SIST

SLOVENSKI STANDARD
SIST ES 202 915-2 V1.2.1:2005

01-januar-2005

Odprti dostop do storitve (OSA) ï Vmesnik za aplikacijsko programiranje (API) ï 2. del: Doloļ itev skupnih podatkov

Open Service Access (OSA); Application Programming Interface (API); Part 2: Common Data Definitions (Parlay 4)

Ta slovenski standard je istoveten z: ES 202 915-2 Version 1.2.1

ICS:

33.040.01	Telekomunikacijski sistemi na splošno	Telecommunication systems in general

SIST ES 202 915-2 V1.2.1:2005 en

ETSI ES 202 915-2 V1.2.1 (2003-08)

ETSI Standard

Open Service Access (OSA);
Application Programming Interface (API);
Part 2: Common Data Definitions
(Parlay 4)

Reference

RES/SPAN-120096-2

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, send your comment to:
editor@etsi.org

Copyright Notification

Contents

Intellectual Property Rights 5

Foreword....5

1 Scope 6

2 References 6

3 Definitions and abbreviations....6
3.1 Definitions....6
3.2 Abbreviations 6

4 Common Data Definitions....6

5 Common System Data Definitions....7
5.1 Standard Data Types 7
5.1.1 TpBoolean....7
5.1.2 TpInt32 7
5.1.3 TpFloat....7
5.1.4 TpLongString....7
5.1.5 TpOctet 7
5.1.6 TpOctetSet 7
5.1.7 TpString 7
5.1.8 TpAssignmentID....7
5.1.9 TpSessionID 7
5.1.10 TpSessionIDSet 8
5.1.11 TpAny....8
5.1.12 TpAttribute 8
5.1.13 TpAttributeType 8
5.1.14 TpAttributeList 8
5.1.15 TpAttributeSet 8
5.1.16 TpInt64 8
5.1.17 TpVersion 9
5.1.18 TpStringSet....9
5.1.19 TpStringList....9
5.2 Other Data Sorts 9
5.2.1 Sequence of Data Elements 9
5.2.2 Tagged Choice of Data Elements 10
5.2.3 Numbered Set of Data Elements....10
5.2.4 Reference 10
5.2.5 Numbered List of Data Elements....10
5.3 Interface Related Data Definitions 10
5.3.1 IpInterface....10
5.3.2 IpInterfaceRef 10
5.4 Exception Classes....11
5.4.1 Underlying Technology Exceptions 11
5.4.2 TpCommonExceptions 11
5.4.3 Constants associated with TpCommonExceptions 11
5.4.4 Exceptions available to all methods on all interfaces 12
5.5 Date and Time Related Data Definitions....13
5.5.1 TpDate 13
5.5.2 TpTime 13
5.5.3 TpDateAndTime 14
5.5.4 TpDuration....14
5.5.5 TpTimeInterval 14
5.6 Address Related Data Definitions 15
5.6.1 TpAddress....15
5.6.2 TpAddressSet....16
5.6.3 TpAddressPresentation 16

5.6.4 TpAddressScreening 16
5.6.5 TpAddressPlan....16
5.6.6 TpAddressError 17
5.6.7 TpAddressRange....17
5.6.8 TpURL....18
5.7 Price-related Data Definitions 18
5.7.1 TpPrice....18
5.7.2 TpAoCInfo....18
5.7.3 TpAoCOrder 18
5.7.4 TpCallAoCOrderCategory....19
5.7.5 TpChargeAdviceInfo 19
5.7.6 TpCAIElements 19
5.7.7 TpChargePerTime....19
5.7.8 TpLanguage 19
5.8 Data Types Common Across Call Control and Data Session Control....20
5.8.1 TpDataSessionQosClass 20

Annex A (normative): OMG IDL Description of the Common Data definitions 21

Annex B (informative): W3C WSDL Description of the Common Data definitions....22

Annex C (informative): Java API Description of the Common Data definitions 23

Annex D (normative): Exception Hierarchy....24

Annex E (informative): Record of changes 32

E.1 Data Definitions 32
E.1.1 New 32
E.1.2 Modified....32
E.1.3 Removed....32

E.2 Exceptions 33
E.2.1 New 33
E.2.2 Modified....33
E.2.3 Removed....33

E.3 Others 33

History 34

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 Services and Protocols for Advanced Networks (SPAN).

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 202 915) 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".

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 4.1 set of specifications.

The present document is equivalent to 3GPP TS 29.198-2 V5.2.0 (Release 5).

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 202 915-1 contain provisions which, through reference in this text, constitute provisions of the present document.

ETSI ES 202 915-1: "Open Service Access (OSA); Application Programming Interface (API); Part 1: Overview (Parlay 4)".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ES 202 915-1 apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations defined in ES 202 915-1 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 Set of Data elements of TpOctet.

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, type, and value. The attribute Value is interpreted based on the value of the attribute Type.

Sequence Element Name	Sequence Element Type	Notes
AttributeName	TpString	The name of the attribute.
AttributeType	TpAttributeType	The type of the attribute. Valid values for Type must include at least TpString, TpInt32 and TpFloat.
AttributeValue	TpAny	The values for the attribute. This model allows multi-valued attributes. Cannot be an empty list.

5.1.13 TpAttributeType

This data type is identical to a TpString, and is defined as a string of characters that uniquely identifies the type of an attribute. Other Network operator specific capabilities may also be used, but should be preceded by the string "SP_". The following values are defined.

Character String Value	Description
NULL	An empty (NULL) string indicates no attribute type.
P_STRING	Attribute type is type TpString.
P_INT32	Attribute type is type TpInt32.
P_FLOAT	Attribute type is type TpFloat.

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`.

5.2 Other Data Sorts

The APIs assumes that the following data syntaxes can be supported.

5.2.1 Sequence of Data Elements

This describes a sequence of data types. This may be defined as a structure (for example, in C++) or simply a sequence of data elements within a structure.

EXAMPLE: The TpAddress data type may be defined in C++ as:

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

5.2.2 Tagged Choice of Data Elements

This describes a data type which actually evaluates to one of a choice of a number of data elements. This data element contains two parts: a tag data type (the *tag* part) which is used to identify the chosen data type, and the chosen data type itself (the *union* part). This form of data type is also referred to as a tagged union.

This data type can be implemented (for example, in C++) as a structure containing an integer for the *tag* part, and a union for the *union* part.

This data type is implementation specific. Please refer to the appropriate IDL documents (and the resulting language mappings) to see how this data type is implemented.

EXAMPLE: The `TpCallError` data type may be defined in C++ as:

```
typedef struct {
  TpCallErrorType Tag;
  union {
    TpCallErrorInfoUndefined     Undefined;
    TpCallErrorInfoRoutingAborted  RoutingAborted;
    TpCallErrorInfoCallAbandoned   CallAbandoned;
    TpCallErrorInfoInvalidAddress  InvalidAddress;
    TpCallErrorInfoInvalidState   InvalidState;
    TpCallErrorInfoInvalidCriteria  InvalidCriteria;
  } callErrorInfo;
} TpCallError;
```

5.2.3 Numbered Set of Data Elements

This describes a data type which comprises an integer which indicates the total number of data elements in the set (the *number* part), and an **unordered** set of data elements (the *data* part). *Set* data types do not contain duplicate data elements.

EXAMPLE: The TpAddressSet data type may be defined in C++ as:

```
typedef struct {
  TpInt32 Number;
  TpAddress Set[Number];
} TpAddressSet;
```

5.2.4 Reference

This describes a reference (or pointer) to a data type.

5.2.5 Numbered List of Data Elements

This describes a data type which comprises an integer which indicates the total number of data elements in the set (the *number* part), and an **ordered** set of data elements (the *data* part). *List* data types can contain duplicate data elements.

EXAMPLE: The TpStringList data type may be defined in C++ as:

```
typedef struct {
  TpInt32 Number;
  TpString List[Number];
} TpStringList;
```

5.3 Interface Related Data Definitions

5.3.1 IpInterface

Defines the address of a generic interface instance.

5.3.2 IpInterfaceRef

Defines a Reference to type IpInterface.