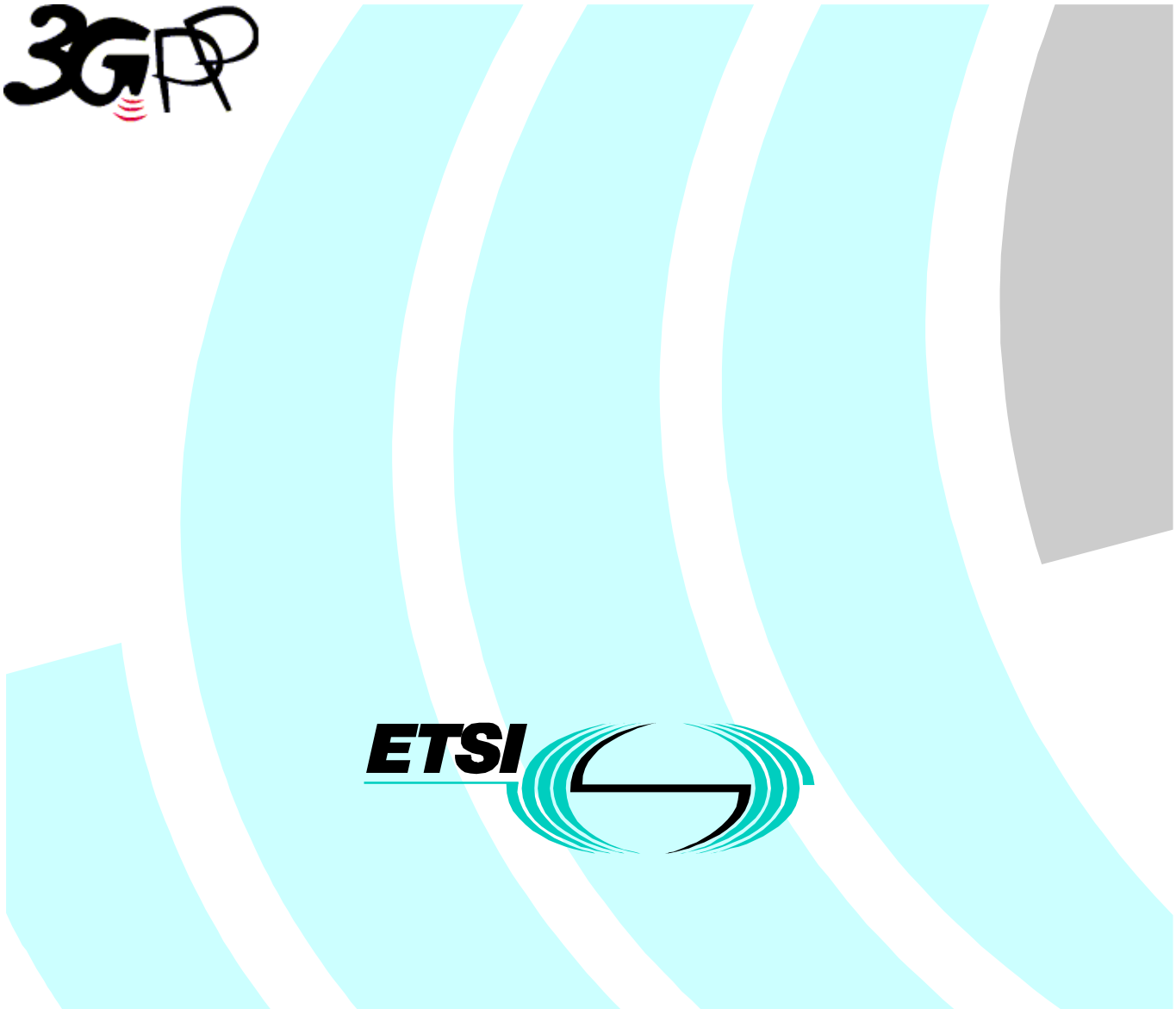
# ETSI TS 129 198-1 V4.1.0 (2001-06)

*Technical Specification*

**Universal Mobile Telecommunications System (UMTS);**
**Open Service Access (OSA);**
**Application Programming Interface (API);**
**Part 1: Overview**
**(3GPP TS 29.198-1 version 4.1.0 Release 4)**



Reference
RTS/TSGN-0529198-1Uv4R1

Keywords
UMTS

***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

# 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://www.etsi.org/ipr).

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 Technical Specification (TS) has been produced by the ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under www.etsi.org/key .

# Contents

# Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

# Introduction

The present document is part 1 of a multi-part TS covering the 3rd Generation Partnership Project: Technical Specification Group Core Network; Open Service Access (OSA); Application Programming Interface (API), as identified below. The **API specification** (3GPP TS 29.198) 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 | (not part of 3GPP Release 4) |
| Part 10: | Connectivity Manager SCF | (not part of 3GPP Release 4) |
| Part 11: | Account Management SCF | |
| Part 12: | Charging SCF | |

The **Mapping specification of the OSA APIs and network protocols** (3GPP TR 29.998) is also structured as above. A mapping to network protocols is however not applicable for all Parts, but the numbering of Parts is kept. Also in case a Part is not supported in a Release, the numbering of the parts is maintained.

| OSA API specifications 29.198-family | | OSA API Mapping - 29.998-family | |
|---|---|---|---|
| **29.198-1** | **Part 1: Overview** | **29.998-1** | **Part 1: Overview** |
| **29.198-2** | **Part 2: Common Data Definitions** | 29.998-2 | Not Applicable |
| **29.198-3** | **Part 3: Framework** | 29.998-3 | Not Applicable |
| **29.198-4** | **Part 4: Call Control SCF** | **29.998-4-1** | **Subpart 1: Generic Call Control – CAP mapping** |
| | | 29.998-4-2 | |
| **29.198-5** | **Part 5: User Interaction SCF** | **29.998-5-1** | **Subpart 1: User Interaction – CAP mapping** |
| | | 29.998-5-2 | |
| | | 29.998-5-3 | |
| | | **29.998-5-4** | **Subpart 4: User Interaction – SMS mapping** |
| **29.198-6** | **Part 6: Mobility SCF** | **29.998-6** | **User Status and User Location – MAP mapping** |
| **29.198-7** | **Part 7: Terminal Capabilities SCF** | 29.998-7 | Not Applicable |
| **29.198-8** | **Part 8: Data Session Control SCF** | **29.998-8** | **Data Session Control – CAP mapping** |
| 29.198-9 | Part 9: Generic Messaging SCF | 29.998-9 | Not Applicable |
| 29.198-10 | Part 10: Connectivity Manager SCF | 29.998-10 | Not Applicable |
| **29.198-11** | **Part 11: Account Management SCF** | 29.998-11 | Not Applicable |
| **29.198-12** | **Part 12: Charging SCF** | 29.998-12 | Not Applicable |

# 1 Scope

The present document is the first part of the 3GPP Specification defining the Application Programming Interface (API) for Open Service Access (OSA), and provides an overview of the content and structure of the various parts of this specification, and of the relation to other standards documents .

The OSA-specifications define an architecture that enables service application developers to make use of network functionality through an open standardised interface, i.e. the OSA APIs. The concepts and the functional architecture for the OSA are contained in 3GPP TS 23.127 [3]. The requirements for OSA are contained in 3GPP TS 22.127 [2].

This specification has been defined jointly between ETSI SPAN12, 3GPP TSG CN WG5 and the Parlay consortium [24], in co-operation with the JAIN consortium [25].

# 2 References

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

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

[1] 3GPP TR 21.905: "3G Vocabulary".

[2] 3GPP TS 22.127: "Stage 1 Service Requirement for the Open Service Access (OSA) (Release 4)".

[3] 3GPP TS 23.127: "Virtual Home Environment (Release 4)".

[4] 3GPP TS 23.078: "CAMEL Phase 3, stage 2".

[5] 3GPP TS 22.101: "Universal Mobile Telecommunications System (UMTS): Service Aspects; Service Principles".

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

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

[8] 3GPP TS 29.078: "CAMEL Phase 3, , CAMEL Application Part (CAP) Specification".

[9] Wireless Application Protocol (WAP), Version 1.2, UAProf Specification (www.wapforum.org).

[10] Wireless Application Protocol (WAP), version 1.2, WAP Service Indication specification, (www.wapforum.org).

[11] Wireless Application Protocol (WAP), version 1.2, WAP Push Architecture Overview (www.wapforum.org).

[12] Wireless Application Protocol (WAP), version 1.2, WAP Architecture (www.wapforum.org).

[13] SUN IDL Compiler (www.javasoft.com/products/jdk/idl/index.html).

[14] UML Unified Modelling Language (www.rational.com/uml).

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

[16] 3GPP TS 22.002: "Circuit Bearer Services supported by a PLMN".

[17] 3GPP TS 22.003: "Circuit Teleservices supported by a PLMN".

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

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

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

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

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

[23] 3GPP TS 22.121: "Service aspects; The Virtual Home Environment (Release 4)".

[24] http://www.parlay.org

[25] http://www.jain.org

[26] 3GPP TS 23.057: "Mobile Station Application Execution Environment (MExE)".

# 3 Definitions and abbreviations

## 3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TS 22.101 [5] and the following apply.

**Applications:** Services, which are designed using Service Capability Features (SCFs).

**Gateway:** Synonym for Service Capability Server (SCS). From the viewpoint of applications, an SCS can be seen as a gateway to the core network.

**HE-VASP:** Home Environment Value Added Service Provider. This is a VASP that has an agreement with the Home Environment to provide services.

**Home Environment:** responsible for overall provision of services to users

**Local Service:** A service, which can be exclusively provided in the current serving network by a Value Added Service Provider.

**OSA Interface:** Standardised Interface used by application to access service capability features.

**Personal Service Environment (PSE):** contains personalised information defining how subscribed services are provided and presented towards the user. The Personal Service Environment is defined in terms of one or more User Profiles.

**Service Capabilities:** Bearers defined by parameters, and/or mechanisms needed to realise services. These are within networks and under network control.

**Service Capability Feature (SCF):** Functionality offered by service capabilities that are accessible via the standardised OSA interface.

**Service Capability Server (SCS)**: Functional Entity providing OSA interfaces towards an application.

**Service**: term used as an alternative for Service Capability Feature in this specification.

**User Interface Profile:** Contains information to present the personalised user interface within the capabilities of the terminal and serving network.

**User Profile:** This is a label identifying a combination of one user interface profile, and one user services profile.

**User Services Profile:** Contains identification of subscriber services, their status and reference to service preferences.

**Value Added Service Provider:** provides services other than basic telecommunications service for which additional charges may be incurred.

**Virtual Home Environment:** A concept for personal service environment portability across network boundaries and between terminals.

## 3.2 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply.

| | |
|---|---|
| API | Application Programming Interface |
| CAMEL | Customised Application for Mobile network Enhanced Logic |
| CAP | CAMEL Application Part |
| CSE | CAMEL Service Environment |
| FW | Framework |
| HE | Home Environment |
| HE-VASP | Home Environment - Value Added Service Provider |
| HLR | Home Location Register |
| INAP | Intelligent Networks Application Part |
| IDL | Interface Description Language |
| MAP | Mobile Application Part |
| ME | Mobile Equipment |
| MExE | Mobile Station (Application) Execution Environment |
| MS | Mobile Station |
| MSC | Mobile Switching Centre |
| OSA | Open Service Access |
| PLMN | Public Land Mobile Network |
| PSE | Personal Service Environment |
| SAT | SIM Application Tool-Kit |
| SCF | Service Capability Feature |
| SCP | Service Control Point |
| SCS | Service Capability Server |
| SIM | Subscriber Identity Module |
| SMS | Short Message Service |
| SMTP | Simple Mail Transfer Protocol |
| UE | User Equipment |
| USIM | Universal Subscriber Identity Module |
| VLR | Visited Location Register |
| VASP | Value Added Service Provider |
| VHE | Virtual Home Environment |
| WAP | Wireless Application Protocol |
| WGP | Wireless Gateway Proxy |
| WPP | Wireless Push Proxy |

# 4 Open Service Access APIs

The OSA-specifications define an architecture that enables service application developers to make use of network functionality through an open standardised interface, i.e. the OSA APIs. The network functionality is describes as Service Capability Features (SCFs) or Services. The OSA Framework is a general component in support of Services (Service Capabilities) and Applications. The concepts and the functional architecture for the OSA are contained in 3GPP TS 23.127 [3]. The requirements for OSA are contained in 3GPP TS 22.127 [2].

The OSA API is split into three types of interface classes, Service and Framework (FW).

1. Interface classes between the Applications and the Framework (FW), that provide applications with basic mechanisms (e.g. Authentication ) that enable them to make use of the service capabilities in the network.

2. Interface classes between Applications and SCFs, which are individual services that may be required by the client to enable the running of third party applications over the interface e.g. Messaging type service.

3. Interface classes between the Framework (FW) and the SCFs, that provide the mechanisms necessary for a multi-vendor environment.

These interfaces represent interfaces 1, 2 and 3 in Figure 1 below. The other interfaces are not yet part of the scope of the work.

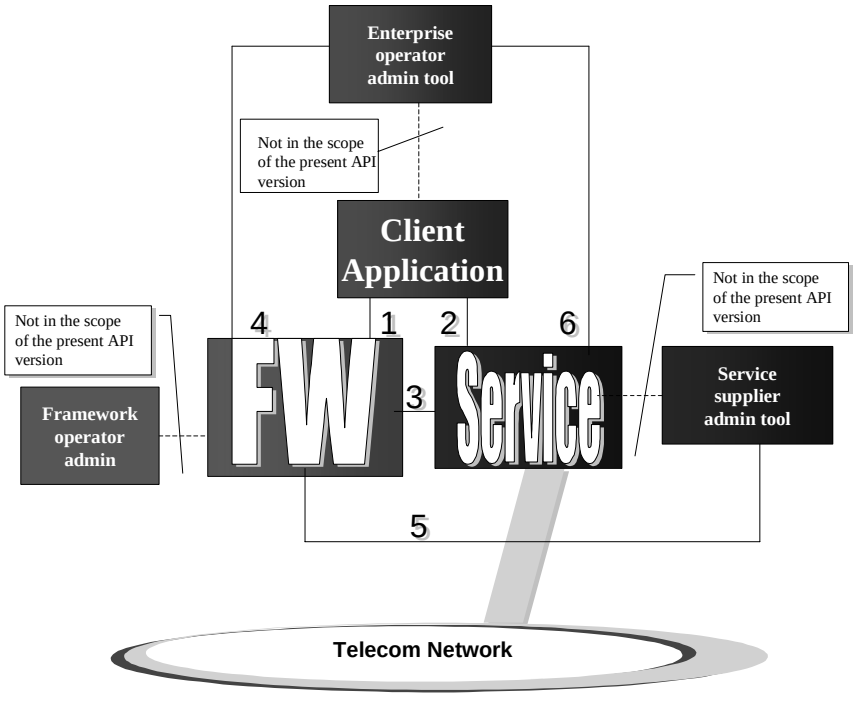



**Figure 1:**

Within the OSA concept a set of Service Capability Features (SCFs) has been specified. The OSA documentation is structured in parts. The first Part (the present document) contains an overview, the second Part contains common data definitions, the third Part the Framework interfaces and the following Parts contain the description of the SCFs.

NOTE: The terms 'Service' and 'Service Capability Feature' are used as alternatives for the same concept in the present document. In the OSA API itself the SCFs as identified in the 3GPP requirements and architecture are reflected as 'service', in terms like serviceFactory, serviceDiscovery.

# 5 Structure of the OSA API (29.198) and Mapping (29.998) documents

The Open Service Access (OSA) Application Programming Interface (API) specifications consist of two sets of documents:

**API specification** (3GPP TS 29.198)
The Parts of 29.198 - apart from Part 1 (the present document) and Part 2 - define the interfaces, parameters and state models that belong to the API specification. UML (Unified Modelling Language) is used to specify the interface classes.
As such it provides a UML interface class description of the methods (API calls) supported by that interface and the relevant parameters and types. The interfaces are specified in IDL (Interface Description Language).

**Mapping specification of the OSA APIs and network protocols** (3GPP TR 29.998)
The Parts of 29.998 contain a possible mapping from the APIs defined in 29.198 to various network protocols (i.e. MAP [7], CAP [8], etc.). It is an informative document, since this mapping is considered as implementation- / vendor-dependent. On the other hand this mapping will provide potential service designers with a better understanding of the relationship of the OSA API interface classes and the behaviour of the network associated to these interface classes.

The purpose of the OSA API is to shield the complexity of the network, its protocols and specific implementation from the applications. This means that applications do not have to be aware of the network nodes, a Service Capability Server interacts with, in order to provide the SCFs to the application. The specific underlying network and its protocols are transparent to the application.

The **API specification** (3GPP TS 29.198) is structured in the following Parts:

| | | |
|---|---|---|
| 29.198-1 | Part 1: | Overview |
| 29.198-2 | Part 2: | Common Data Definitions |
| 29.198-3 | Part 3: | Framework |
| 29.198-4 | Part 4: | Call Control SCF |
| 29.198-5 | Part 5: | User Interaction SCF |
| 29.198-6 | Part 6: | Mobility SCF |
| 29.198-7 | Part 7: | Terminal Capabilities SCF |
| 29.198-8 | Part 8: | Data Session Control SCF |
| 29.198-9 | Part 9: | Generic Messaging SCF |
| 29.198-10 | Part 10: | Connectivity Manager SCF |
| 29.198-11 | Part 11: | Account Management SCF |
| 29.198-12 | Part 12: | Charging SCF |

The **Mapping specification of the OSA APIs and network protocols** (3GPP TR 29.998) is also structured as above. A mapping to network protocols is however not applicable for all Parts, but the numbering of Parts is kept.
Also in case a Part is not supported in a Release, the numbering of the parts is maintained.

**Structure of the Parts of 29.198**

The Parts with API specification themselves are structured as follows:

- The Sequence diagrams give the reader a practical idea of how each of the SCF is implemented.
- The Class relationships clause shows how each of the interfaces applicable to the SCF, relate to one another.
- The Interface specification clause describes in detail each of the interfaces shown within the Class diagram part.

- The State Transition Diagrams (STD) show the progression of internal processes either in the application, or Gateway.
- The Data definitions clauses show a detailed expansion of each of the data types associated with the methods within the classes. It is to be noted that some data types are used in other methods and classes and are therefore defined within the Common Data types part of this specification.
- IDL description of the interface (normative Annex).

# 6 Methodology

Following is a description of the methodology used for the establishment of API specification for OSA.

## 6.1 Tools and Languages

The Unified Modelling Language (UML) [14] is used as the means to specify class and state transition diagrams. Additionally, Object Management Group's (OMG) [15] Interface Definition Language (IDL) is used as the means to programmatically define the interfaces. IDL files are either generated manually from class diagrams or by using a UML tool. In the case IDLs are manually written and/or being corrected manually, correctness has been verified using a CORBA2 (orbos/97-02-25) compliant IDL compiler, e.g. [13].

## 6.2 Packaging

A hierarchical packaging scheme is used to avoid polluting the global name space. The root is defined as:

org.csapi

NOTE: the CORBA module hierarchy defined in the IDLs does not necessarily parallels the logical UML package hierarchy.

## 6.3 Colours

For clarity, class diagrams follow a certain colour scheme. Blue for application interface packages and yellow for all the others.

## 6.4 Naming scheme

The following naming scheme is used for both documentation and IDLs.

**packages**

lowercase.

Using the domain-based naming (For example, org.csapi)

**classes, structures and types. Start with T**

TpCapitalizedWithInternalWordsAlsoCapitalized

**Exception class:**

TpClassNameEndsWithException

**Interface. Start with Ip:**

IpThisIsAnInterface

**constants:**

P_UPPER_CASE_WITH_UNDERSCORES_AND_START_WITH_P

**methods:**

firstWordLowerCaseButInternalWordsCapitalized()

**method's parameters**

firstWordLowerCaseButInternalWordsCapitalized

**collections (set, array or list types)**

TpCollectionEndsWithSet

**class/structure members**

FirstWordAndInternalWordsCapitalized

Spaces in-between words are not allowed.

## 6.5 State Transition Diagram text and text symbols

The descriptions of the State Transitions in the State Transition Diagrams follow the convention:

when_this_event_is_received [guard condition is true] /do_this_action ^send_this_message

Furthermore, text underneath a line through the middle of a State indicates an exit or entry event (normally specified which one).

## 6.6 Error results

As OMG IDL supports exception handling with high efficiency, OSA methods communicate errors in the form of CORBA exceptions of type TpGeneralException in the IDLs; the CORBA methods themselves always return void. But in the documentation, errors are communicated using a return parameter of type TpGeneralResult.

## 6.7 References

In the interface specification whenever parameters are to be passed by reference, the "Ref" suffix is appended to their corresponding data type (e.g. IpAnInterfaceRef anInterface), a reference can also be viewed as a logical indirection. Therefore, structured or primitive data type passed as *out* parameters are references. An interface passed as an *in* parameter is also a reference but an interface passed as an *out* parameter is a double indirection (i.e.: RefRef)

| Original Data type | IN parameter declaration | OUT parameter declaration |
|---|---|---|
| TpPrimitive | parm : IN TpPrimitive | parm : OUT TpPrimitiveRef |
| TpStructured | parm : IN TpStructured | parm : OUT TpStructuredRef |
| IpInterface | parm : IN IpInterfaceRef | parm : OUT IpInterfaceRefRef |

In IDL, however, the following rules apply:

- Interfaces are implicitly passed by reference.
- *out* parameters are also implicitly passed by reference.

This leads to:

- Interface as an *in* parameter: Passed by Reference.
- Structure or primitive type as an *in* parameter: Passed by Value.
- Structure or primitive type as an *out* parameter: Passed by Reference.

- Interface as an *out* parameter: As reference passed by reference.

To simplify the documentation without adding ambiguities, parameters (interfaces, structures and primitive data types) are used as is when specified as *in* or *out* parameters in the IDL. This means that there will be no "Ref" added after the data types of parameters in the IDL.

## 6.8 Number of out parameters

In order to support mapping to as many languages as possible, there is only 1 out parameter allowed per operation.

## 6.9 Strings and Collections

For character strings, the *String* data type is used without regard to the maximum length of the string. In IDL, the data type *String* is typedefed (see Note below) from the CORBA primitive *string*. This CORBA primitive is made up of a length and a variable array of byte.

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

For homogeneous collections of instances of a particular data type the following naming scheme is used: <datatype>Set. In OMG IDL, this maps to a sequence of the data type. A CORBA sequence is implicitly made of a length and a variable array of elements of the same type.

**Example 1:** typedef sequence<TpSessionID> TpSessionIDSet;

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

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

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

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

## 6.10 Prefixes

OSA constants and data types are defined in the global name space: *org.csapi*.

## 6.11 Naming space across CORBA modules

The following shows the naming space used in this specification.

```
module org {
   module csapi {
    /* The fully qualified name of the following constant is
    org::csapi::P_THIS_IS_AN_OSA_GLOBAL_CONST */
    const long P_THIS_IS_AN_OSA_GLOBAL_CONST= 1999;
    // Add other OSA global constants and types here
    module fw {
    /* no scoping required to access  P_THIS_IS_AN_OSA_GLOBAL_CONST */
      const long P_FW_CONST= THIS_IS_AN_OSA_GLOBAL_CONST;
    };
    module mm {
    // scoping required to access P_FW_CONST
      const long P_M_CONST= fw::P_FW_CONST;
    };
   };
};
```

# Annex A (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 |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

# History

| Document history | | |
|---|---|---|
| V4.0.0 | March 2001 | Publication |
| V4.1.0 | June 2001 | Publication |
| | | |
| | | |
| | | |