

**Open Service Access;
Application Programming Interface;
Part 2: Common Data Definitions**



Reference

DES/SPAN-120070-2

Keywords

API, OSA, IDL, 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.fr

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2001.
© The Parlay Group 2001.
All rights reserved.

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.....	6
5.1 Standard Data Types	6
5.1.1 TpBoolean.....	7
5.1.2 TpInt32	7
5.1.3 TpInt32Ref.....	7
5.1.4 TpFloat.....	7
5.1.5 TpFloatRef.....	7
5.1.6 TpLongstring	7
5.1.7 TpLongstringRef.....	7
5.1.8 TpOctet	7
5.1.9 TpOctetRef	7
5.1.10 TpOctetSet	7
5.1.11 TpString	7
5.1.12 TpStringRef	7
5.1.13 TpAssignmentID.....	7
5.1.14 TpAssignmentIDRef	8
5.1.15 TpSessionID	8
5.1.16 TpSessionIDRef.....	8
5.1.17 TpSessionIDSet	8
5.2 Other Data Sorts	8
5.2.1 Sequence of Data Elements	8
5.2.2 Tagged Choice of Data Elements	8
5.2.3 Numbered Set of Data Elements.....	9
5.2.4 Reference	9
5.3 Interface Related Data Definitions	9
5.3.1 IpInterface.....	9
5.3.2 IpInterfaceRef	9
5.3.3 IpInterfaceRefRef	9
5.4 Exception Classes.....	10
5.4.1 TpCommonExceptions	10
5.4.2 Constants associated with TpCommonExceptions	10
5.4.3 Exceptions available to all methods on all interfaces	10
5.5 Date and Time Related Data Definitions.....	11
5.5.1 TpDate	11
5.5.2 TpTime	12
5.5.3 TpDateAndTime	12
5.5.4 TpDateAndTimeRef	12
5.5.5 TpDuration.....	13
5.5.6 TpTimeInterval	13
5.6 Address Related Data Definitions	13
5.6.1 TpAddress.....	13
5.6.2 TpAddressRef	14
5.6.3 TpAddressSet.....	14
5.6.4 TpAddressSetRef	14
5.6.5 TpAddressPresentation	14

5.6.6	TpAddressScreening	14
5.6.7	TpAddressPlan	15
5.6.8	TpAddressError	15
5.6.9	TpAddressRange	15
5.6.10	TpURL	16
5.7	Price-related Data Definitions	16
5.7.1	TpPrice	16
5.7.2	TpAoCInfo	17
5.7.3	TpAoCOrder	17
5.7.4	TpCallAoCOrderCategory	17
5.7.5	TpChargeAdviceInfo	17
5.7.6	TpCAIElements	17
5.7.7	TpChargePerTime	18
5.7.8	TpLanguage	18
Annex A (normative):	OMG IDL Description of the Common Data Definitions	19
History		20

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), and is now submitted for the ETSI standards Membership Approval Procedure.

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 201 915) is structured in the following parts:

- Part 1: "Overview";
- Part 2: "Common Data Definitions";**
- Part 3: "Framework";
- Part 4: "Call Control SCF";
- 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".

The present document has been defined jointly between ETSI, The Parlay Group [24] and the 3GPP, in co-operation with a number of JAIN™ Community [25] member companies.

The present document forms part of the Parlay 3.0 set of specifications.

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 interfaces

2 References

The references listed in clause 2 of ES 201 915-1 contain provisions which, through reference in this text, constitute provisions of the present document.

ETSI ES 201 915-1: "Open Service Access; Application Programming Interface; Part 1: Overview".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ES 201 915-1 apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in ES 201 915-1 apply.

4 Common Data Definitions

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

The order is as follows:

- The Data Definitions section shows a detailed expansion of each of the data types associated with the methods within the classes. Note that some data types are used in other methods and classes and are therefore defined within the Common Data types part of this specification.

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 TpInt32Ref

Defines a Reference to a TpInt32.

5.1.4 TpFloat

Defines a single precision real number.

5.1.5 TpFloatRef

Defines a Reference to a TpFloat.

5.1.6 TpLongstring

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

5.1.7 TpLongstringRef

Defines a Reference to a TpLongstring.

5.1.8 TpOctet

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

5.1.9 TpOctetRef

Defines a Reference to a TpOctet.

5.1.10 TpOctetSet

Defines a Numbered Set of Data elements of TpOctet.

5.1.11 TpString

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

5.1.12 TpStringRef

Defines a Reference to a TpString.

5.1.13 TpAssignmentID

This data type is identical to a TpInt32. It specifies a number which identifies an individual event notification enabled by the application or service.

5.1.14 TpAssignmentIDRef

Defines a Reference to type TpAssignmentID.

5.1.15 TpSessionID

Defines a network unique session ID. The API uses this ID to identify sessions, e.g. call or call leg sessions, within an object implementing an interface capable of handling multiple sessions. For the different services, the sessionIDs are unique only in the context of a service manager instantiation (e.g. within the context of one generic call control manager). As such if an application creates two instances of the same service manager it shall use different instantiations of the callback objects which implement the callback interfaces.

The session ID is identical to a TpInt32 type.

5.1.16 TpSessionIDRef

Defines a Reference to a TpSessionID.

5.1.17 TpSessionIDSet

Defines a Numbered_Set_of_Data_Elements of TpSessionID.

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 MIDL as:

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

5.2.4 Reference

This describes a reference (or pointer) to a data type. This is primarily used to describe 'out' method parameters.

This data type may be implemented (for example, in C++) as a pointer. However, in some languages it may not be necessary for 'out' parameters to be implemented as pointers.

Example

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

```
typedef TpAddress *    TpAddressRef
```

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`.

5.3.3 IpInterfaceRefRef

Defines a Reference to type `IpInterfaceRef`.

5.4 Exception Classes

5.4.1 TpCommonExceptions

Defines the structure of the exception class which is applicable to all methods.

Structure Element Name	Structure Element Type	Structure Element Description
ExceptionType	TpInt32	Carries a constant from the list in the table below.
ExtraInformation	TpString	Carries extra information to help identify the source of the exception, e.g. a parameter name.

5.4.2 Constants associated with TpCommonExceptions

Name	Value	Description
P_RESOURCES_UNAVAILABLE	000Dh	The required resources in the network are not available.
P_TASK_REFUSED	000Eh	The requested method has been refused.
P_TASK_CANCELLED	000Fh	The requested method has been cancelled.
P_NO_CALLBACK_ADDRESS_SET	0011h	The requested method is refused because no callback address has been set (this may be the result of a timing issue between setting the callback address and invoking the method).
P_METHOD_NOT_SUPPORTED	0016h	The method is not allowed or supported within the context of the current service agreement.
P_INVALID_STATE	0306h	Unexpected sequence of methods, i.e., the sequence does not match the specified state diagrams.

5.4.3 Exceptions available to all methods on all interfaces

The following are the list of exception classes which are available to all interfaces of the API.

Name	Description
P_APPLICATION_NOT_ACTIVATED	An application is unauthorised to access information and request services with regards to users that have deactivated that particular application. In case the request was for information related to multiple user identities the reference to user identities that are causing this exception will be returned in the extra information of the exception.
P_INFORMATION_NOT_AVAILABLE	The requested information is not available. A reason might be that the information is unavailable in the core network or that the application is unauthorised to access the information. In case the request was for information related to multiple user identities, the reference to user identities that are causing this exception will be returned in the extra information of the exception.
P_INVALID_ADDRESS	Invalid address specified.
P_INVALID_AMOUNT	Invalid amount specified.
P_INVALID_ASSIGNMENT_ID	The assignment ID is invalid.
P_INVALID_CRITERIA	Invalid criteria specified.
P_INVALID_CURRENCY	Invalid currency specified.
P_INVALID_EVENT_TYPE	Invalid event type.
P_INVALID_INTERFACE_NAME	Invalid interface name.
P_INVALID_INTERFACE_TYPE	The interface reference supplied by the client is the wrong type.
P_INVALID_NETWORK_STATE	Although the sequence of method calls is allowed by the gateway, the underlying protocol can not support it. E.g. in some protocols some methods are only allowed by the protocol, when the call processing is suspended, e.g. after reporting an event that was monitored in interrupt mode.
P_INVALID_SESSION_ID	Invalid session ID.
P_INVALID_TIME_AND_DATE_FORMAT	Invalid date and time format provided.
P_SET_LENGTH_EXCEEDED	The maximum set size is exceeded in a method parameter value.
P_UNAUTHORISED_PARAMETER_VALUE	A method parameter value violates the Service Level Agreement.
P_UNKNOWN_SUBSCRIBER	The subscriber is not known in the network or the application is unauthorised to access information. In case the request was for information related to multiple user identities, the reference to user identities that are causing this exception will be returned in the extra information of the exception.
P_UNSUPPORTED_ADDRESS_PLAN	An address contains an address plan which is not supported.

5.5 Date and Time Related Data Definitions

5.5.1 TpDate

This data type is identical to a TpString. It specifies the data in accordance with International Standard ISO 8601. This is defined as the string of characters in the following format:

YYYY-MM-DD

where the date is specified as:

YYYY four digits year

MM two digits month

DD two digits day

The date elements are separated by a hyphen character (-).

EXAMPLE 1: The 4 December 1998, is encoded as the string:

1998-12-04

5.5.2 TpTime

This data type is identical to a TpString. It specifies the time in accordance with International Standard ISO 8601. This is defined as the string of characters in the following format:

HH:MM:SS.mmm

or

HH:MM:SS.mmmZ

where the time is specified as:

HH two digits hours (24h notation)

MM two digits minutes

SS two digits seconds

mmm three digits fractions of a second (i.e. milliseconds)

The time elements are separated by a colon character (:). The date and time are separated by a space. Optionally, a capital letter Z may be appended to the time field to indicate Universal Time (UTC). Otherwise, local time is assumed.

EXAMPLE 2: 10:30 and 15 seconds is encoded as the string:

10:30:15.000

for local time, or in UTC it would be:

10:30:15.000Z

5.5.3 TpDateAndTime

This data type is identical to a TpString. It specifies the data and time in accordance with International Standard ISO 8601. This is defined as the string of characters in the following format:

YYYY-MM-DD HH:MM:SS.mmm

or

YYYY-MM-DD HH:MM:SS.mmmZ

where the date is specified as:

YYYY four digits year

MM two digits month

DD two digits day

The date elements are separated by a hyphen character (-).

The time is specified as:

HH two digits hours (24h notation)

MM two digits minutes

SS two digits seconds

mmm three digits fractions of a second (i.e. milliseconds)

The time elements are separated by a colon character (:). The date and time are separated by a space. Optionally, a capital letter Z may be appended to the time field to indicate Universal Time (UTC). Otherwise, local time is assumed.

EXAMPLE 3: The 4 December 1998, at 10:30 and 15 seconds is encoded as the string:

1998-12-04 10:30:15.000

for local time, or in UTC it would be:

1998-12-04 10:30:15.000Z

5.5.4 TpDateAndTimeRef

Defines a Reference to type TpDateAndTime.

5.5.5 TpDuration

This data type is a TpInt32 representing a time interval in milliseconds. A value of "-1" defines infinite duration and a value of "-2" represents a default duration.

5.5.6 TpTimeInterval

Defines the Sequence of Data Elements that specify a time interval.

Sequence Element Name	Sequence Element Type
StartTime	TpDateAndTime
StopTime	TpDateAndTime

5.6 Address Related Data Definitions

5.6.1 TpAddress

Defines the Sequence_of_Data_Elements that specify an address.

Sequence Element Name	Sequence Element Type
Plan	TpAddressPlan
AddrString	TpString
Name	TpString
Presentation	TpAddressPresentation
Screening	TpAddressScreening
SubAddressString	TpString

The AddrString defines the actual address information and the structure of the string depends on the Plan. The following table gives an overview of the format of the AddrString for the different address plans.

Address Plan	AddrString Format Description	Example
P_ADDRESS_PLAN_NOT_PRESENT	Not applicable.	
P_ADDRESS_PLAN_UNDEFINED	Not applicable.	
P_ADDRESS_PLAN_IP	For Ipv4 the dotted quad notation is used. Also for Ipv6 the dotted notation is used. The address can optionally be followed by a port number separated by a colon.	"127.0.0.1:42"
P_ADDRESS_PLAN_MULTICAST	An Ipv4 class D address or Ipv6 equivalent in dotted notation.	"224.0.0.0"
P_ADDRESS_PLAN_UNICAST	A non multicast or broadcast IP address in dotted notation.	"127.0.0.1"
P_ADDRESS_PLAN_E164	An international number without the international access code, including the country code and excluding the leading zero of the area code.	"31161249111"
P_ADDRESS_PLAN_AESA	The ATM End System Address in binary format (40 bytes).	01234567890ABCDEF01234567890AB CDEF01234567
P_ADDRESS_PLAN_URL	A uniform resource locator as defined in IETF RFC 1738.	"http://www.parlay.org"
P_ADDRESS_PLAN_NSAP	The binary representation of the Network Service Access Point.	490001AA000400010420
P_ADDRESS_PLAN_SMTP	An e-mail address as specified in IETF RFC822.	"webmaster@parlay.org"
P_ADDRESS_PLAN_MSMAIL	Identical to P_ADDRESS_PLAN_SMTP.	"john.doe@hitech.com"

Address Plan	AddrString Format Description	Example
P_ADDRESS_PLAN_X400	The X400 address structured as a set of attribute value pairs separated by semicolons.	"C=nl;ADMD=;PRMD=uninet;O=parlay;S=Doe;I=S;G=John'
P_ADDRESS_PLAN_SIP (Note 1)	A valid SIP address string.	sip:user@parlay.org <sip:enquiries@1.2.3.4:5060> Enquiries
P_ADDRESS_PLAN_ANY (Note 2)	Not applicable.	

NOTE 1: It should be noted that two SIP addresses will be regarded as equivalent by a gateway if they correspond to the same user at the same network address. The textual form of the two addresses need not be the same. For example, sip:enquiries@parlay.org will be deemed to match <sip:Enquiries@1.2.3.4:5060>Enquiries (if parlay.org resolves to 1.2.3.4).

NOTE 2: This is only to be used with TpAddressRange.

5.6.2 TpAddressRef

Defines a Reference to type TpAddress.

5.6.3 TpAddressSet

Defines a Numbered_Set_of_Data_Elements of TpAddress.

5.6.4 TpAddressSetRef

Defines a Reference to type TpAddressSet.

5.6.5 TpAddressPresentation

Defines whether an address can be presented to an end user.

Name	Value	Description
P_ADDRESS_PRESENTATION_UNDEFINED	0	Undefined
P_ADDRESS_PRESENTATION_ALLOWED	1	Presentation Allowed
P_ADDRESS_PRESENTATION_RESTRICTED	2	Presentation Restricted
P_ADDRESS_PRESENTATION_ADDRESS_NOT_AVAILABLE	3	Address not available for presentation

5.6.6 TpAddressScreening

Defines whether an address can be presented to an end user.

Name	Value	Description
P_ADDRESS_SCREENING_UNDEFINED	0	Undefined
P_ADDRESS_SCREENING_USER_VERIFIED_PASSED	1	user provided address verified and passed
P_ADDRESS_SCREENING_USER_NOT_VERIFIED	2	user provided address not verified
P_ADDRESS_SCREENING_USER_VERIFIED_FAILED	3	user provided address verified and failed
P_ADDRESS_SCREENING_NETWORK	4	Network provided address (Note that even though the application may provide the address to the gateway, from the end-user point of view it is still regarded as a network provided address)

5.6.7 TpAddressPlan

Defines the address plan (or numbering plan) used. It is also used to indicate whether an address is actually defined in a TpAddress data element.

Name	Value	Description
P_ADDRESS_PLAN_NOT_PRESENT	-1	No Address Present
P_ADDRESS_PLAN_UNDEFINED	0	Undefined
P_ADDRESS_PLAN_IP	1	IP
P_ADDRESS_PLAN_MULTICAST	2	Multicast
P_ADDRESS_PLAN_UNICAST	3	Unicast
P_ADDRESS_PLAN_E164	4	E.164
P_ADDRESS_PLAN_AESA	5	AESA
P_ADDRESS_PLAN_URL	6	URL
P_ADDRESS_PLAN_NSAP	7	NSAP
P_ADDRESS_PLAN_SMTP	8	SMTP
P_ADDRESS_PLAN_MSMAIL (see note)	9	Microsoft Mail
P_ADDRESS_PLAN_X400	10	X.400
P_ADDRESS_PLAN_SIP	11	SIP
P_ADDRESS_PLAN_ANY	12	Any address plan is deemed to match (This is only used for TpAddressRange)

NOTE: This value is not used in the scope of 3GPP.

For the case where the P_ADDRESS_PLAN_NOT_PRESENT and P_ADDRESS_PLAN_ANY are indicated, the rest of the information in the TpAddress is not valid.

5.6.8 TpAddressError

Defines the reasons why an address is invalid.

Name	Value	Description
P_ADDRESS_INVALID_UNDEFINED	0	Undefined error
P_ADDRESS_INVALID_MISSING	1	Mandatory address not present
P_ADDRESS_INVALID_MISSING_ELEMENT	2	Mandatory address element not present
P_ADDRESS_INVALID_OUT_OF_RANGE	3	Address is outside of the valid range
P_ADDRESS_INVALID_INCOMPLETE	4	Address is incomplete
P_ADDRESS_INVALID_CANNOT_DECODE	5	Address cannot be decoded

5.6.9 TpAddressRange

Defines the Sequence of Data Elements that specify a range of addresses.

Sequence Element Name	Sequence Element Type
Plan	TpAddressPlan
AddrString	TpString
Name	TpString
SubAddressString	TpString

The AddrString defines the actual address information and the structure of the string depends on the Plan.

An overview of the AddrString formats can be found at the description of the TpAddress data-type.

The difference with TpAddress is that there are no Presentation and Screening elements, the AddrString can contain wildcards and Plan may contain P_ADDRESS_PLAN_ANY.

If P_ADDRESS_PLAN_ANY is set then the TpAddressRange will be deemed by the gateway to match any TpAddress. If a specific Plan is set (including P_ADDRESS_PLAN_NOT_PRESENT) then the address plan of the range must be identical to the plan contained in an address for the two to match.

Two wildcards are allowed: * which matches zero or more characters and ? which matches exactly one character. For E164 addresses, *wildcards are only allowed at the end of the string whereas ? are allowed at the beginning or end.

Some examples for E164 addresses:

- "123" matches specifies number;
- "123*" matches all numbers starting with 123 (including 123 itself);
- "123???" matches all numbers starting with 123 and at least 5 digits long;
- "123??" matches all numbers starting with 123 and exactly 6 digits long.

The following address ranges are illegal:

- 1?3
- 1*3
- ?123*

Legal occurrences of the '*' and '?' characters in AddrString should be escaped by a '\' character. To specify a '\' character '\\' must be used.

For e-mail style addresses, the wildcards are allowed at the beginning of the AddrString:

- "*"@parlay.org" matches all email addresses in the parlay.org domain.

For SIP addresses, wildcards are allowed between the 'sip:' and the '@' in the AddrString, e.g.

- "sip:*@parlay.org" matches all SIP addresses at parlay.org:5060.

5.6.10 TpURL

This data type is identical to a TpString and contains a URL address. The usage of this type is distinct from TpAddress, which can also hold a URL. The latter contains a user address which can be specified in many ways: IP, e-mail, URL etc. On the other hand, the TpURL type does not hold the address of a user and always represents a URL. This type is used in user interaction and defines the URL of the test or stream to be sent to an end-user. It is therefore inappropriate to use a general address here.

5.7 Price-related Data Definitions

5.7.1 TpPrice

This data type is identical to a TpString. It specifies price information. This is defined as a string of characters (digits) in the following format:

DDDDDD.DD

5.7.2 TpAoCInfo

Defines the Sequence of Data Elements that specify the Advice Of Charge information to be sent to the terminal.

Sequence Element Name	Sequence Element Type	Description
ChargeOrder	TpAoCOrder	Charge order
Currency	TpString	Currency unit according to ISO-4217:1995

5.7.3 TpAoCOrder

Defines the Tagged Choice of Data Elements that specify the charge plan for the call.

Tag Element Type
TpAoCOrderCategory

Tag Element Value	Choice Element Type	Choice Element Name
P_CHARGE_ADVICE_INFO	TpChargeAdviceInfo	ChargeAdviceInfo
P_CHARGE_PER_TIME	TpChargePerTime	ChargePerTime
P_CHARGE_NETWORK	TpString	NetworkCharge

5.7.4 TpCallAoCOrderCategory

Name	Value	Description
P_CHARGE_ADVICE_INFO	0	Set of GSM Charge Advice Information elements according to TS 122 024
P_CHARGE_PER_TIME	1	Charge per time
P_CHARGE_NETWORK	2	Operator specific charge plan specification, e.g. charging table name / charging table entry

5.7.5 TpChargeAdviceInfo

Defines the Sequence of Data Elements that specify the two sets of Advice of Charge parameters. The first set defines the current tariff. The second set may be used in case of a tariff switch in the network.

Sequence Element Name	Sequence Element Type	Description
CurrentCAI	TpCAIElements	Current tariff
NextCAI	TpCAIElements	Next tariff after tariff switch

5.7.6 TpCAIElements

Defines the Sequence of Data Elements that specify the Charging Advice Information elements according to TS 122 024.

Sequence Element Name	Sequence Element Type	Description
UnitsPerInterval	TpInt32	Units per interval
SecondsPerTimeInterval	TpInt32	Seconds per time interval
ScalingFactor	TpInt32	Scaling factor
UnitIncrement	TpInt32	Unit increment
UnitsPerDataInterval	TpInt32	Units per data interval
SegmentsPerDataInterval	TpInt32	Segments per data interval
InitialSecsPerTimeInterval	TpInt32	Initial secs per time interval

5.7.7 TpChargePerTime

Defines the Sequence of Data Elements that specify the time based charging information.

Sequence Element Name	Sequence Element Type	Description
InitialCharge	TpInt32	Initial charge amount (in currency units * 0.0001)
CurrentChargePerMinute	TpInt32	Current tariff (in currency units * 0.0001)
NextChargePerMinute	TpInt32	Next tariff (in currency units * 0.0001) after tariff switch Only used in setAdviceOfCharge()

5.7.8 TpLanguage

This data type is identical to a TpString, and defines the language. In case an indication for the language is not needed an empty string must be used. In other cases valid language strings are defined in ISO 639.

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

The OMG IDL representation of the present document is contained in a text file (osa.idl contained in archive es_20191502v010101m0.ZIP) which accompanies the present document.

History

Document history		
V1.1.1	December 2001	Membership Approval Procedure MV 20020215: 2001-12-18 to 2002-02-15