ETSI ES 203 915-2 V1.3.1 (2008-04)

ETSI Standard

Open Service Access (OSA); Application Programming Interface (API); Part 2: Common Data Definitions (Parlay 5)

Reference

RES/TISPAN-01055-02-OSA

Keywords

API, IDL, OSA, UML

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This ETSI Standard (ES) has been produced by ETSI Technical Committee Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN).

The present document is part 2 of a multi-part deliverable covering Open Service Access (OSA); Application Programming Interface (API), as identified below. The API specification (ES 203 915) is structured in the following parts:

Part 1: "Overview";

Part 2: "Common Data Definitions";

Part 3: "Framework";

Part 4: "Call Control";

Part 5: "User Interaction SCF";

Part 6: "Mobility SCF";

Part 7: "Terminal Capabilities SCF";

Part 8: "Data Session Control SCF";

Part 9: "Generic Messaging SCF";

Part 10: "Connectivity Manager SCF";

Part 11: "Account Management SCF";

Part 12: "Charging SCF";

Part 13: "Policy Management SCF";

Part 14: "Presence and Availability Management SCF";

Part 15: "Multi-Media Messaging SCF".

The present document has been defined jointly between ETSI, The Parlay Group (http://www.parlay.org) and the 3GPP, in co-operation with a number of JAIN™ Community (http://www.java.sun.com/products/jain) member companies.

The present document forms part of the Parlay 5.2 set of specifications.

The present document is equivalent to 3GPP TS 29.198-2 V6.6.0 (Release 6).

1 Scope

The present document is part 2 of the Stage 3 specification for an Application Programming Interface (API) for Open Service Access (OSA).

The OSA specifications define an architecture that enables application developers to make use of network functionality through an open standardised interface, i.e. the OSA APIs.

The present document specifies the Common Data Definitions of the OSA. The Common Data Definitions contain data-types that are common across the rest of the OSA API. All aspects of the Common Data are defined here, these being:

- Data Definitions.
- IDL Description of the data types.
- WSDL Description of the data types.
- Reference to the Java™ API description of the data types.

2 References

The references listed in clause 2 of ES 203 915-1 contain provisions which, through reference in this text, constitute provisions of the present document.

ETSI ES 203 915-1: "Open Service Access (OSA); Application Programming Interface (API); Part 1: Overview (Parlay 5)".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ES 203 915-1 apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in ES 203 915-1 apply.

4 Common data definitions

The following clauses describe each aspect of the Common data definitions.

The order is as follows:

- The Data Definitions clause shows a detailed expansion of each of the data types associated with the methods within the classes.
- IDL description of the data types (normative annex).
- WSDL description of the data types (informative annex).
- Reference to the Java™ API description of the data types (informative annex).

5 Common system data definitions

These data definitions are assumed to be provided by the client operating system.

5.1 Standard data types

The APIs assume that the following data types can be supported.

5.1.1 TpBoolean

Defines a Boolean data type.

5.1.2 TpInt32

Defines a signed 32-bit integer.

5.1.3 TpFloat

Defines a single precision real number.

5.1.4 TpLongString

Defines a Byte string, comprising length and data. The length must be at least a 32-bit integer.

5.1.5 TpOctet

Defines an 8-bit quantity that is not translated during transmission.

5.1.6 TpOctetSet

Defines a Numbered List of Data elements of TpOctet. Note that this is an ordered list.

5.1.7 TpString

Defines a Byte string, comprising length and data. The length must be at least a 16-bit integer.

5.1.8 TpAssignmentID

Defines an assignment ID with a value that is unique to an instance of an implementation of a given interface (i.e. an object), irrespective of the method invoked on it. This ID may be used, for example, to identify single or multiple event notifications enabled by an object; or by a requesting object to modify or stop functionality (e.g. event notifications, call load control) associated with a previously supplied assignment ID.

The assignment ID is identical to a TpInt32 type.

5.1.9 TpSessionID

Defines a session ID with a value that is at least unique within the context of a specific instance of an SCF. An instance of an SCF is a single service manager instance plus the associated subordinate instances. For example, a single MultiPartyCallControlManager instance plus all associated MultiPartyCall and MultiPartyCallLeg instances. The session ID is used to identify different sessions (e.g. different call or call leg sessions) of an interface capable of handling multiple sessions.

EXAMPLE 1: myCallObject may implement the IpCall interface. If so, myCallObject may handle multiple call sessions, and each call session will be identified by a call session ID value (e.g. 1, 2, 3) that is unique within the context of the SCF instance.

EXAMPLE 2: myCallAndCallLegObject may implement the IpCall and IpCallLeg interfaces. If so, myCallAndCallLegObject may handle multiple call sessions and multiple call leg sessions. Each call session will be identified by a call session ID value (e.g. 1, 2, 3) that is unique within the context of the SCF instance. Similarly, each call leg session will be identified by a call leg session ID value (e.g. 1, 2, 3, 4, 5, 6) that is also unique within the context of the SCF instance. Because call session IDs and call leg session IDs are different data types, overlapping values are permitted and their uniqueness still remains.

The session ID is identical to a TpInt32 type.

5.1.10 TpSessionIDSet

Defines a Numbered Set of Data Elements of TpSessionID.

5.1.11 TpAny

Defines a type that can hold any type. This is not restricted to only the primitive types.

5.1.12 TpAttribute

This is a `Sequence of Data Elements` containing the attribute name and value.

Sequence Element Name	Sequence Element Type	Notes
AttributeName	TpString	The name of the attribute.
AttributeValue	TpAttributeValue	The typed value(s) for the attribute.

5.1.13 TpAttributeValue

This is a tagged choice of data elements to hold attribute values of different complexity.

	Tag Element Type	
	TpAttributeTagInfo	

Tag Element Value	Choice Element Type	Choice Element Name
P_SIMPLE_TYPE	TpSimpleAttributeValue	SimpleValue
P_STRUCTURED_TYPE	TpStructuredAttributeValue	StructuredValue
P_XML_TYPE	TpXMLString	XMLValue

5.1.14 TpAttributeList

This is a `Numbered List of Data Elements` of type TpAttribute.

5.1.15 TpAttributeSet

This is a `Numbered Set of Data Elements` of type TpAttribute.

5.1.16 TpInt64

Defines a signed 64-bit integer.

5.1.17 TpVersion

This data type is identical to `TpString`. It is used to uniquely identify the implemented version of the framework or an SCF. The syntax for this datatype is defined as:

`P_<publishing body>_<version number>`

Where:

`<publishing body>` is one of the strings listed in the table below.

Character String Value	Description
PARLAY	Specification released by The Parlay Group.
ETSI	Specification released by ETSI.
3GPP	Specification released by 3GPP.

`<version number>` consists of numbers separated by underscores (e.g. 3_1). It is recommended that not more than the two most significant numbers (major and minor version) of the version are used.

Examples of version strings are:

Character String Value	Description
P_PARLAY_3_1	Parlay v3.1.
P_ETSI_2_0	ETSI v2.0.
P_3GPP_4_3	3GPP Release 4.3.

Note that different version strings can be aliases of each other all pointing to the same SCF/Framework version.

5.1.18 TpStringSet

Defines a `Numbered Set of Data Elements` of type `TpString`.

5.1.19 TpStringList

Defines a `Numbered List of Data Elements` of type `TpString`.

5.1.20 TpAttributeTagInfo

TpAttributeTagInfo is an enumerated type used as a discriminator for the TpAttributeValue structure, and can contain the following values.

Name	Value	Description
P_SIMPLE_TYPE	0	Simple type
P_STRUCTURED_TYPE	1	Structured type
P_XML_TYPE	2	XML type

5.1.21 TpSimpleAttributeValue

This is a tagged choice of data elements to hold attribute values of different complexity.

	Tag Element Type	
	TpSimpleAttributeTypeInfo	

Tag Element Value	Choice Element Type	Choice Element Name
P_BOOLEAN	TpBoolean	BooleanValue
P_OCTET	TpOctet	OctetValue
P_CHAR	TpChar	CharValue
P_WCHAR	TpWChar	WCharValue
P_STRING	TpString	StringValue
P_WSTRING	TpWString	WStringValue
P_INT16	TpInt16	Int16Value
P_UNSIGNED_INT16	TpUnsignedInt16	UnsignedInt16Value
P_INT32	TpInt32	Int32Value
P_UNSIGNED_INT32	TpUnsignedInt32	UnsignedInt32Value
P_INT64	TpInt64	Int64Value
P_UNSIGNED_INT64	TpUnsignedInt64	UnsignedInt64Value
P_FLOAT	TpFloat	FloatValue
P_DOUBLE	TpDouble	DoubleValue

5.1.22 TpSimpleAttributeTypeInfo

TpSimpleAttributeTypeInfo is an enumerated type used as a discriminator for the TpSimpleAttributeValue structure, and can contain the following values.

Name	Value	Description
P_BOOLEAN	0	Attribute type is type TpBoolean.
P_OCTET	1	Attribute type is type TpOctet.
P_CHAR	2	Attribute type is type TpChar.
P_WCHAR	3	Attribute type is type TpWChar.
P_STRING	4	Attribute type is type TpString.
P_WSTRING	5	Attribute type is type TpWString.
P_INT16	6	Attribute type is type TpInt16.
P_UNSIGNED_INT16	7	Attribute type is type TpUnsignedInt16.
P_INT32	8	Attribute type is type TpInt32.
P_UNSIGNED_INT32	9	Attribute type is type TpUnsignedInt32.
P_INT64	10	Attribute type is type TpInt64.
P_UNSIGNED_INT64	11	Attribute type is type TpUnsignedInt64.
P_FLOAT	12	Attribute type is type TpFloat.
P_DOUBLE	13	Attribute type is type TpDouble.

5.1.23 TpStructuredAttributeType

This data type is identical to a TpString, and is defined as a string of characters that uniquely identifies the type of a structured data type. Network operator specific capabilities may also be used, but should be preceded by the string "SP_". The pattern of values is defined, where module names and class names map to a fully specified class name.

Character String Value	Description
P_module1/module2/module3/className	An object of the specified, fully qualified class.

5.1.24 TpStructuredAttributeValue

This is a `Sequence of Data Elements` containing the structured attribute type tag and the value to be interpreted using that type.

Sequence Element Name	Sequence Element Type	Notes
Type	TpStructuredAttributeType	The type for the value.
Value	TpAny	The structured values for the attribute.

5.1.25 TpChar

This type is an 8-bit quantity that may undergo conversion when transmitted by the communication system.

5.1.26 TpWChar

This type is a quantity that may undergo conversion when transmitted by the communication system. The size of this type is implementation-dependent.

5.1.27 TpWString

Defines a TpWChar string, comprising length and data. The length shall be at least a 16-bit integer.

5.1.28 TpInt16

Defines a signed 16-bit integer.

5.1.29 TpUnsignedInt16

Defines an unsigned 16-bit integer.

5.1.30 TpUnsignedInt32

Defines an unsigned 32-bit integer.

5.1.31 TpUnsignedInt64

Defines an unsigned 64-bit integer.

5.1.32 TpDouble

Defines a double precision real number.

5.1.33 TpXMLString

This data type is TpString containing well-formed XML and may contain a reference to/include a DTD or Schema.

5.1.34 TpUnorderedOctetSet

Defines a Numbered Set of Data elements of TpOctet. Note that this is an un-ordered set.

Note that this type should not be removed from the present document, even if unused by any part of the OSA specifications. It is included to ensure that TpOctetSet is correctly used as a Numbered List of Data Elements, and not a Numbered Set.