of change:	⌘	Replace the current Annex C that refers to Jain SPA as the informative Java Realisation with the Java Realisation rulebook produced by the Parlay Java Realisation Workgroup.
Consequences if not approved:	⌘	The API specification will reference a realisation that is no longer supported or valid.

Clauses affected:	⌘	Annex C

		Y	N			
Other specs affected:	⌘	X		Other core specifications	⌘	Rel-5 29.198-xy
			X	Test specifications		
			X	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/specs/CR.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.

KEEP the History box of the TS to be changed (see end of the present document)

********** Start of Change # 1 **********

~~Annex C (informative):~~
~~Java API Description of User Interaction SCF~~

~~The Java API representation of this specification can be obtained from the following URL:~~

~~JAIN User Interaction (http://jcp.org/jsr/detail/103.jsp)~~

~~Each JSR webpage contains a table identifying the relationships between the different versions of the Parlay, ETSI/OSA, 3GPP/OSA and JAIN SPA specifications. In addition, each JAIN SPA specification version indicates to which Parlay, ETSI/OSA and 3GPP/OSA specification versions it corresponds to.~~

Annex C (informative):
Java API Description of the User Interaction SCF

The Java API realisation of this specification is produced in accordance with the Java Realisation rules defined in Part 1 of this specification. These rules aim to deliver for Java, a developer API, provided as a realisation, supporting a Java API that represents the UML specifications. The rules support the production of both J2SE and J2EE versions of the API from the common UML specifications.

The J2SE representation of this specification is provided as Java Code, contained in archive 2919805J2SE.ZIP that accompanies the present document.

The J2EE representation of this specification is provided as Java Code, contained in archive 2919805J2EE.ZIP that accompanies the present document.

********** End of Change # 1 **********

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	4.1.0
Sep 2001	CN_13	NP-010468	002	--	Changing references to JAIN	4.1.0	4.2.0
Dec 2001	CN_14	NP-010598	003	--	Replace Out Parameters with Return Types	4.2.0	4.3.0
Dec 2001	CN_14	NP-010598	004	--	Correction of description of sendInfoRes()	4.2.0	4.3.0
Dec 2001	CN_14	NP-010598	005	--	Correction to handling of deassign on related object	4.2.0	4.3.0
Dec 2001	CN_14	NP-010598	006	--	Correction to Exceptions Raised in UI	4.2.0	4.3.0
Dec 2001	CN_14	NP-010598	007	--	Correction to values of TpUIInfoType	4.2.0	4.3.0
Mar 2002	CN_15	NP-020107	008	--	Add P_INVALID_INTERFACE_TYPE exception to IpService.setCallback() and IpService.setCallbackWithSessionID()	4.3.0	4.4.0
Jun 2002	CN_16	NP-020181	009	--	Addition of support for Java API technology realisation	4.4.0	5.0.0
Jun 2002	CN_16	NP-020189	010	--	Improve the vague description of P_ID_NOT_FOUND	4.4.0	5.0.0
Jun 2002	CN_16	NP-020182	011	--	Addition of support for WSDL realisation	4.4.0	5.0.0
Jun 2002	CN_16	NP-020189	012	--	Detach call leg before playing announcement or collecting digits	4.4.0	5.0.0
Jun 2002	CN_16	NP-020189	013	--	Delete P_INVALID_CRITERIA from sendInfoAndCollectReq()	4.4.0	5.0.0
Jun 2002	CN_16	NP-020183	014	--	Addition of Support for Network Controlled Notifications UI	4.4.0	5.0.0
Jun 2002	CN_16	NP-020189	015	--	Correcting erroneous description of UI behaviour in call control	4.4.0	5.0.0
Sep 2002	CN_17	NP-020432	018	--	Add text to clarify requirements on support of methods	5.0.0	5.1.0
Sep 2002	CN_17	NP-020432	019	--	Correction on use of NULL in User Interaction API	5.0.0	5.1.0
Sep 2002	CN_17	NP-020432	020	--	Correction to TpUIInfo data type to support binary data for SMS services	5.0.0	5.1.0
Sep 2002	CN_17	NP-020395	021		Add text to clarify relationship between 3GPP and ETSI/Parlay OSA specifications	5.0.0	5.1.0
Mar 2003	CN_19	NP-030021	023	--	Correction to User Interaction Prepaid Sequence Diagrams	5.1.0	5.2.0
Mar 2003	CN_19	NP-030021	025	--	Correction to getNotification to remove P_INVALID_CRITERIA exception	5.1.0	5.2.0
Mar 2003	CN_19	NP-030021	028	--	Addition of status of methods to User Interaction interfaces	5.1.0	5.2.0
Mar 2003	CN_19	NP-030021	031	--	Corrections to User Interaction	5.1.0	5.2.0
Mar 2003	CN_19	NP-030021	033	--	Correction of User Interaction Event Notification to support non text encodings	5.1.0	5.2.0
Mar 2003	CN_19	NP-030033	029	--	Inconsistent description of use of secondary callback	5.1.0	5.2.0
Jun 2003	CN_20	NP-030238	035	--	Correction of the description for callEventNotify & reportNotification	5.2.0	5.3.0
Jun 2003	CN_20	NP-030244	036	--	Clarify IpUI sendInfoReq()	5.2.0	5.3.0
Jun 2003	CN_20	NP-030244	037	--	Update TpUIInfo for consistency with GMS capabilities	5.2.0	5.3.0
Jun 2003	CN_20	NP-030299	038	1	Specifying the origin of a GUI message	5.2.0	5.3.0

CR-Form-v7

CHANGE REQUEST

⌘ **29.198-06** **CR** **023** ⌘ **rev** **-** ⌘ Current version: **5.2.0** ⌘

*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 ☒

Title:	⌘	Correction to Java Realisation Annex
Source:	⌘	CN5 AePONA – Eamonn Murray
Work item code:	⌘	OSA2
Date:	⌘	08/09/2003
Category:	⌘	**F**
Release:	⌘	*REL-5*

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

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

Reason for change:	⌘	Correction to Annex C of the current specification. The current Annex references Jain SPA. Jain SPA is no longer a supported activity or deliverable. Replace the current Annex with the Parlay Java Realisation as an informative Annex to the body of OSA API specfication deliverables.
Summary of change:	⌘	Replace the current Annex C that refers to Jain SPA as the informative Java Realisation with the Java Realisation rulebook produced by the Parlay Java Realisation Workgroup.
Consequences if not approved:	⌘	The API specification will reference a realisation that is no longer supported or valid.

Clauses affected:	⌘	Annex C			
Other specs affected:	⌘	**Y** / **N**	Other core specifications	⌘	Rel-5 29.198-xy
		X /	Other core specifications		
		/ **X**	Test specifications		
		/ **X**	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/specs/CR.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.

KEEP the History box of the TS to be changed (see end of the present document)

********** Start of Change # 1 **********

~~# Annex C (informative):~~
~~# Java API Description of the Mobility SCFs~~

~~The Java API representation of this specification can be obtained from the following URL:~~

~~- JAIN User Location and Status (http://jcp.org/jsr/detail/98.jsp)~~

~~Each JSR webpage contains a table identifying the relationships between the different versions of the Parlay, ETSI/OSA, 3GPP/OSA and JAIN SPA specifications. In addition, each JAIN SPA specification version indicates to which Parlay, ETSI/OSA and 3GPP/OSA specification versions it corresponds to.~~

Annex C (informative):
Java API Description of the Mobility SCFs

The Java API realisation of this specification is produced in accordance with the Java Realisation rules defined in Part 1 of this specification. These rules aim to deliver for Java, a developer API, provided as a realisation, supporting a Java API that represents the UML specifications. The rules support the production of both J2SE and J2EE versions of the API from the common UML specifications.

The J2SE representation of this specification is provided as Java Code, contained in archive 2919806J2SE.ZIP that accompanies the present document.

The J2EE representation of this specification is provided as Java Code, contained in archive 2919806J2EE.ZIP that accompanies the present document.

********** End of Change # 1 **********

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	4.1.0
Jul 2001	--	--	--	--	29198-06-410.zip archive was packaged with the wrong accompanying IDL zip file, which should be mm.idl and not Call Control IDLs (04).	4.1.0	4.1.1
Sep 2001	CN_13	NP-010520	002	--	Changing references to JAIN	4.1.1	4.2.0
Sep 2001	CN_13	NP-010520	003	--	Introduction of missing mobility exceptions	4.1.1	4.2.0
Oct 2001	--	--	--	--	Correction to ToC	4.2.0	4.2.1
Dec 2001	CN_14	NP-010599	004	--	Replace Out Parameters with Return Types	4.2.1	4.3.0
Dec 2001	CN_14	NP-010599	005	--	Methods accepting an interface as a parameter need to be able to raise P_INVALID_INTERFACE_TYPE	4.2.1	4.3.0
Dec 2001	CN_14	NP-010599	006	--	Correction of references to 3GPP specifications	4.2.1	4.3.0
Dec 2001	CN_14	NP-010599	007	--	Correction to callback interface reference in method IpTriggeredUserLocation.triggeredLocationReportingStartReq	4.2.1	4.3.0
Mar 2002	CN_15	NP-020108	008	--	Add P_INVALID_INTERFACE_TYPE exception to IpService.setCallback() and IpService.setCallbackWithSessionID()	4.3.0	4.4.0
Jun 2002	CN_16	NP-020181	009	--	Addition of support for Java API technology realisation	4.4.0	5.0.0
Jun 2002	CN_16	NP-020182	010	--	Addition of support for WSDL realisation	4.4.0	5.0.0
Jun 2002	CN_16	NP-020190	011	--	Delete the repetitive description of P_APPLICATION_NOT_ACTIVATED and other exceptions	4.4.0	5.0.0
Jun 2002	CN_16	NP-020190	012	--	Correcting type of assignmentID parameters from TpSessionID to TpAssignmentID	4.4.0	5.0.0
Jun 2002	CN_16	NP-020190	013	--	Correcting erroneous references to Service Factory	4.4.0	5.0.0
Sep 2002	CN_17	NP-020433	014	--	Remove all parameter error and network error sequence diagrams	5.0.0	5.1.0
Sep 2002	CN_17	NP-020433	015	--	Removal of unnecessary exception from IpUserLocation.LocationReportReq(), IpUserLocation.extendedLocationReportReq(), IpUserLocation.periodicLocationReportingStartReq()	5.0.0	5.1.0
Sep 2002	CN_17	NP-020433	016	--	Remove unusable exceptions from IpUserLocationCamel.periodicLocationReportingStartReq()	5.0.0	5.1.0
Sep 2002	CN_17	NP-020433	017	--	Add text to clarify requirements on support of methods	5.0.0	5.1.0
Sep 2002	CN_17	NP-020395	018	--	Add text to clarify relationship between 3GPP and ETSI/Parlay OSA specifications	5.0.0	5.1.0
Mar 2003	CN_19	NP-030022	020		Addition of status of methods to Mobility interfaces	5.1.0	5.2.0

CR-Form-v7

CHANGE REQUEST

⌘ 29.198-07 CR 014 ⌘rev - ⌘ Current version: 5.3.0 ⌘

*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 ☒

Title:	⌘	Correction to Java Realisation Annex
Source:	⌘	CN5 AePONA – Eamonn Murray
Work item code:	⌘	OSA2
Date:	⌘	08/09/2003
Category:	⌘	F
Release:	⌘	REL-5

Use one of the following categories:
- ***F*** *(correction)*
- ***A*** *(corresponds to a correction in an earlier release)*
- ***B*** *(addition of feature),*
- ***C*** *(functional modification of feature)*
- ***D*** *(editorial modification)*

Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

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

Reason for change:	⌘	Correction to Annex C of the current specification. The current Annex C includes the document change history rather than referring to the Java Realisation of the API consistent with the other parts of the specification set. Introduce the Parlay Java Realisation as an informative Annex to the body of OSA API specfication deliverables.
Summary of change:	⌘	Introduce the informative Java Realisation consistent with the Java Realisation rulebook produced by the Parlay Java Realisation Workgroup as Annex C.
Consequences if not approved:	⌘	The API specification will not reference a realisation consistent with other individual parts of the specification release.

Clauses affected:	⌘	Annex C

		Y	N			
Other specs affected:	⌘	X		Other core specifications	⌘	Rel-5 29.198-xy
			X	Test specifications		
			X	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/specs/CR.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.

KEEP the History box of the TS to be changed (see end of the present document)

********** Start of Change # 1 **********

Annex C (informative):
Java API Description of the Terminal Capabilities SCF

The Java API realisation of this specification is produced in accordance with the Java Realisation rules defined in Part 1 of this specification. These rules aim to deliver for Java, a developer API, provided as a realisation, supporting a Java API that represents the UML specifications. The rules support the production of both J2SE and J2EE versions of the API from the common UML specifications.

The J2SE representation of this specification is provided as Java Code, contained in archive 2919807J2SE.ZIP that accompanies the present document.

The J2EE representation of this specification is provided as Java Code, contained in archive 2919807J2EE.ZIP that accompanies the present document.

********** End of Change # 1 **********

Annex C (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	4.1.0
Sep 2001	CN_13	NP-010470	002	--	Changing references to JAIN	4.1.0	4.2.0
Dec 2001	CN_14	NP-010600	003	--	Replace Out Parameters with Return Types	4.2.0	4.3.0
Mar 2002	CN_15	NP-020109	004	--	Add P_INVALID_INTERFACE_TYPE exception to IpService.setCallback() and IpService.setCallbackWithSessionID()	4.3.0	4.4.0
Mar 2002	CN_15	NP-020113	005	--	Addition of terminal capability change notifications	4.4.0	5.0.0
Jun 2002	CN_16	NP-020182	006	--	Addition of support for WSDL realisation	5.0.0	5.1.0
Sep 2002	CN-17	NP-020434	007	--	Add text to clarify requirements on support of methods	5.1.0	5.2.0
Sep 2002	CN-17	NP-020395	008	--	Add text to clarify relationship between 3GPP and ETSI/Parlay OSA specifications	5.1.0	5.2.0
Mar 2003	CN_19	NP-030023	011	--	Addition of status of methods to Terminal Capabilities interfaces	5.2.0	5.3.0
Mar 2003	CN_19	NP-030023	013	--	Correction to TpTerminalCapabilities in Terminal Capabilities	5.2.0	5.3.0

CR-Form-v7

CHANGE REQUEST

⌘ **29.198-08** **CR** **026** ⌘ **rev** **-** ⌘ Current version: **5.3.0** ⌘

*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 ☒

Title:	⌘	Correction to Java Realisation Annex
Source:	⌘	CN5 AePONA – Eamonn Murray
Work item code:	⌘	OSA2
Date:	⌘	08/09/2003
Category:	⌘	**F**
Release:	⌘	*REL-5*

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

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

Reason for change:	⌘	Correction to Annex C of the current specification. The current Annex C includes the document change history rather than referring to the Java Realisation of the API consistent with the other parts of the specification set. Introduce the Parlay Java Realisation as an informative Annex to the body of OSA API specfication deliverables.
Summary of change:	⌘	Introduce the informative Java Realisation consistent with the Java Realisation rulebook produced by the Parlay Java Realisation Workgroup as Annex C.
Consequences if not approved:	⌘	The API specification will not reference a realisation consistent with other individual parts of the specification release.

Clauses affected:	⌘	Annex C

		Y	N			
Other specs affected:	⌘	X		Other core specifications	⌘	Rel-5 29.198-xy
			X	Test specifications		
			X	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/specs/CR.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.

KEEP the History box of the TS to be changed (see end of the present document)

********** Start of Change # 1 **********

Annex C (informative):
Java API Description of the Data Session Control SCF

The Java API realisation of this specification is produced in accordance with the Java Realisation rules defined in Part 1 of this specification. These rules aim to deliver for Java, a developer API, provided as a realisation, supporting a Java API that represents the UML specifications. The rules support the production of both J2SE and J2EE versions of the API from the common UML specifications.

The J2SE representation of this specification is provided as Java Code, contained in archive 2919808J2SE.ZIP that accompanies the present document.

The J2EE representation of this specification is provided as Java Code, contained in archive 2919808J2EE.ZIP that accompanies the present document.

********** End of Change # 1 **********

Annex C (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	1.0.0
Jun 2001	CN_12	NP-010330	001	--	Corrections to OSA API Rel4	4.0.0	4.1.0
Sep 2001	CN_13	NP-010471	002	--	Changing references to JAIN	4.1.0	4.2.0
Dec 2001	CN_14	NP-010601	003	--	Replace Out Parameters with Return Types	4.2.0	4.3.0
Dec 2001	CN_14	NP-010601	004	--	Corrections and alignment additions to the Data Session Control SCF	4.2.0	4.3.0
Mar 2002	CN_15	NP-020110	005	--	Add P_INVALID_INTERFACE_TYPE exception to IpService.setCallback() and IpService.setCallbackWithSessionID()	4.3.0	4.4.0
Jun 2002	CN_16	NP-020182	006	--	Addition of support for WSDL realisation	4.4.0	5.0.0
Jun 2002	CN_16	NP-020183	007	--	Addition of Support for Network Controlled Notifications DSC	4.4.0	5.0.0
Jun 2002	CN_16	NP-020192	008	--	Adding missing text concerning the activity timer and criteria overlap	4.4.0	5.0.0
Sep 2002	CN_17	NP-020435	011		Remove duplicate exception from IpDataSessionControlManager.createNotification()	5.0.0	5.1.0
Sep 2002	CN_17	NP-020435	012		Remove P_SERVICE_INFORMATION_MISSING and P_SERVICE_FAULT_ENCOUNTERED exceptions from_DataSessionControl methods.	5.0.0	5.1.0
Sep 2002	CN_17	NP-020435	013		Introduce new method getNotifications to correct the result type of IpDataSessionControlManager.getNotification() to permit retreival of all created notifications.	5.0.0	5.1.0
Sep 2002	CN_17	NP-020435	014		Add P_INVALID_INTERFACE_TYPE exception to IpDataSessionControlManager.createNotification(), resulting in new createNotifications() method	5.0.0	5.1.0
Sep 2002	CN_17	NP-020435	015		Add text to clarify requirements on support of methods	5.0.0	5.1.0
Sep 2002	CN_17	NP-020435	016		Correction on use of NULL in Data Session Control API	5.0.0	5.1.0
Sep 2002	CN_17	NP-020395	017		Add text to clarify relationship between 3GPP and ETSI/Parlay OSA specifications	5.0.0	5.1.0
Mar 2003	CN_19	NP-030024	019	--	Addition of status of methods to Data Session Control interfaces	5.1.0	5.2.0
Mar 2003	CN_19	NP-030024	021	--	Corrections to data types in Data Session Control	5.1.0	5.2.0
Mar 2003	CN_19	NP-030034	022	--	Inconsistent description of use of secondary callback	5.1.0	5.2.0
Mar 2003	CN_19	NP-030034	023	--	Promotion of TpDataSessionQosClass data type definition to the Common Data Types	5.1.0	5.2.0
Jun 2003	CN_20	NP-030238	025	--	Correction of the description for callEventNotify & reportNotification	5.2.0	5.3.0

CR-Form-v7

CHANGE REQUEST

⌘ **29.198-11** **CR** **020** ⌘ **rev** **-** ⌘ Current version: **5.2.0** ⌘

*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 ☒

Title:	⌘	Correction to Java Realisation Annex
Source:	⌘	CN5 AePONA – Eamonn Murray
Work item code:	⌘	OSA2
Date:	⌘	08/09/2003
Category:	⌘	**F**
Release:	⌘	*REL-5*

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

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

Reason for change:	⌘	Correction to Annex C of the current specification. The current Annex C includes the document change history rather than referring to the Java Realisation of the API consistent with the other parts of the specification set. Introduce the Parlay Java Realisation as an informative Annex to the body of OSA API specfication deliverables.
Summary of change:	⌘	Introduce the informative Java Realisation consistent with the Java Realisation rulebook produced by the Parlay Java Realisation Workgroup as Annex C.
Consequences if not approved:	⌘	The API specification will not reference a realisation consistent with other individual parts of the specification release.

Clauses affected:	⌘	Annex C

		Y	N			
Other specs affected:	⌘	X		Other core specifications	⌘	Rel-5 29.198-xy
			X	Test specifications		
			X	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/specs/CR.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.

KEEP the History box of the TS to be changed (see end of the present document)

********** Start of Change # 1 **********

Annex C (informative):
Java API Description of the Account Management SCF

The Java API realisation of this specification is produced in accordance with the Java Realisation rules defined in Part 1 of this specification. These rules aim to deliver for Java, a developer API, provided as a realisation, supporting a Java API that represents the UML specifications. The rules support the production of both J2SE and J2EE versions of the API from the common UML specifications.

The J2SE representation of this specification is provided as Java Code, contained in archive 2919811J2SE.ZIP that accompanies the present document.

The J2EE representation of this specification is provided as Java Code, contained in archive 2919811J2EE.ZIP that accompanies the present document.

********** End of Change # 1 **********

Annex C (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	1.0.0
Jun 2001	CN_12	NP-010327	--	--	Approved at TSG CN#12 and placed under Change Control	2.0.0	4.0.0
Sep 2001	CN_13	NP-010472	001	--	Changing references to JAIN	4.0.0	4.1.0
Sep 2001	CN_13	NP-010472	002	--	Missing exceptions for enabling and changing the notifications	4.0.0	4.1.0
Dec 2001	CN_14	NP-010602	003	--	Replace Out Parameters with Return Types	4.1.0	4.2.0
Dec 2001	CN_14	NP-010602	004	--	Replace erroneous use of incorrect data type TpSessionID by TpAssignmentID in Account Management interface	4.1.0	4.2.0
Mar 2002	CN_15	NP-020111	005	--	Add P_INVALID_INTERFACE_TYPE exception to IpService.setCallback() and IpService.setCallbackWithSessionID()	4.2.0	4.3.0
Mar 2002	CN_15	NP-020111	006	--	Correction of parameter name in IpAccountManager.createNotification()	4.2.0	4.3.0
Mar 2002	CN_15	NP-020111	007	--	Correction of result parameter of getNotification, set in stead of single result	4.2.0	4.3.0
Jun 2002	CN_16	NP-020193	008	--	Change to new Service Property P_MAX_ADDRESSES_PER_QUERY for Account Management	4.3.0	5.0.0
Jun 2002	CN_16	NP-020182	009	--	Addition of support for WSDL realisation	4.3.0	5.0.0
Jun 2002	CN_16	NP-020183	010	--	Addition of Support for Network Controlled Notifications AM	4.3.0	5.0.0
Sep 2002	CN_17	NP-020436	011	--	Correction of IpAccountManager STD to permit multiple notifications	5.0.0	5.1.0
Sep 2002	CN_17	NP-020436	012	--	Add text to clarify requirements on support of methods	5.0.0	5.1.0
Sep 2002	CN_17	NP-020436	013	--	Add missing callback interface for notifications in Account Management	5.0.0	5.1.0
Sep 2002	CN_17	NP-020395	014	--	Add text to clarify relationship between 3GPP and ETSI/Parlay OSA specifications	5.0.0	5.1.0
Mar 2003	CN_19	NP-030025	016	--	Correction to TpChargingEventCriteria in Account Management	5.1.0	5.2.0
Mar 2003	CN_19	NP-030025	018	--	Addition of status of methods to Account Management interfaces	5.1.0	5.2.0
Mar 2003	CN_19	NP-030035	019	--	Inconsistent description of use of secondary callback	5.1.0	5.2.0

CR-Form-v7

CHANGE REQUEST

⌘ 29.198-12 CR 022 ⌘rev - ⌘ Current version: 5.2.0 ⌘

*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:	⌘	Correction to Java Realisation Annex
Source:	⌘	CN5 AePONA – Eamonn Murray
Work item code:	⌘	OSA2
Date:	⌘	08/09/2003
Category:	⌘	F
Release:	⌘	*REL-5*

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

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

Reason for change:	⌘	Correction to Annex C of the current specification. The current Annex references Jain SPA. Jain SPA is no longer a supported activity or deliverable. Replace the current Annex with the Parlay Java Realisation as an informative Annex to the body of OSA API specfication deliverables.
Summary of change:	⌘	Replace the current Annex C that refers to Jain SPA as the informative Java Realisation with the Java Realisation rulebook produced by the Parlay Java Realisation Workgroup.
Consequences if not approved:	⌘	The API specification will reference a realisation that is no longer supported or valid.

Clauses affected:	⌘	Annex C			
Other specs affected:	⌘	**Y**	**N**		
		X		Other core specifications	⌘ Rel-5 29.198-xy
			X	Test specifications	
			X	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/specs/CR.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.

KEEP the History box of the TS to be changed (see end of the present document)

********** Start of Change # 1 **********

~~# Annex C (informative):~~
~~# Java API Description of the Charging SCF~~

~~The Java API representation of this specification can be obtained from the following URL:~~

~~- Java Pay (http://jcp.org/jsr/detail/182.jsp)~~

~~Each JSR webpage contains a table identifying the relationships between the different versions of the Parlay, ETSI/OSA, 3GPP/OSA and JAIN SPA specifications. In addition, each JAIN SPA specification version indicates to which Parlay, ETSI/OSA and 3GPP/OSA specification versions it corresponds to.~~

Annex C (informative):
Java API Description of the Charging SCF

The Java API realisation of this specification is produced in accordance with the Java Realisation rules defined in Part 1 of this specification. These rules aim to deliver for Java, a developer API, provided as a realisation, supporting a Java API that represents the UML specifications. The rules support the production of both J2SE and J2EE versions of the API from the common UML specifications.

The J2SE representation of this specification is provided as Java Code, contained in archive 2919812J2SE.ZIP that accompanies the present document.

The J2EE representation of this specification is provided as Java Code, contained in archive 2919812J2EE.ZIP that accompanies the present document.

********** End of Change # 1 **********

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	1.0.0
Jun 2001	CN_12	NP-010328	--	--		2.0.0	4.0.0
Sep 2001	CN_13	NP-010473	001	--	Changing references to JAIN	4.0.0	4.1.0
Sep 2001	CN_13	NP-010473	002	--	Error corrections charging	4.0.0	4.1.0
Sep 2001	CN_13	NP-010473	003	--	Changed semantics of closeReservation parameter	4.0.0	4.1.0
Sep 2001	CN_13	NP-010473	004	--	Missing errors in definition of (credit/debit)(Amoun/Unit)Err	4.0.0	4.1.0
Sep 2001	CN_13	NP-010473	005	--	Clarification of Unit Reservation	4.0.0	4.1.0
Sep 2001	CN_13	NP-010473	006	--	Improving correlation request and response for applications	4.0.0	4.1.0
Sep 2001	CN_13	NP-010473	007	--	Remove the P_CHS_PARAM_RESULT value from the TpChargingParameterID type	4.0.0	4.1.0
Sep 2001	CN_13	NP-010473	008	--	Align the order of parameters for similar methods	4.0.0	4.1.0
Dec 2001	CN_14	NP-010603	009	--	Replace Out Parameters with Return Types	4.1.0	4.2.0
Mar 2002	CN_15	NP-020112	010	--	Add P_INVALID_INTERFACE_TYPE exception to IpService.setCallback() and IpService.setCallbackWithSessionID()	4.2.0	4.3.0
Mar 2002	CN_15	NP-020112	011	--	Correction of parameter name in IpAppChargingSession.extendLifeTimeRes()	4.2.0	4.3.0
Jun 2002	CN_16	NP-020194	012	--	Clarify the use of setCallback with charging	4.3.0	5.0.0
Jun 2002	CN_16	NP-020194	013	--	Adding Service Properties for the Content Based Charging API	4.3.0	5.0.0
Jun 2002	CN_16	NP-020194	014	--	Addition of support for interactive authorization of payments ("User Confirmation")	4.3.0	5.0.0
Jun 2002	CN_16	NP-020194	015	--	Addition of support for Split Charging feature	4.3.0	5.0.0
Jun 2002	CN_16	NP-020181	016	--	Addition of support for Java API technology realisation	4.3.0	5.0.0
Jun 2002	CN_16	NP-020182	017	--	Addition of support for WSDL realisation	4.3.0	5.0.0
Sep 2002	CN_17	NP-020437	018		Add text to clarify requirements on support of methods	5.0.0	5.1.0
Sep 2002	CN_17	NP-020395	019		Add text to clarify relationship between 3GPP and ETSI/Parlay OSA specifications	5.0.0	5.1.0
Mar 2003	CN_19	NP-030026	021	--	Addition of status of methods to Charging interfaces	5.1.0	5.2.0

CR-Form-v7

CHANGE REQUEST

⌘ 29.198-13 CR 004 ⌘rev - ⌘ Current version: 5.1.0 ⌘

*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 ☒

Title:	⌘	Correction to Java Realisation Annex
Source:	⌘	CN5 AePONA – Eamonn Murray
Work item code:	⌘	OSA2
Date:	⌘	08/09/2003
Category:	⌘	**F**
Release:	⌘	*REL-5*

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

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

Reason for change:	⌘	Correction to Annex B of the current specification. The current Annex B includes the document change history rather than referring to the Java Realisation of the API consistent with the other parts of the specification set. Introduce the Parlay Java Realisation as an informative Annex to the body of OSA API specfication deliverables.
Summary of change:	⌘	Introduce the informative Java Realisation consistent with the Java Realisation rulebook produced by the Parlay Java Realisation Workgroup as Annex B.
Consequences if not approved:	⌘	The API specification will not reference a realisation consistent with other individual parts of the specification release.

Clauses affected:	⌘	Annex B

		Y	N			
Other specs affected:	⌘	X		Other core specifications	⌘	Rel-5 29.198-xy
			X	Test specifications		
			X	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/specs/CR.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.

KEEP the History box of the TS to be changed (see end of the present document)

********** Start of Change # 1 **********

Annex B (informative):
Java API Description of the Policy Management SCF

The Java API realisation of this specification is produced in accordance with the Java Realisation rules defined in Part 1 of this specification. These rules aim to deliver for Java, a developer API, provided as a realisation, supporting a Java API that represents the UML specifications. The rules support the production of both J2SE and J2EE versions of the API from the common UML specifications.

The J2SE representation of this specification is provided as Java Code, contained in archive 2919813J2SE.ZIP that accompanies the present document.

The J2EE representation of this specification is provided as Java Code, contained in archive 2919813J2EE.ZIP that accompanies the present document.

********** End of Change # 1 **********

Annex B (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
April 2002	--	--	--	--	Draft v100 submitted to TSG CN email list for Information	--	1.0.0
June 2002	CN_16	NP-020195	--	--	Draft v200 submitted to TSG CN#16 for Approval	2.0.0	5.0.0
Sep 2002	CN_17	NP-020439	001	--	Add text to clarify requirements on support of methods	5.0.0	5.1.0
Sep 2002	CN_17	NP-020395	002	--	Add text to clarify relationship between 3GPP and ETSI/Parlay OSA specifications	5.0.0	5.1.0

CR-Form-v7

CHANGE REQUEST

⌘ 29.198-14 CR 013 ⌘rev - ⌘ Current version: 5.2.0 ⌘

*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:	⌘	Correction to Java Realisation Annex
Source:	⌘	CN5 AePONA – Eamonn Murray
Work item code:	⌘	OSA2
Date:	⌘	08/09/2003
Category:	⌘	F
Release:	⌘	REL-5

Use one of the following categories:
- ***F*** *(correction)*
- ***A*** *(corresponds to a correction in an earlier release)*
- ***B*** *(addition of feature),*
- ***C*** *(functional modification of feature)*
- ***D*** *(editorial modification)*

Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

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

Reason for change:	⌘	Correction to Annex B of the current specification. The current Annex references Jain SPA. Jain SPA is no longer a supported activity or deliverable. Replace the current Annex with the Parlay Java Realisation as an informative Annex to the body of OSA API specfication deliverables.
Summary of change:	⌘	Replace the current Annex B that refers to Jain SPA as the informative Java Realisation with the Java Realisation rulebook produced by the Parlay Java Realisation Workgroup.
Consequences if not approved:	⌘	The API specification will reference a realisation that is no longer supported or valid.

Clauses affected:	⌘	Annex B			
Other specs affected:	⌘	Y: X / N:	Other core specifications	⌘	Rel-5 29.198-xy
		Y: / N: X	Test specifications		
		Y: / N: X	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/specs/CR.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.

KEEP the History box of the TS to be changed (see end of the present document)

********** Start of Change # 1 **********

~~Annex B (informative):~~
~~Java API Description of the Presence and Availability Management SCFs~~

~~The Java API representation of this specification can be obtained from the following URL:~~

~~- JAIN Presence and Availability Management (http://jcp.org/jsr/detail/123.jsp)~~

~~Each JSR webpage contains a table identifying the relationships between the different versions of the Parlay, ETSI/OSA, 3GPP/OSA and JAIN SPA specifications. In addition, each JAIN SPA specification version indicates to which Parlay, ETSI/OSA and 3GPP/OSA specification versions it corresponds to.~~

Annex B (informative):
Java API Description of the Presence and Availability Management SCFs

The Java API realisation of this specification is produced in accordance with the Java Realisation rules defined in Part 1 of this specification. These rules aim to deliver for Java, a developer API, provided as a realisation, supporting a Java API that represents the UML specifications. The rules support the production of both J2SE and J2EE versions of the API from the common UML specifications.

The J2SE representation of this specification is provided as Java Code, contained in archive 2919814J2SE.ZIP that accompanies the present document.

The J2EE representation of this specification is provided as Java Code, contained in archive 2919814J2EE.ZIP that accompanies the present document.

********** End of Change # 1 **********

Annex C (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
April 2002	--	--	--	--	Draft v100 submitted to TSG CN email list for Information		1.0.0
June 2002	CN_16	NP-020196	--	--	Draft v200 submitted to TSG CN#16 for Approval	2.0.0	5.0.0
Sep 2002	CN_17	NP-020440	001	--	Add text to clarify requirements on support of methods	5.0.0	5.1.0
Sep 2002	CN_17	NP-020440	002	--	Remove declaration of unused datatype TpPAMTime	5.0.0	5.1.0
Sep 2002	CN_17	NP-020395	003	--	Add text to clarify relationship between 3GPP and ETSI/Parlay OSA specifications	5.0.0	5.1.0
Jun 2003	CN_20	NP-030245	004	--	Make TpPAMCapability extensible by changing its type to TpString	5.1.0	5.2.0
Jun 2003	CN_20	NP-030240	005	--	Change the type of TpPAMFQName to TpURN	5.1.0	5.2.0
Jun 2003	CN_20	NP-030245	006	--	Clarifiy use of askerData parameter to getAuthToken method in each PAM SCF	5.1.0	5.2.0
Jun 2003	CN_20	NP-030245	007	--	Add authToken parameter to computeAvailability method	5.1.0	5.2.0
Jun 2003	CN_20	NP-030245	008	--	Replace use of IpInterfaceRef in PAM with actual application interfaces	5.1.0	5.2.0
Jun 2003	CN_20	NP-030245	009	--	Add expiration time for PAM event registrations	5.1.0	5.2.0
Jun 2003	CN_20	NP-030245	010	--	Send subscription notification cancellation to watchers	5.1.0	5.2.0
Jun 2003	CN_20	NP-030241	011	--	Change PAM Presence and Availability SCF name to PAM Access	5.1.0	5.2.0
		NP-030245	012	--	Move Access Control Mechanism to Manager Interface		