Open Service Access (OSA);
Application Programming Interface (API);
Part 1: Overview
(Parlay 3)
Contents

Intellectual Property Rights.................................................................................................................................4
Foreword.................................................................................................................................................................4
1 Scope .................................................................................................................................................................5
2 References ........................................................................................................................................................5
3 Definitions and abbreviations.............................................................................................................................6
  3.1 Definitions....................................................................................................................................................6
  3.2 Abbreviations ............................................................................................................................................7
4 Open Service Access APTs ..................................................................................................................................8
5 Document structure ............................................................................................................................................9
6 Methodology ......................................................................................................................................................10
  6.1 Tools and Languages.................................................................................................................................10
  6.2 Packaging Structure.................................................................................................................................10
  6.3 Colours ....................................................................................................................................................13
  6.4 Naming scheme .......................................................................................................................................13
  6.5 State Transition Diagram text and text symbols.....................................................................................14
  6.6 Exception handling and passing results.................................................................................................14
  6.7 References ................................................................................................................................................14
  6.8 Strings and Collections............................................................................................................................14
  6.9 Prefixes.....................................................................................................................................................14
7 Relationship between ETSI, Parlay and 3GPP OSA releases..............................................................................14
Annex A (normative): OMG IDL ..........................................................................................................................16
  A.1 Tools and languages.................................................................................................................................16
  A.2 Strings and collections.............................................................................................................................16
  A.3 Naming space across CORBA modules.................................................................................................16
Annex B (informative): Bibliography....................................................................................................................17
History....................................................................................................................................................................18
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 1 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 (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 3.3 set of specifications.

The present document is equivalent to 3GPP TS 29.198-1 V4.3.2 (Release 4).
1 Scope

The present document is part 1 of the Stage 3 specification for an Application Programming Interface (API) for Open Service Access (OSA), and provides an overview of the content and structure of the various parts of the present document, 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 standardized interface, i.e. the OSA APIs.

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 and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

[1] ETSI TR 121 905: "Universal Mobile Telecommunications System (UMTS); Vocabulary for 3GPP Specifications (3GPP TR 21.905)".

[2] ETSI TS 122 024: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Description of Charge Advice Information (CAI) (3GPP TS 22.024)".


[5] ETSI TS 122 101: "Universal Mobile Telecommunications System (UMTS); Service aspects; Service principles (3GPP TS 22.101)".


NOTE: http://www.w3.org/TR/NOTE-CCPP/.

[7] ETSI TS 129 002: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Mobile Application Part (MAP) specification (3GPP TS 29.002)".

[8] ETSI TS 129 078: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Customized Applications for Mobile network Enhanced Logic (CAMEL); CAMEL Application Part (CAP) specification (3GPP TS 29.078)".


NOTE: http://www.wapforum.org/what/technical_1_2.htm.


NOTE: http://www.wapforum.org/what/technical_1_2.htm.


ETSI TS 122 002: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Circuit Bearer Services (BS) supported by a Public Land Mobile Network (PLMN) (3GPP TS 22.002)".

ETSI TS 122 003: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Circuit Teleservices supported by a Public Land Mobile Network (PLMN) (3GPP TS 22.003)".

ETSI TS 124 002: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); GSM-UMTS Public Land Mobile Network (PLMN) access reference configuration (3GPP TS 24.002)".

ITU-T Recommendation Q.763: "Signalling System No. 7 - ISDN User Part formats and codes".

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

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

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

ISO 639: "Code for the representation of names of languages".

IETF RFC 822: "Standard for the format of ARPA Internet text messages".

IETF RFC 1738: "Uniform Resource Locators (URL)".

ETSI TS 129 198-1: "ETSI TS 129 198 (V3.4.0): "Universal Mobile Telecommunications System (UMTS); Open Service Architecture (OSA) Application Programming Interface (API) - Part 1 (3GPP TS 29.198 version 3.4.0 Release 1999)".

ETSI TS 129 198 V4 (all parts): "Universal Mobile Telecommunications System (UMTS); Open Service Access (OSA) Application Programming Interface (API); (3GPP TS 29.198 Release 4)".

ETSI TS 123 107: "Universal Mobile Telecommunications System (UMTS); Quality of Service (QoS) concept and architecture (3GPP TS 23.107)".

ETSI TS 123 271: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Location Services (LCS); Functional description; Stage 2 (3GPP TS 23.271)".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TS 122 101 [5] and the following apply:

applications: services, which are designed using service capability features
gateway: synonym for Service Capability Server

NOTE 1: From the viewpoint of applications, a Service Capability Server can be seen as a gateway to the core network.

NOTE 2: 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: service which can be exclusively provided in the current serving network by a Value Added Service Provider

OSA Interface: standardized Interface used by application to access service capability features

Personal Service Environment (PSE): contains personalized information defining how subscribed services are provided and presented towards the user

NOTE: The Personal Service Environment is defined in terms of one or more User Profiles.

Service: alternative for Service Capability Feature (in the present document)

Service Capabilities (SC): bearers defined by parameters, and/or mechanisms needed to realize services

NOTE: These are within networks and under network control.

Service Capability Feature (SCF): functionality offered by service capabilities that are accessible via the standardized OSA interface

Service Capability Server: Functional Entity providing OSA interfaces towards an application

User Interface Profile: contains information to present the personalized user interface within the capabilities of the terminal and serving network

User Profile: 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: concept for personal service environment portability across network boundaries and between terminals

3.2 Abbreviations

For the purposes of the present document, the abbreviations defined in TR 121 905 [1] and the following apply:

- API: Application Programming Interface
- CAMEL: Customized Application for Mobile Network Enhanced Logic
- CSE: Camel Service Environment
- HE: Home Environment
- HE-VASP: Home Environment Value Added Service Provider
- HLR: Home Location Register
- IDL: Interface Description Language
- INAP: Intelligent Networks Application Part
- MAP: Mobile Application Part
- ME: Mobile Equipment
- ME-xE: 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
- SC: Service Capabilities
The OSA specifications define an architecture that enables service application developers to make use of network functionality through an open standardized interface, i.e. the OSA API's. The network functionality is described as Service Capability Features or Services (see note). The OSA Framework is a general component in support of Services (Service Capabilities) and Applications.

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

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

- Interface classes between Applications and Service Capability Features (SCF), 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.

- Interface classes between the Framework and the Service Capability Features, that provide the mechanisms necessary for multi-vendorship.

These interfaces represent interfaces 1, 2 and 3 of figure 1. The other interfaces are not yet part of the scope of the work.
Within the OSA concept a set of Service Capability Features 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. The rest of the 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 Service Capability Features as identified in the 3GPP requirements and architecture are reflected as 'service', in terms like serviceFactory, serviceDiscovery.

### 5 Document structure

The parts of the present document ES 201 915 (apart from 1 (the present document) and 2) define the interfaces, parameters and state models that form part of the API specification. UML 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.

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 Service Capability Features to the application. The specific underlying network and its protocols are transparent to the application.

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".

A 3GPP mapping document, TR 129 998, is also structured according to the same parts. It contains a possible mapping from some of the APIs defined in ES 202 915 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. A mapping to network protocols is 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 ES 201 915:**

The parts with API specification themselves are structured as follows:

- The Sequence diagrams give the reader a practical idea of how each of the service capability feature is implemented.
- The Class relationships clause show 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 clause show 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 the present document.
- IDL description of the interface used as 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) is used as the means to specify class and state transition diagrams.

### 6.2 Packaging Structure

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

```plaintext
orv.csapi
```
The following diagram shows the packaging hierarchy. The root package is shown on the left most side of the figure. Extending from the root package are the framework and services branch packages, then the associated leaf packages. Listed against each package are the interfaces, data types, exceptions and service properties it contains.

<table>
<thead>
<tr>
<th>Packaging Hierarchy</th>
<th>Contains</th>
</tr>
</thead>
</table>
| org.csapi           | IpInterface  
                      IpService  
                      All common data types  
                      All common exceptions  
                      All common service properties |
| .fw                 | Common Framework data types  
                      Common Framework exceptions  
                      Common Framework service properties |
| .access             | Package interfaces  
                      Package data types  
                      Package exceptions  
                      Package service properties |
| .trust_and_security | Package interfaces  
                      Package data types  
                      Package exceptions  
                      Package service properties |
| .application        | Package interfaces  
                      Package data types  
                      Package exceptions  
                      Package service properties |
| .notification       | Package interfaces  
                      Package data types  
                      Package exceptions  
                      Package service properties |
| .integrity          | Package interfaces  
                      Package data types  
                      Package exceptions  
                      Package service properties |
| .service_agreement  | Package interfaces  
                      Package data types  
                      Package exceptions  
                      Package service properties |
| .discovery          | Package interfaces  
                      Package data types  
                      Package exceptions  
                      Package service properties |
| .enterprise_operator| Package interfaces  
                      Package data types  
                      Package exceptions  
                      Package service properties |
| .service_subscription| Package interfaces  
                         Package data types  
                         Package exceptions  
                         Package service properties |
| service              | Package interfaces  
                         Package data types  
                         Package exceptions  
                         Package service properties |
| .notification       | Package interfaces  
                         Package data types  
                         Package exceptions  
                         Package service properties |
| .integrity          | Package interfaces  
                         Package data types  
                         Package exceptions  
                         Package service properties |
| .discovery          | Package interfaces  
                         Package data types  
                         Package exceptions  
                         Package service properties |
| .service_lifecycle  | Package interfaces  
                         Package data types  
                         Package exceptions  
                         Package service properties |
| .service_registration| Package interfaces  
                        Package data types  
                        Package exceptions  
                        Package service properties |
| .services           | Common Service data types  
                         Common Service exceptions  
                         Common Service service properties |
<table>
<thead>
<tr>
<th>Packaging Hierarchy</th>
<th>Contains</th>
</tr>
</thead>
</table>
| .cc                 | Common Call Control data types  
|                     | Common Call Control exceptions  
|                     | Common Call Control service properties |
| .gccs               | Package interfaces  
|                     | Package data types  
|                     | Package exceptions  
|                     | Package service properties |
| .mpccs              | Package interfaces  
|                     | Package data types  
|                     | Package exceptions  
|                     | Package service properties |
| .mmccs              | Package interfaces  
|                     | Package data types  
|                     | Package exceptions  
|                     | Package service properties |
| .cccs               | Package interfaces  
|                     | Package data types  
|                     | Package exceptions  
|                     | Package service properties |
| .ui                 | Package interfaces  
|                     | Package data types  
|                     | Package exceptions  
|                     | Package service properties |
| .mm                 | Common Mobility management data types  
|                     | Common Mobility management exceptions  
|                     | Common Mobility management service properties |
| .ul                 | Package interfaces  
|                     | Package data types  
|                     | Package exceptions  
|                     | Package service properties |
| .ulc                | Package interfaces  
|                     | Package data types  
|                     | Package exceptions  
|                     | Package service properties |
| .ule                | Package interfaces  
|                     | Package data types  
|                     | Package exceptions  
|                     | Package service properties |
| .us                 | Package interfaces  
|                     | Package data types  
|                     | Package exceptions  
|                     | Package service properties |
| .termcap            | Package interfaces  
|                     | Package data types  
|                     | Package exceptions  
|                     | Package service properties |
| .dsc                | Package interfaces  
|                     | Package data types  
|                     | Package exceptions  
|                     | Package service properties |
| .gms                | Package interfaces  
|                     | Package data types  
|                     | Package exceptions  
|                     | Package service properties |
| .cm                 | Package interfaces  
|                     | Package data types  
|                     | Package exceptions  
|                     | Package service properties |
| .am                 | Package interfaces  
|                     | Package data types  
|                     | Package exceptions  
|                     | Package service properties |
Packaging Hierarchy

<table>
<thead>
<tr>
<th>.cs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains</td>
</tr>
<tr>
<td>Package interfaces</td>
</tr>
<tr>
<td>Package data types</td>
</tr>
<tr>
<td>Package exceptions</td>
</tr>
<tr>
<td>Package service properties</td>
</tr>
</tbody>
</table>

NOTE 1: Not all the packages given above may be found in the 3GPP OSA specifications.

NOTE 2: Where data types, exceptions and service properties are indicated in the figure above their presence, or otherwise, is dependent upon the package in question. For example, if there are no common Framework exceptions then none will be present in the org.csapi.fw package.

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 documentation.

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 Exception handling and passing results

OSA methods communicate errors in the form of exceptions. OSA methods themselves always use the return parameter to pass results. If no results are to be returned a void is used instead of the return parameter. In order to support mapping to as many languages as possible, no method `out` parameters are allowed.

6.7 References

In the interface specification whenever Interface parameters are to be passed as an `in` parameter, they are done so by reference, and the "Ref" suffix is appended to their corresponding type (e.g. IpAnInterfaceRef anInterface), a reference can also be viewed as a logical indirection.

<table>
<thead>
<tr>
<th>Original type</th>
<th>IN parameter declaration</th>
</tr>
</thead>
<tbody>
<tr>
<td>IpInterface</td>
<td>param : IN IpInterfaceRef</td>
</tr>
</tbody>
</table>

6.8 Strings and Collections

For character strings, the `String` data type is used without regard to the maximum length of the string. For homogeneous collections of instances of a particular data type the following naming scheme is used: `<datatype>Set`.

6.9 Prefixes

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

7 Relationship between ETSI, Parlay and 3GPP OSA releases

The following table explains how the various releases of ETSI, Parlay and 3GPP OSA specifications correspond. Each ETSI and 3GPP specification carries a version number and is updated independently. The frequency of 3GPP updates may be up to every 3 months, which is greater than that of ETSI or Parlay, therefore, while there is a corresponding version of 3GPP TS 29.198 for every version of ETSI ES 201 915 or ES 202 915, there is not necessarily a corresponding version of the ETSI specification for each version of the 3GPP specification. For example, there is no ETSI or Parlay specification version which corresponds exactly to the 3GPP issue of TS 29.198 Release 4 from December 2001.
### ETSI ES 201 915 / Parlay 3 / 3GPP TS 29.198 Release 4 (version 4.x.x)

<table>
<thead>
<tr>
<th>ETSI OSA Specification Set</th>
<th>Parlay Phase</th>
<th>3GPP TS 29.198 version</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Release 4, March 2001 Plenary</td>
</tr>
<tr>
<td>ES 201 915 v.1.1.1 (complete release)</td>
<td>Parlay 3.0</td>
<td>Release 4, September 2001 Plenary</td>
</tr>
<tr>
<td>ES 201 915 v.1.2.1 (complete release)</td>
<td>Parlay 3.1</td>
<td>Release 4, March 2002 Plenary</td>
</tr>
<tr>
<td>ES 201 915 v.1.3.1 (complete release)</td>
<td>Parlay 3.2</td>
<td>Release 4, June 2002 Plenary</td>
</tr>
<tr>
<td>ES 201 915 v.1.4.1 (complete release)</td>
<td>Parlay 3.3</td>
<td>Release 4, September 2002 Plenary</td>
</tr>
</tbody>
</table>

### ETSI ES 202 915 / Parlay 4 / 3GPP TS 29.198 Release 5 (version 5.x.x)

<table>
<thead>
<tr>
<th>ETSI OSA Specification Set</th>
<th>Parlay Phase</th>
<th>3GPP TS 29.198 version</th>
</tr>
</thead>
<tbody>
<tr>
<td>ES 202 915 v.1.1.1 (complete release)</td>
<td>Parlay 4.0</td>
<td>Release 5, September 2002 Plenary</td>
</tr>
<tr>
<td>ES 202 915 v.1.2.1 (not parts 9, 13, 14)</td>
<td>Parlay 4.1</td>
<td>Release 5, March 2003 Plenary</td>
</tr>
</tbody>
</table>
Annex A (normative):
OMG IDL

A.1 Tools and languages

The Object Management Group's (OMG) (http://www.omg.org/) Interface Definition Language (IDL) is used as a 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. "SUN IDL Compiler (http://java.sun.com/products/jdk/idl/index.html).

A.2 Strings and collections

In IDL, the data type String is typedefed (see note) 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.

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:

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

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

A.3 Naming space across CORBA modules

The following shows the naming space used in the present document.

```cpp
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= P_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 B (informative):
Bibliography

- ETSI TS 129 998: "Universal Mobile Telecommunications System (UMTS); Open Service Access (OSA) Application Programming Interface (API) Mapping for Open Service Access (3GPP TR 29.998 Release 5)"

- ETSI TS 123 127: "Universal Mobile Telecommunications System (UMTS); Virtual Home Environment/Open Service Architecture (3GPP TS 23.127 Release 4)".

- ETSI TS 122 127: "Universal Mobile Telecommunications System (UMTS); Service Requirement for the Open Services Access (OSA); Stage 1 (3GPP TS 22.127 Release 4)".

- ETSI TS 123 057: "Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); Mobile Execution Environment (MExE); Functional description; Stage 2 (3GPP TS 23.057)"

- ETSI TS 123 078: "Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); Customised Applications for Mobile network Enhanced Logic (CAMEL) Phase 3 - Stage 2 (3GPP TS 23.078)"

- ETSI TS 129 198 (V4): "Universal Mobile Telecommunications System (UMTS); Open Service Access (OSA); Application Programming Interface (API) (3GPP TS 29.198 Version 4 Release 4)"

- "IDL to Java Compiler". (http://java.sun.com/products/jdk/idl/index.html)

- "UML Unified Modelling Language". (http://www.omg.org/uml/)

- "Object Management Group". (http://www.omg.org/)

- "The Parlay Group homepage". (http://www.parlay.org)

- "JAIN Community homepage". (http://www.java.sun.com/products/jain)
## History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1.1.1</td>
<td>February 2002</td>
<td>Publication</td>
</tr>
<tr>
<td>V1.2.1</td>
<td>July 2002</td>
<td>Publication</td>
</tr>
<tr>
<td>V1.3.1</td>
<td>October 2002</td>
<td>Publication</td>
</tr>
<tr>
<td>V1.4.1</td>
<td>July 2003</td>
<td>Publication</td>
</tr>
</tbody>
</table>