3GPP TS 29.198-16 V9.0.0 (2009-12)

Technical Specification

3rd Generation Partnership Project;
Technical Specification Group Core Network;
Open Service Access (OSA);
Application Programming Interface (API);
Part 16: Service Broker Service Capability Feature (SCF)
(Release 9)

GLOBAL SYSTEM FOR
MOBILE COMMUNICATIONS

Keywords
UMTS, API, OSA

3GPP

Postal address

3GPP support office address
650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet
http://www.3gpp.org

ETSI

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 16 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";	
	Sub-part 1: "Call Control Common Definitions";	
	Sub-part 2: "Generic Call Control SCF";	
	Sub-part 3: "Multi-Party Call Control SCF";	
	Sub-part 4: "Multi-Media Call Control SCF";	
	Sub-part 5: "Conference 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 8)
Part 10:	"Connectivity Manager SCF";	(new in Release 8)
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";8**	

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.

Table: Overview of the OSA APIs & Protocol Mappings 29.198 & 29.998-family

OSA API specifications 29.198-family						OSA API Mapping - 29.998-family	
29.198-01	Overview					29.998-01	Overview
29.198-02	Common Data Definitions					*29.998-02*	*Not Applicable*
29.198-03	Framework					*29.998-03*	*Not Applicable*
Call Control (CC) SCF	29.198-04-1 Common CC data definitions	29.198-04-2 Generic CC SCF	29.198-04-3 Multi-Party CC SCF	29.198-04-4 Multi-media CC SCF	29.198-04-5 Conference Call Control SCF	29.998-04-1	Generic Call Control – CAP mapping
						29.998-04-2	*Generic Call Control – INAP mapping*
						29.998-04-3	*Generic Call Control – Megaco mapping*
						29.998-04-4	Multiparty Call Control – ISC mapping
29.198-05	User Interaction SCF					29.998-05-1	User Interaction – CAP mapping
						29.998-05-2	*User Interaction – INAP mapping*
						29.998-05-3	*User Interaction – Megaco mapping*
						29.998-05-4	User Interaction – SMS mapping
29.198-06	Mobility SCF					29.998-06	User Status and User Location – MAP mapping
29.198-07	Terminal Capabilities SCF					*29.998-07*	*Not Applicable*
29.198-08	Data Session Control SCF					29.998-08	Data Session Control – CAP mapping
29.198-09	*Generic Messaging SCF*					29.998-09	*Not Applicable*
29.198-10	Connectivity Manager SCF					*29.998-10*	*Not Applicable*
29.198-11	Account Management SCF					*29.998-11*	*Not Applicable*
29.198-12	Charging SCF					*29.998-12*	*Not Applicable*
29.198-13	Policy Management SCF					*29.998-13*	*Not Applicable*
29.198-14	Presence & Availability Management SCF					*29.998-14*	*Not Applicable*
29.198-15	Multi-media Messaging SCF					*29.998-15*	*Not Applicable*
29.198-16	**Service Broker SCF**					*29.998-16*	*Not Applicable*

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 concepts and the functional architecture for the OSA are contained in 3GPP TS 23.198 [3]. The requirements for OSA are contained in 3GPP TS 22.127 [2].

The present document specifies the Service Broker 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

- WSDL 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).

This specification has been defined jointly between 3GPP TSG CT WG5, ETSI TISPAN and the Parlay Group, in co-operation with a number of JAIN™ Community member companies.

Maintenance of up to 3GPP Rel-8 and new OSA Stage 1, 2 and 3 work beyond Rel-9 was moved to OMA in June 2008.

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 TS 29.198-1 "Open Service Access; Application Programming Interface; Part 1: Overview".

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

[3]   3GPP TS 23.198: "Open Service Access (OSA); Stage 2".

[4]   3GPP TS 29.198-2: "Open Service Access (OSA) Application Programming Interface (API); Part 2: Common data definitions".

[5]   ITU-T Recommendation Q.713: "Signalling connection control part formats and codes".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TS 29.198-1 [1] apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in TS 29.198-1 [1] apply.

4 Service Broker SCF

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

The order is as follows:

- The Sequence diagrams give the reader a practical idea of how each SCF 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 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 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 this specification.

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.
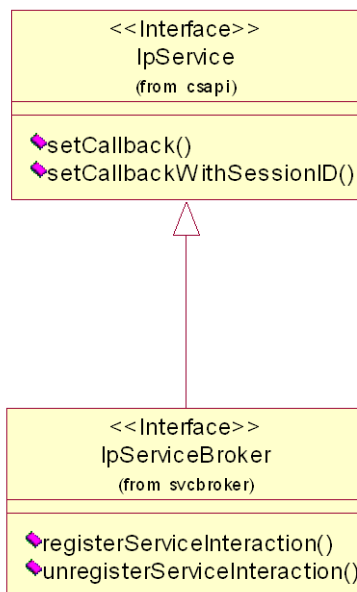
6 Class Diagrams

```
                    ┌────────────────────────────┐
                    │       <<Interface>>        │
                    │         IpService          │
                    │       (from csapi)         │
                    ├────────────────────────────┤
                    ├────────────────────────────┤
                    │ ◆setCallback()             │
                    │ ◆setCallbackWithSessionID()│
                    └────────────────────────────┘
                                  △
                                  │
                                  │
                    ┌────────────────────────────┐
                    │       <<Interface>>        │
                    │       IpServiceBroker      │
                    │      (from svcbroker)      │
                    ├────────────────────────────┤
                    ├────────────────────────────┤
                    │ ◆registerServiceInteraction()   │
                    │ ◆unregisterServiceInteraction() │
                    └────────────────────────────┘
```

Figure: 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.
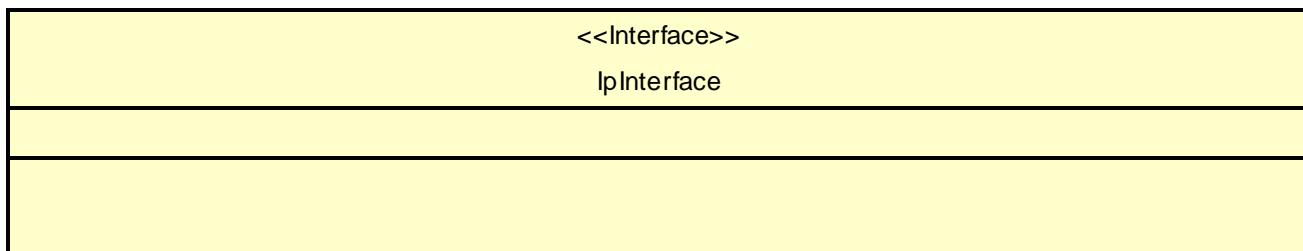
7.1.4   State Model

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.

<<Interface>>
IpInterface

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.

<<Interface>>
IpService
setCallback (appInterface : in IpInterfaceRef) : void setCallbackWithSessionID (appInterface : in IpInterfaceRef, sessionID : in TpSessionID) : void
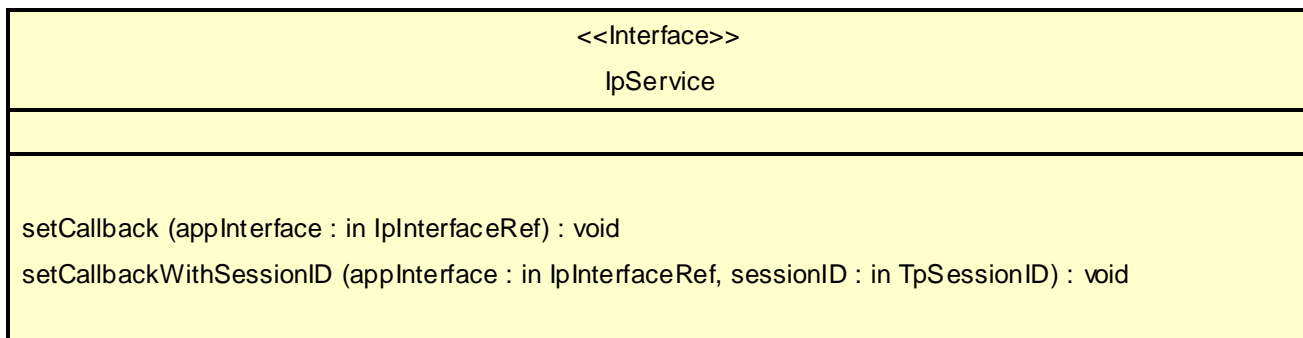
7.4.1.1   Method setCallback()

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 registerServiceInteraction and unregisterServiceInteraction.

8.1 Interface Class IpServiceBroker

Inherits from: IpService.

The ServiceBroker SCF interface IpServiceBroker contains two synchronous methods, registerServiceInteraction and unregisterServiceInteraction. The application has to provide its name, endpoint address and optionally a service identifier as input to the registerServiceInteraction 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>>
                          IpServiceBroker

registerServiceInteraction (clientBrokerID : in TpString, endpointAddress : in TpEndpointAddress,
    serviceKey : in TpServiceKey) : TpAssignmentID
unregisterServiceInteraction (assignmentID : in TpAssignmentID) : void
```

8.1.1 Method registerServiceInteraction()

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 robining 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 registerServiceInteraction() 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 registerServiceInteraction() multiple times with the same clientBrokerID, endpointAddress and serviceKey then the service Broker SCF will return the same assignmentID.

The method will return an unique assignmentID for each invocation of the registerServiceInteraction() method specified with an 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 unregisterServiceInteraction()

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 registerServiceInteraction() 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 unregisterServiceInteraction() 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 registerServiceInteraction() 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 3GPP TS 29.198-2 [4].

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 [5]

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 2919816V800IDL.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 2919816V800WSDL.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 Part 1 of this specification series. 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 2919816V800J2SE.ZIP that accompanies the present document.

The J2EE™ representation of this interface specification is provided as Java™ Code, contained in archive 2919816V800J2EE.ZIP that accompanies the present document.

Annex D (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:

[1] 3GPP2 P.S0001-B: "Wireless IP Network Standard", Version 1.0, September 2000.

[2] 3GPP2 S.R0037-0: "IP Network Architecture Model for cdma2000 Spread Spectrum Systems", Version 2.0, May 14, 2002.

[3] 3GPP2 X.S0013: "All-IP Core Network Multimedia Domain", December 2003.

These requirements are expressed as additions to and/or exclusions from the 3GPP specification.
The information given here is to be used by developers in 3GPP2 cdma2000 network architecture to interpret the 3GPP OSA specifications.

D.1 General Exceptions

The term UMTS is not applicable for the cdma2000 family of standards. Nevertheless these terms are used (3GPP TR 21.905) 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.

D.2 Specific Exceptions

D.2.1 Clause 1: Scope

There are no additions or exclusions.

D.2.2 Clause 2: References

Normative references on 3GPP TS 23.078 and on 3GPP TS 29.078 are not applicable for cdma2000 systems.

D.2.3 Clause 3: Definitions and abbreviations

There are no additions or exclusions.

D.2.4 Clause 4: Service Broker SCF

There are no additions or exclusions.

D.2.5 Clause 5: Sequence Diagrams

There are no additions or exclusions.

D.2.6 Clause 6 Class Diagrams

There are no additions or exclusions.

D.2.7 Clause 7: The Service Interface Specifications

There are no additions or exclusions.

D.2.8 Clause 8: Service Broker Interface Classes

There are no additions or exclusions.

D.2.9 Clause 9: State Transition Diagrams

There are no additions or exclusions.

D.2.10 Clause 10: Service Broker Service Properties

There are no additions or exclusions.

D.2.11 Clause 11: Data Definitions

There are no additions or exclusions.

D.2.12 Annex A (normative): OMG IDL Description of Service Broker SCF

There are no additions or exclusions.

D.2.13 Annex B (informative): W3C WSDL Description of Service Broker SCF

There are no additions or exclusions.

Annex E (informative):
Change history

Change history								
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Cat**	**Old**	**New**
Jun 2006	CT_32	CP-060212	--	--	Submitted to TSG CT#32 for Information. Aligned with Stage 2 CR 23.198 in CP-060204.	--	1.0.0	
Nov 2006	CT_34	CP-060608	--	--	Submitted to TSG CT#34 for Approval.	--	2.0.0	7.0.0
Dec 2008	CT_42				Upgraded unchanged from Rel-7		7.0.0	8.0.0
2009-12	-	-	-	-	Update to Rel-9 version (MCC)		8.0.0	9.0.0