

**Open Service Access (OSA);  
Application Programming Interface (API);  
Part 16: Service Broker SCF  
(Parlay 6)**



---

Reference

DES/TISPAN-01032-16-OSA

---

Keywords

API, IDL, OSA, UML

**ETSI**

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

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

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

---

**Important notice**

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

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

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, please send your comment to one of the following services:

[http://portal.etsi.org/chaicor/ETSI\\_support.asp](http://portal.etsi.org/chaicor/ETSI_support.asp)

---

**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 2008.

© The Parlay Group 2008.

All rights reserved.

**DECT**<sup>TM</sup>, **PLUGTESTS**<sup>TM</sup>, **UMTS**<sup>TM</sup>, **TIPHON**<sup>TM</sup>, the TIPHON logo and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.

**3GPP**<sup>TM</sup> is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

# 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 Service Broker SCF.....	7
4.1 General requirements on support of methods.....	7
5 Sequence Diagrams .....	7
6 Class Diagrams.....	8
7 The Service Interface Specifications .....	8
7.1 Interface Specification Format .....	8
7.1.1 Interface Class .....	8
7.1.2 Method descriptions.....	8
7.1.3 Parameter descriptions.....	8
7.1.4 State Model.....	9
7.2 Base Interface.....	9
7.2.1 Interface Class IpInterface .....	9
7.3 Service Interfaces .....	9
7.3.1 Overview .....	9
7.4 Generic Service Interface .....	9
7.4.1 Interface Class IpService .....	9
7.4.1.1 Method setCallback().....	10
7.4.1.2 Method setCallbackWithSessionID().....	10
8 Service Broker Interface Classes.....	10
8.1 Interface Class IpServiceBroker.....	10
8.1.1 Method register() .....	11
8.1.2 Method unregister() .....	12
9 State Transition Diagrams .....	12
10 Service Broker Service Properties.....	12
11 Data Definitions .....	12
11.1 Service Broker Data Definitions .....	12
11.1.1 clientBrokerID .....	12
11.1.2 TpEndpointAddress .....	13
11.1.3 TpEndpointAddressCategory.....	13
11.1.4 TpServiceKey .....	13
11.1.5 TpServiceKeyType .....	13
12 Exception Classes.....	13
<b>Annex A (normative):</b> <b>OMG IDL Description of Service Broker SCF .....</b>	<b>14</b>
<b>Annex B (informative):</b> <b>W3C WSDL Description of Service Broker SCF.....</b>	<b>15</b>
<b>Annex C (informative):</b> <b>Java API Description of the Service Broker SCF .....</b>	<b>16</b>
<b>Annex D (informative):</b> <b>Contents of 3GPP OSA R7 Service Broker .....</b>	<b>17</b>
<b>Annex E (informative):</b> <b>Description of Service Broker for 3GPP2 cdma2000 networks.....</b>	<b>18</b>

E.1	General Exceptions.....	18
E.2	Specific Exceptions .....	18
E.2.1	Clause 1: Scope .....	18
E.2.2	Clause 2: References .....	18
E.2.3	Clause 3: Definitions and abbreviations .....	18
E.2.4	Clause 4: Service Broker SCF .....	18
E.2.5	Clause 5: Sequence Diagrams .....	18
E.2.6	Clause 6: Class Diagrams .....	18
E.2.7	Clause 7: The Service Interface Specifications .....	18
E.2.8	Clause 8: Service Broker Interface Classes .....	19
E.2.9	Clause 9: State Transition Diagrams .....	19
E.2.10	Clause 10: Service Broker Service Properties .....	19
E.2.11	Clause 11: Data Definitions.....	19
E.2.12	Annex A (normative): OMG IDL Description of Service Broker SCF .....	19
E.2.13	Annex B (informative): W3C WSDL Description of Service Broker SCF .....	19
<b>Annex F (informative): Record of changes .....</b>		<b>20</b>
F.1	Interfaces .....	20
F.1.1	New .....	20
F.1.2	Deprecated.....	20
F.1.3	Removed.....	20
F.2	Methods.....	20
F.2.1	New .....	20
F.2.2	Deprecated.....	20
F.2.3	Modified.....	21
F.2.4	Removed.....	21
F.3	Data Definitions .....	21
F.3.1	New .....	21
F.3.2	Modified.....	21
F.3.3	Removed.....	21
F.4	Service Properties.....	21
F.4.1	New .....	21
F.4.2	Deprecated.....	22
F.4.3	Modified.....	22
F.4.4	Removed.....	22
F.5	Exceptions .....	22
F.5.1	New .....	22
F.5.2	Modified.....	22
F.5.3	Removed.....	22
F.6	Others .....	22
	History .....	23

---

## Intellectual Property Rights

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

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

---

## Foreword

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

The present document is part 16 of a multi-part deliverable covering Open Service Access (OSA); Application Programming Interface (API), as identified below. The API specification (ES 204 915) is structured in the following parts:

- Part 1: "Overview";
- Part 2: "Common Data Definitions";
- Part 3: "Framework";
- Part 4: "Call Control";
- Part 5: "User Interaction SCF";
- Part 6: "Mobility SCF";
- Part 7: "Terminal Capabilities SCF";
- Part 8: "Data Session Control SCF";
- Part 9: "Generic Messaging SCF";
- Part 10: "Connectivity Manager SCF";
- Part 11: "Account Management SCF";
- Part 12: "Charging SCF";
- Part 13: "Policy Management SCF";
- Part 14: "Presence and Availability Management SCF";
- Part 15: "Multi-Media Messaging SCF";
- Part 16: "Service Broker 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 6.0 set of specifications.**

**The present document is equivalent to 3GPP TS 29.198-16 V1.0.0 (Release 7).**

---

# 1 Scope

The present document is part 16 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 Service Broker Service Capability Feature (SCF) aspects of the interface. All aspects of the Service Broker SCF are defined here, these being:

- Sequence Diagrams.
- Class Diagrams.
- Interface specification plus detailed method descriptions.
- State Transition diagrams.
- Data Definitions.
- IDL Description of the interfaces.

The process by which this task is accomplished is through the use of object modelling techniques described by the Unified Modelling Language (UML).

---

# 2 References

The references listed in clause 2 of ES 204 915-1 contain provisions which, through reference in this text, constitute provisions of the present document.

ETSI ES 204 915-1: "Open Service Access (OSA); Application Programming Interface (API); Part 1: Overview (Parlay 6)".

---

# 3 Definitions and abbreviations

## 3.1 Definitions

For the purposes of the present document, the terms and definitions given in ES 204 915-1 apply.

## 3.2 Abbreviations

For the purposes of the present document, the abbreviations given in ES 204 915-1 apply.

---

## 4 Service Broker SCF

The following clauses describe each aspect of the Service Broker Service Capability Feature (SCF).

The order is 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 transition between states in the SCF. The states and transitions are well-defined; either methods specified in the Interface specification or events occurring in the underlying networks cause state transitions.
- The Data Definitions clause 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 ES 204 915-2.

### 4.1 General requirements on support of methods

An implementation of this API which supports or implements a method described in the present document, shall support or implement the functionality described for that method, for at least one valid set of values for the parameters of that method.

Where a method is not supported by an implementation of a Service interface, the exception `P_METHOD_NOT_SUPPORTED` shall be returned to any call of that method.

Where a method is not supported by an implementation of an Application interface, a call to that method shall be possible, and no exception shall be returned.

---

## 5 Sequence Diagrams

There are no Sequence Diagrams for the Service Broker SCF.

## 6 Class Diagrams

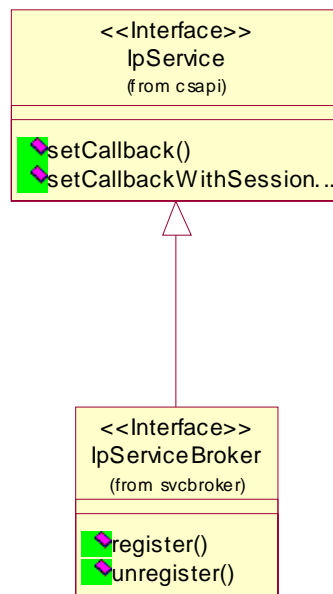


Figure 1: Service Broker Interfaces Overview

## 7 The Service Interface Specifications

### 7.1 Interface Specification Format

This clause defines the interfaces, methods and parameters that form a part of the API specification. The Unified Modelling Language (UML) is used to specify the interface classes. The general format of an interface specification is described below.

#### 7.1.1 Interface Class

This shows a UML interface class description of the methods supported by that interface, and the relevant parameters and types. The Service and Framework interfaces for enterprise-based client applications are denoted by classes with name `Ip<name>`. The callback interfaces to the applications are denoted by classes with name `IpApp<name>`. For the interfaces between a Service and the Framework, the Service interfaces are typically denoted by classes with name `IpSvc<name>`, while the Framework interfaces are denoted by classes with name `IpFw<name>`.

#### 7.1.2 Method descriptions

Each method (API method “call”) is described. Both synchronous and asynchronous methods are used in the API. Asynchronous methods are identified by a 'Req' suffix for a method request, and, if applicable, are served by asynchronous methods identified by either a 'Res' or 'Err' suffix for method results and errors, respectively. To handle responses and reports, the application or service developer must implement the relevant `IpApp<name>` or `IpSvc<name>` interfaces to provide the callback mechanism.

#### 7.1.3 Parameter descriptions

Each method parameter and its possible values are described. Parameters described as 'in' represent those that must have a value when the method is called. Those described as 'out' are those that contain the return result of the method when the method returns.



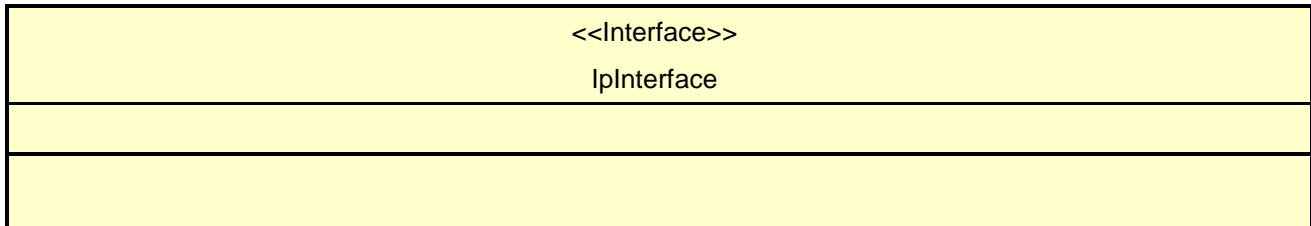
## 7.1.4 State Model

If relevant, a state model is shown to illustrate the states of the objects that implement the described interface.

## 7.2 Base Interface

### 7.2.1 Interface Class IpInterface

All application, framework and service interfaces inherit from the following interface. This API Base Interface does not provide any additional methods.



## 7.3 Service Interfaces

### 7.3.1 Overview

The Service Interfaces provide the interfaces into the capabilities of the underlying network - such as call control, user interaction, messaging, mobility and connectivity management.

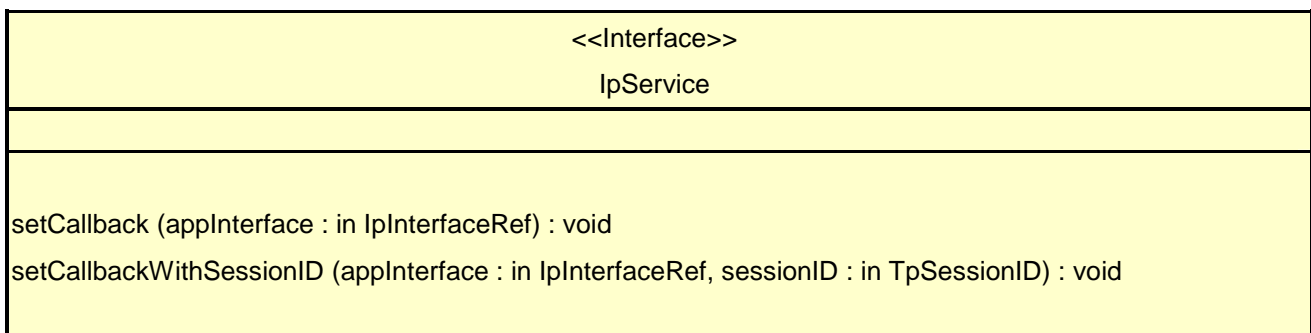
The interfaces that are implemented by the services are denoted as 'Service Interface'. The corresponding interfaces that must be implemented by the application (e.g. for API callbacks) are denoted as 'Application Interface'.

## 7.4 Generic Service Interface

### 7.4.1 Interface Class IpService

Inherits from: IpInterface

.All service interfaces inherit from the following interface.



### 7.4.1.1 Method setCallback()

This method specifies the reference address of the callback interface that a service uses to invoke methods on the application. It is not allowed to invoke this method on an interface that uses SessionIDs. Multiple invocations of this method on an interface shall result in multiple callback references being specified. The SCS shall use the most recent callback interface provided by the application using this method. In the event that a callback reference fails or is no longer available, the next most recent callback reference available shall be used.

#### *Parameters*

**appInterface : in IpInterfaceRef**

Specifies a reference to the application interface, which is used for callbacks.

#### *Raises*

**TpCommonExceptions, P\_INVALID\_INTERFACE\_TYPE**

### 7.4.1.2 Method setCallbackWithSessionID()

This method specifies the reference address of the application's callback interface that a service uses for interactions associated with a specific session ID: e.g. a specific call, or call leg. It is not allowed to invoke this method on an interface that does not use SessionIDs. Multiple invocations of this method on an interface shall result in multiple callback references being specified. The SCS shall use the most recent callback interface provided by the application using this method. In the event that a callback reference fails or is no longer available, the next most recent callback reference available shall be used.

#### *Parameters*

**appInterface : in IpInterfaceRef**

Specifies a reference to the application interface, which is used for callbacks.

**sessionID : in TpSessionID**

Specifies the session for which the service can invoke the application's callback interface.

#### *Raises*

**TpCommonExceptions, P\_INVALID\_SESSION\_ID, P\_INVALID\_INTERFACE\_TYPE**

## 8 Service Broker Interface Classes

The Service Broker SCF enables the application to register its interest in particular traffic as part of service interactions. The Service Broker service provides a SCF interface that is called IpServiceBroker. There is no need for an application interface, since IpServiceBroker only contains two synchronous methods register and unregister.

### 8.1 Interface Class IpServiceBroker

Inherits from: IpService;.

The ServiceBroker SCF interface IpServiceBroker contains two synchronous methods, register and unregister. The application has to provide its name, endpoint address and optionally a service identifier as input to the register method. The result indicates whether or not the service brokering scenario is available in the Service Broker SCF and, in case they are, it will return an assignment identifier in order to identify the particular interworking scenario.

An application may register multiple times with the same clientBrokerID. This is to facilitate, though not mandate, load sharing to be possible and the ability of two or more instances of an application to be involved in service interworking. Moreover, the same application may register with the service broker using more than one clientBrokerID to facilitate partitioning of services among subscribers.

<<Interface>> <b>IpServiceBroker</b>
<pre> register (clientBrokerID : in TpString, endpointAddress : in TpEndpointAddress, serviceKey : in     TpServiceKey) : TpAssignmentID unregister (assignmentID : in TpAssignmentID) : void </pre>

### 8.1.1 Method register()

This method is used by an application or SCF to register interest in a particular service interaction which has already been provisioned on the Service Broker entity.

The method may be called multiple times for individual instances of the same application or service i.e. individual instances using the same clientBrokerID. The behaviour of the Service Broker SCF for this scenario is regarded as implementation detail but may include such behaviour as round robinning of traffic to the applications or services identified by the clientBrokerID or implementing a primary/secondary hot standby traffic distribution for high availability.

The method may also be called multiple times by the same application instances but each identified by a unique clientBrokerID, in order to facilitate partitioning of subscribers. For example, where multiple charging platforms have been provisioned by subscriber number.

If two applications attempt to call register with the same clientBrokerID but on different service managers then a P\_INVALID\_CRITERIA exception will be returned.

Returns assignmentID: Specifies an instance of a registered service interaction. This is used by the application in order to unregister the service interaction at a later stage. If the service or application calls register multiple times with the same clientBrokerID, endpointAddress and serviceKey then the service Broker SCF will return the same assignmentID.

The method will return a unique assignmentID for each invocation of the register method specified with a unique clientBrokerID.

A P\_INVALID\_SERVICE\_INTERACTION is returned if the Service Broker entity has no prior knowledge of the service or application.

#### *Parameters*

**clientBrokerID : in TpString**

Identifies the name of the service or application requiring service interaction.

**endpointAddress : in TpEndpointAddress**

Identifies the network address of the service or application. This is to allow the Service Broker SCF to direct network traffic to the service or application at a later stage.

**serviceKey : in TpServiceKey**

Identifies the service for which applications require service interaction. Service interactions may be grouped or assigned by a single service key. This parameter is optional; if the application does not use this parameter then its value will be assigned NULL by the application.

#### *Returns*

**TpAssignmentID**

*Raises*

**TpCommonExceptions, P\_INVALID\_SERVICE\_INTERACTION, P\_INVALID\_CRITERIA**

**8.1.2 Method unregister()**

This method is used by a service or application to unregister previously registered service interactions on the Service Broker SCF.

As a result of calling this method, the service or application will no longer receive network traffic from the Service Broker SCF for that service interaction identified by the specific assignmentID. However, if the service or application has previously called register() more than once then it may still receive network traffic. In order to completely unregister from all service interactions, the service or applications must call unregister() for each previously allocated assignmentID.

The method returns P\_INVALID\_ASSIGNMENT\_ID if the supplied assignmentID value does not correspond to a previously returned assignmentID value via the register() method.

*Parameters*

**assignmentID : in TpAssignmentID**

Identifies the specific service interaction.

*Raises*

**TpCommonExceptions, P\_INVALID\_ASSIGNMENT\_ID**

**9 State Transition Diagrams**

There are no State Transition Diagrams for the Service Broker SCF.

**10 Service Broker Service Properties**

The following table lists properties relevant for the Service Broker API.

Property	Type	Description/Interpretation
P_ADDRESSPLAN	INTEGER_SET	Indicates the supported address plans (defined in TpAddressPlan.) E.g. P_ADDRESS_PLAN_IP. Note that more than one address plan may be supported.

**11 Data Definitions**

All data types referenced but not defined in this clause are common data definitions which may be found in ES 204 915-2.

**11.1 Service Broker Data Definitions****11.1.1 clientBrokerID**

Identifies the application or service requiring interaction.

Name	Type	Documentation
clientBrokerID	TpString	Identifies the application or service requiring the service interaction

### 11.1.2 TpEndpointAddress

This data type defines the Tagged Choice of Data Elements that specify the address of the end point to which network traffic should be sent as a result of service interactions.

	Tag Element Type	
	TpEndpointAddressCategory	

Tag Element Value	Choice Element Type	Choice Element Name
P_NETWORK_ADDRESS	TpAddress	NetworkAddress
P_SS7_ADDRESS	TpOctetSet	SS7Address

### 11.1.3 TpEndpointAddressCategory

Name	Value	Description
P_NETWORK_ADDRESS	0	Network address for protocol specific traffic
P_SS7_ADDRESS	1	SS7 Address for endpoint. For example, encoded Global Title or Point Code with SSN as specified in ITU-T Q.713.

### 11.1.4 TpServiceKey

Defines a Tagged Choice of Data Elements that specify services on which the application is requesting interaction.

	Tag Element Type	
	TpServiceKeyType	

Tag Element Value	Choice Element Type	Choice Element Name
P_SERVICE_KEY	TpInt32	ServiceKeyValue

### 11.1.5 TpServiceKeyType

Defines the type of service key used.

Name	Value	Description
P_SERVICE_KEY_UNDEFINED	0	Undefined
P_SERVICE_KEY	1	The service key value

---

## 12 Exception Classes

The following are the list of exception classes that are used in this interface of the API.

Name	Description
P_INVALID_SERVICE_INTERACTION	The request cannot be processed as there is insufficient information for the Service Broker SCF to carry out the service interaction.

Each exception class contains the following structure:

Structure Element Name	Structure Element Type	Structure Element Description
ExtraInformation	TpString	Carries extra information to help identify the source of the exception, e.g. a parameter name

---

## Annex A (normative): OMG IDL Description of Service Broker SCF

The OMG IDL representation of this interface specification is contained in a text file (svcbroker.idl contained in archive es\_20491516IDL.zip) which accompanies the present document.

This archive can be found in es\_20491516v010101p0.zip which accompanies the present document.

---

## Annex B (informative): W3C WSDL Description of Service Broker SCF

The W3C WSDL representation of this interface specification is contained in zip file es\_20491516WSDL.zip which accompanies the present document.

This archive can be found in es\_20491516v010101p0.zip which accompanies the present document.

---

## Annex C (informative): Java API Description of the Service Broker SCF

The Java™ API realisation of this interface specification is produced in accordance with the Java™ Realisation rules defined in ES 204 915-1. These rules aim to deliver for Java™, a developer API, provided as a realisation, supporting a Java™ API that represents the UML specifications. The rules support the production of both J2SE™ and J2EE™ versions of the API from the common UML specifications.

The J2SE™ representation of this interface specification is provided as Java™ Code, contained in archive 20491516J2SE.zip that accompanies the present document.

The J2EE™ representation of this interface specification is provided as Java™ Code, contained in archive 20491516J2EE.zip that accompanies the present document.

Both these archives can be found in es\_20491516v010101p0.zip which accompanies the present document.



---

## Annex D (informative): Contents of 3GPP OSA R7 Service Broker

All of the present document is relevant for TS 129 198-16 V7 (Release 7).

---

## Annex E (informative): Description of Service Broker for 3GPP2 cdma2000 networks

This annex is intended to define the OSA API Stage 3 interface definitions and it provides the complete OSA specifications. It is an extension of OSA API specifications capabilities to enable operation in cdma2000 systems environment. They are in alignment with 3GPP2 Stage 1 requirements and Stage 2 architecture defined in [52], [53] and [54] of ES 204 915-1, clause 2. These requirements are expressed as additions to and/or exclusions from the 3GPP Release 7 specification. The information given here is to be used by developers in 3GPP2 cdma2000 network architecture to interpret the 3GPP OSA specifications.

---

### E.1 General Exceptions

The term UMTS is not applicable for the cdma2000 family of standards. Nevertheless these terms are used (TR 121 905) mostly in the broader sense of "3G Wireless System". If not stated otherwise there are no additions or exclusions required.

CAMEL and CAP mappings are not applicable for cdma2000 systems.

---

### E.2 Specific Exceptions

#### E.2.1 Clause 1: Scope

There are no additions or exclusions.

#### E.2.2 Clause 2: References

Normative references on TS 123 078 and on TS 129 078 are not applicable for cdma2000 systems.

#### E.2.3 Clause 3: Definitions and abbreviations

There are no additions or exclusions.

#### E.2.4 Clause 4: Service Broker SCF

There are no additions or exclusions.

#### E.2.5 Clause 5: Sequence Diagrams

There are no additions or exclusions.

#### E.2.6 Clause 6: Class Diagrams

There are no additions or exclusions.

#### E.2.7 Clause 7: The Service Interface Specifications

There are no additions or exclusions.

## **E.2.8 Clause 8: Service Broker Interface Classes**

There are no additions or exclusions.

## **E.2.9 Clause 9: State Transition Diagrams**

There are no additions or exclusions.

## **E.2.10 Clause 10: Service Broker Service Properties**

There are no additions or exclusions.

## **E.2.11 Clause 11: Data Definitions**

There are no additions or exclusions.

## **E.2.12 Annex A (normative): OMG IDL Description of Service Broker SCF**

There are no additions or exclusions.

## **E.2.13 Annex B (informative): W3C WSDL Description of Service Broker SCF**

There are no additions or exclusions.

---

## Annex F (informative): Record of changes

The following is a list of the changes made to the present document for each release. The list contains the names of all changed, deprecated, added or removed items in the specifications and not the actual changes. Any type of change information that is important to the reader is put in the final clause of this annex.

Changes are specified as changes to the prior major release, but every minor release will have its own part of the table allowing the reader to know when the actual change was made.

---

### F.1 Interfaces

#### F.1.1 New

Identifier	Comments
<b>Interfaces added in ES 204 915-16 version 1.1.1 (Parlay 6.0)</b>	

#### F.1.2 Deprecated

Identifier	Comments
<b>Interfaces deprecated in ES 204 915-16 version 1.1.1 (Parlay 6.0)</b>	

#### F.1.3 Removed

Identifier	Comments
<b>Interfaces removed in ES 204 915-16 version 1.1.1 (Parlay 6.0)</b>	

---

### F.2 Methods

#### F.2.1 New

Identifier	Comments
<b>Methods added in ES 204 915-16 version 1.1.1 (Parlay 6.0)</b>	

#### F.2.2 Deprecated

Identifier	Comments
<b>Methods deprecated in ES 204 915-16 version 1.1.1 (Parlay 6.0)</b>	

### F.2.3 Modified

Identifier	Comments
Methods modified in ES 204 915-16 version 1.1.1 (Parlay 6.0)	

### F.2.4 Removed

Identifier	Comments
Methods removed in ES 204 915-16 version 1.1.1 (Parlay 6.0)	

---

## F.3 Data Definitions

### F.3.1 New

Identifier	Comments
Data Definitions added in ES 204 915-16 version 1.1.1 (Parlay 6.0)	

### F.3.2 Modified

Identifier	Comments
Data Definitions modified in ES 204 915-16 version 1.1.1 (Parlay 6.0)	

### F.3.3 Removed

Identifier	Comments
Data Definitions removed in ES 204 915-16 version 1.1.1 (Parlay 6.0)	

---

## F.4 Service Properties

### F.4.1 New

Identifier	Comments
Service Properties added in ES 204 915-16 version 1.1.1 (Parlay 6.0)	

## F.4.2 Deprecated

Identifier	Comments
Service Properties deprecated in ES 204 915-16 version 1.1.1 (Parlay 6.0)	

## F.4.3 Modified

Identifier	Comments
Service Properties modified in ES 204 915-16 version 1.1.1 (Parlay 6.0)	

## F.4.4 Removed

Identifier	Comments
Service Properties removed in ES 204 915-16 version 1.1.1 (Parlay 6.0)	

---

## F.5 Exceptions

### F.5.1 New

Identifier	Comments
Exceptions added in ES 204 915-16 version 1.1.1 (Parlay 6.0)	

### F.5.2 Modified

Identifier	Comments
Exceptions modified in ES 204 915-16 version 1.1.1 (Parlay 6.0)	

### F.5.3 Removed

Identifier	Comments
Exceptions removed in ES 204 915-16 version 1.1.1 (Parlay 6.0)	

---

## F.6 Others

None.

---

## History

<b>Document history</b>		
V1.1.1	February 2008	Membership Approval Procedure    MV 20080425: 2008-02-26 to 2008-04-25
V1.1.1	May 2008	Publication