



**SLOVENSKI STANDARD**  
**SIST ES 201 915-4 V1.4.1:2005**  
**01-januar-2005**

---

**Odpri dostop do storitve (OSA) – Vmesnik za aplikacijsko programiranje (API) – 4.**  
**del: Krmiljenje klica SCF**

Open Service Access (OSA); Application Programming Interface (API); Part 4: Call Control SCF (Parlay 3)

**iTeh STANDARD PREVIEW**  
**(standards.iteh.ai)**

Ta slovenski standard je istoveten z: **ES 201 915-4 Version 1.4.1**

SIST ES 201 915-4 V1.4.1:2005  
<https://standards.iteh.ai/catalog/standards/sis/31122eaf-dc1a-4597-9b38-06e77b7e70cb/sist-es-201-915-4-v1-4-1-2005>

---

**ICS:**

33.040.01	Telekomunikacijski sistemi na splošno	Telecommunication systems in general
-----------	--	---

**SIST ES 201 915-4 V1.4.1:2005**                      **en**

**iTeh STANDARD PREVIEW**  
**(standards.iteh.ai)**

[SIST ES 201 915-4 V1.4.1:2005](https://standards.iteh.ai/catalog/standards/sist/3ff22eaf-dcfa-4397-9b38-06e77b7e70cb/sist-es-201-915-4-v1-4-1-2005)

<https://standards.iteh.ai/catalog/standards/sist/3ff22eaf-dcfa-4397-9b38-06e77b7e70cb/sist-es-201-915-4-v1-4-1-2005>

# ETSI ES 201 915-4 V1.4.1 (2003-07)

ETSI Standard

## Open Service Access (OSA); Application Programming Interface (API); Part 4: Call Control SCF (Parlay 3)



### iTeh STANDARD PREVIEW (standards.iteh.ai)

[SIST ES 201 915-4 V1.4.1:2005](https://standards.iteh.ai/catalog/standards/sist/3ff22eaf-dcfa-4397-9b38-06e77b7e70cb/sist-es-201-915-4-v1-4-1-2005)

<https://standards.iteh.ai/catalog/standards/sist/3ff22eaf-dcfa-4397-9b38-06e77b7e70cb/sist-es-201-915-4-v1-4-1-2005>



## Reference

---

RES/SPAN-120095-4

## Keywords

---

API, OSA, IDL, UML**ETSI**

---

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

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

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

**iTeh STANDARD PREVIEW**  
**(standards.iteh.ai)**

SIST ES 201 915-4 V1.4.1:2005

<https://standards.iteh.ai/catalog/standards/sist/3ff22eaf-dcfa-4397-9b38-06e77b7e704d/etsi-es-201-915-4-v1-4-1-2005>  
**Important notice**

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

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

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

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

If you find errors in the present document, send your comment to:

[editor@etsi.org](mailto:editor@etsi.org)

---

**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 2003.

© The Parlay Group 2003.

All rights reserved.

**DECT™**, **PLUGTESTS™** and **UMTS™** are Trade Marks of ETSI registered for the benefit of its Members.  
**TIPHON™** and the **TIPHON logo** are Trade Marks currently being registered by ETSI for the benefit of its Members.  
**3GPP™** is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

# Contents

Intellectual Property Rights .....	9
Foreword.....	9
1 Scope .....	10
2 References .....	10
3 Definitions and abbreviations.....	10
3.1 Definitions .....	10
3.2 Abbreviations .....	10
4 Call Control SCF .....	10
4.1 Call Model Description .....	11
4.2 General requirements on support of methods.....	12
5 The Service Interface Specifications .....	12
5.1 Interface Specification Format .....	12
5.1.1 Interface Class .....	12
5.1.2 Method descriptions.....	12
5.1.3 Parameter descriptions.....	12
5.1.4 State Model.....	12
5.2 Base Interface.....	12
5.2.1 Interface Class IpInterface .....	12
5.3 Service Interfaces .....	13
5.3.1 Overview .....	13
5.4 Generic Service Interface .....	13
5.4.1 Interface Class IpService .....	13
6 Generic Call Control Service .....	14
6.1 Sequence Diagrams.....	14
6.1.1 Additional Callbacks.....	14
6.1.2 Alarm Call .....	16
6.1.3 Application Initiated Call.....	18
6.1.4 Call Barring 1 .....	20
6.1.5 Number Translation 1 .....	22
6.1.6 Number Translation 1 (with callbacks).....	24
6.1.7 Number Translation 2 .....	26
6.1.8 Number Translation 3 .....	28
6.1.9 Number Translation 4 .....	30
6.1.10 Number Translation 5 .....	32
6.1.11 Prepaid .....	33
6.1.12 Pre-Paid with Advice of Charge (AoC) .....	35
6.2 Class Diagrams.....	38
6.3 Generic Call Control Service Interface Classes.....	39
6.3.1 Interface Class IpCallControlManager .....	40
6.3.2 Interface Class IpAppCallControlManager .....	43
6.3.3 Interface Class IpCall.....	46
6.3.4 Interface Class IpAppCall.....	50
6.4 Generic Call Control Service State Transition Diagrams .....	54
6.4.1 State Transition Diagrams for IpCallControlManager.....	54
6.4.1.1 Active State .....	55
6.4.1.2 Notification terminated State .....	55
6.4.2 State Transition Diagrams for IpCall.....	56
6.4.2.1 Network Released State .....	56
6.4.2.2 Finished State.....	56
6.4.2.3 Application Released State .....	57
6.4.2.4 No Parties State.....	57
6.4.2.5 Active State .....	57
6.4.2.6 1 Party in Call State.....	57

6.4.2.7	2 Parties in Call State .....	57
6.4.2.8	Routing to Destination(s) State .....	58
6.5	Generic Call Control Service Properties .....	58
6.5.1	List of Service Properties.....	58
6.5.2	Service Property values for the CAMEL Service Environment.....	59
6.6	Generic Call Control Data Definitions .....	59
6.6.1	Generic Call Control Event Notification Data Definitions .....	60
6.6.1.1	TpCallEventName.....	60
6.6.1.2	TpCallNotificationType .....	60
6.6.1.3	TpCallEventCriteria .....	61
6.6.1.4	TpCallEventInfo.....	61
6.6.1.5	Generic Call Control Data Definitions .....	61
6.6.1.6	IpCall.....	61
6.6.1.7	IpCallRef.....	61
6.6.1.8	IpAppCall.....	61
6.6.1.9	IpAppCallRef .....	61
6.6.1.10	TpCallIdentifier.....	61
6.6.1.11	IpAppCallControlManager .....	62
6.6.1.12	IpAppCallControlManagerRef .....	62
6.6.1.13	IpCallControlManager .....	62
6.6.1.14	IpCallControlManagerRef .....	62
6.6.1.15	TpCallAppInfo .....	62
6.6.1.16	TpCallAppInfoType .....	63
6.6.1.17	TpCallAppInfoSet .....	63
6.6.1.18	TpCallEndedReport.....	63
6.6.1.19	TpCallFault .....	63
6.6.1.20	TpCallInfoReport.....	63
6.6.1.21	TpCallReleaseCause .....	64
6.6.1.22	TpCallReport.....	64
6.6.1.23	TpCallAdditionalReportInfo .....	65
6.6.1.24	TpCallReportRequest.....	65
6.6.1.25	TpCallAdditionalReportCriteria.....	65
6.6.1.26	TpCallReportRequestSet.....	66
6.6.1.27	TpCallReportType .....	66
6.6.1.28	TpCallTreatment .....	66
6.6.1.29	TpCallEventCriteriaResultSet.....	66
6.6.1.30	TpCallEventCriteriaResult .....	66
7	MultiParty Call Control Service.....	67
7.1	Sequence Diagrams .....	67
7.1.1	Application initiated call setup .....	67
7.1.2	Call Barring 2 .....	69
7.1.3	Call forwarding on Busy Service .....	70
7.1.4	Call Information Collect Service .....	71
7.1.5	Complex Card Service .....	74
7.1.6	Hotline Service .....	77
7.1.7	Use of the Redirected event .....	80
7.2	Class Diagrams.....	80
7.3	MultiParty Call Control Service Interface Classes .....	82
7.3.1	Interface Class IpMultiPartyCallControlManager .....	82
7.3.2	Interface Class IpAppMultiPartyCallControlManager .....	86
7.3.3	Interface Class IpMultiPartyCall .....	88
7.3.4	Interface Class IpAppMultiPartyCall.....	93
7.3.5	Interface Class IpCallLeg .....	95
7.3.6	Interface Class IpAppCallLeg.....	101
7.4	MultiParty Call Control Service State Transition Diagrams .....	106
7.4.1	State Transition Diagrams for IpMultiPartyCallControlManager.....	106
7.4.1.1	Active State .....	106
7.4.1.2	Interrupted State .....	106
7.4.1.3	Overview of allowed methods.....	106
7.4.2	State Transition Diagrams for IpMultiPartyCall.....	107
7.4.2.1	IDLE State.....	107

7.4.2.2	ACTIVE State .....	108
7.4.2.3	RELEASED State .....	108
7.4.2.4	Overview of allowed methods.....	108
7.4.3	State Transition Diagrams for IpCallLeg.....	108
7.4.3.1	Originating Call Leg .....	109
7.4.3.1.1	Initiating State .....	110
7.4.3.1.2	Analysing State.....	111
7.4.3.1.3	Active State .....	112
7.4.3.1.4	Releasing State .....	114
7.4.3.1.5	Overview of allowed methods, Originating Call Leg STD .....	116
7.4.3.2	Terminating Call Leg .....	117
7.4.3.2.1	Idle (terminating) State.....	117
7.4.3.2.2	Active (terminating) State .....	118
7.4.3.2.3	Releasing (terminating) State .....	121
7.4.3.2.4	Overview of allowed methods and trigger events, Terminating Call Leg STD.....	123
7.5	Multi-Party Call Control Service Properties .....	123
7.5.1	List of Service Properties.....	123
7.5.2	Service Property values for the CAMEL Service Environment.....	124
7.6	Multi-Party Call Control Data Definitions .....	125
7.6.1	Event Notification Data Definitions .....	125
7.6.2	Multi-Party Call Control Data Definitions.....	125
7.6.2.1	IpCallLeg .....	125
7.6.2.2	IpCallLegRef.....	125
7.6.2.3	IpAppCallLeg.....	125
7.6.2.4	IpAppCallLegRef.....	125
7.6.2.5	IpMultiPartyCall .....	125
7.6.2.6	IpMultiPartyCallRef.....	125
7.6.2.7	IpAppMultiPartyCall.....	125
7.6.2.8	IpAppMultiPartyCallRef.....	126
7.6.2.9	IpMultiPartyCallControlManager .....	126
7.6.2.10	IpMultiPartyCallControlManagerRef .....	126
7.6.2.11	IpAppMultiPartyCallControlManager .....	126
7.6.2.12	IpAppMultiPartyCallControlManagerRef .....	126
7.6.2.13	TpAppCallLegRefSet.....	126
7.6.2.14	TpMultiPartyCallIdentifier .....	126
7.6.2.15	TpAppMultiPartyCallBack .....	126
7.6.2.16	TpAppMultiPartyCallBackRefType .....	127
7.6.2.17	TpAppCallLegCallBack.....	127
7.6.2.18	TpMultiPartyCallIdentifierSet .....	127
7.6.2.19	TpCallAppInfo .....	127
7.6.2.20	TpCallAppInfoType .....	128
7.6.2.21	TpCallAppInfoSet.....	128
7.6.2.22	TpCallEventRequest .....	128
7.6.2.23	TpCallEventRequestSet .....	128
7.6.2.24	TpCallEventType .....	129
7.6.2.25	TpAdditionalCallEventCriteria .....	131
7.6.2.26	TpCallEventInfo.....	131
7.6.2.27	TpCallAdditionalEventInfo.....	132
7.6.2.28	TpCallNotificationRequest .....	132
7.6.2.29	TpCallNotificationScope.....	132
7.6.2.30	TpCallNotificationInfo.....	133
7.6.2.31	TpCallNotificationReportScope.....	133
7.6.2.32	TpNotificationRequested .....	133
7.6.2.33	TpNotificationRequestedSet .....	133
7.6.2.34	TpReleaseCause .....	133
7.6.2.35	TpReleaseCauseSet .....	133
7.6.2.36	TpCallLegIdentifier.....	134
7.6.2.37	TpCallLegIdentifierSet .....	134
7.6.2.38	TpCallLegAttachMechanism .....	134
7.6.2.39	TpCallLegConnectionProperties .....	134
7.6.2.40	TpCallLegInfoReport.....	134
7.6.2.41	TpCallLegInfoType.....	135

7.6.2.42	TpCallLegSuperviseTreatment .....	135
8	MultiMedia Call Control Service .....	135
8.1	Sequence Diagrams .....	135
8.1.1	Barring for media combined with call routing, alternative 1 .....	135
8.1.2	Barring for media combined with call routing, alternative 2 .....	137
8.1.3	Barring for media, simple .....	139
8.1.4	Call Volume charging supervision .....	140
8.2	Class Diagrams .....	142
8.3	MultiMedia Call Control Service Interface Classes .....	143
8.3.1	Interface Class IpMultiMediaCallControlManager .....	144
8.3.2	Interface Class IpAppMultiMediaCallControlManager .....	146
8.3.3	Interface Class IpMultiMediaCall .....	147
8.3.4	Interface Class IpAppMultiMediaCall .....	148
8.3.5	Interface Class IpMultiMediaCallLeg .....	149
8.3.6	Interface Class IpAppMultiMediaCallLeg .....	150
8.3.7	Interface Class IpMultiMediaStream .....	151
8.4	MultiMedia Call Control Service State Transition Diagrams .....	152
8.5	Multi-Media Call Control Data Definitions .....	152
8.5.1	Event Notification Data Definitions .....	152
8.5.1.1	TpMediaStreamRequestSet .....	152
8.5.1.2	TpMediaStreamRequest .....	152
8.5.1.3	TpMediaStreamDirection .....	153
8.5.1.4	TpMediaStreamDataTypeRequest .....	153
8.5.1.5	TpAudioCapabilitiesType .....	153
8.5.1.6	TpVideoCapabilitiesType .....	154
8.5.1.7	TpDataCapabilities .....	154
8.5.1.8	TpMediaStreamEventType .....	154
8.5.1.9	TpMediaStreamSet .....	154
8.5.1.10	TpMediaStream .....	154
8.5.1.11	TpMediaStreamDataType .....	154
8.5.2	Multi-Media Call Control Data Definitions .....	154
8.5.2.1	IpMultiMediaCall .....	154
8.5.2.2	IpMultiMediaCallRef .....	154
8.5.2.3	IpAppMultiMediaCall .....	155
8.5.2.4	IpAppMultiMediaCallRef .....	155
8.5.2.5	IpMultiMediaCallLeg .....	155
8.5.2.6	IpMultiMediaCallLegRef .....	155
8.5.2.7	IpAppMultiMediaCallLeg .....	155
8.5.2.8	IpAppMultiMediaCallLegRef .....	155
8.5.2.9	TpAppMultiMediaCallLegRefSet .....	155
8.5.2.10	TpMultiMediaCallIdentifier .....	155
8.5.2.11	TpMultiMediaCallIdentifierSet .....	155
8.5.2.12	TpMultiMediaCallLegIdentifier .....	155
8.5.2.13	TpMultiMediaCallLegIdentifierSet .....	155
8.5.2.14	IpAppMultiMediaCallControlManager .....	156
8.5.2.15	IpAppMultiMediaCallControlManagerRef .....	156
8.5.2.16	TpAppMultiMediaCallBack .....	156
8.5.2.17	TpAppMultiMediaCallBackRefType .....	156
8.5.2.18	TpAppMultiMediaCallLegCallBack .....	156
8.5.2.19	TpCallSuperviseVolume .....	157
8.5.2.20	TpNotificationMediaRequest .....	157
8.5.2.21	TpMediaNotificationRequested .....	157
8.5.2.22	TpMediaNotificationsRequestedSet .....	157
9	Conference Call Control Service .....	157
9.1	Sequence Diagrams .....	157
9.1.1	Meet-me conference without subconferencing .....	157
9.1.2	Non-add hoc add-on with subconferencing .....	159
9.1.3	Non-addhoc add-on multimedia .....	161
9.1.4	Resource Reservation .....	164
9.2	Class Diagrams .....	165



9.3	Conference Call Control Service Interface Classes.....	166
9.3.1	Interface Class IpConfCallControlManager .....	167
9.3.2	Interface Class IpAppConfCallControlManager.....	170
9.3.3	Interface Class IpConfCall.....	171
9.3.4	Interface Class IpAppConfCall.....	172
9.3.5	Interface Class IpSubConfCall .....	174
9.3.6	Interface Class IpAppSubConfCall.....	177
9.4	Conference Call Control Service State Transition Diagrams .....	178
9.5	Conference Call Control Data Definitions .....	178
9.5.1	Event Notification Data Definitions .....	179
9.5.2	Conference Call Control Data Definitions.....	179
9.5.2.1	IpConfCall.....	179
9.5.2.2	IpConfCallRef.....	179
9.5.2.3	IpAppConfCall.....	179
9.5.2.4	IpAppConfCallRef.....	179
9.5.2.5	IpSubConfCall .....	179
9.5.2.6	IpSubConfCallRef.....	179
9.5.2.7	IpAppSubConfCall.....	179
9.5.2.8	IpAppSubConfCallRef.....	179
9.5.2.9	TpSubConfCallIdentifierSet.....	179
9.5.2.10	TpConfCallIdentifier.....	180
9.5.2.11	TpSubConfCallIdentifier.....	180
9.5.2.12	IpAppConfCallControlManager.....	180
9.5.2.13	IpAppConfCallControlManagerRef.....	180
9.5.2.14	TpConfPolicyType.....	180
9.5.2.15	TpConfPolicy .....	180
9.5.2.16	TpMonoMediaConfPolicy.....	181
9.5.2.17	TpJoinEventInfo.....	181
9.5.2.18	TpConfSearchCriteria .....	181
9.5.2.19	TpConfSearchResult .....	181
9.5.2.20	TpMultiMediaConfPolicy .....	182
9.5.2.21	TpResourceReservation .....	182
9.5.2.22	TpVideoHandlingType.....	182
10	Common Call Control Data Types.....	182
10.1	TpCallAlertingMechanism .....	182
10.2	TpCallBearerService .....	183
10.3	TpCallChargePlan .....	183
10.4	TpCallPartyToChargeAdditionalInfo.....	183
10.5	TpCallPartyToChargeType .....	184
10.6	TpCallChargeOrderCategory .....	184
10.7	TpCallEndedReport.....	184
10.8	TpCallError .....	184
10.9	TpCallAdditionalErrorInfo.....	185
10.10	TpCallErrorType .....	185
10.11	TpCallInfoReport .....	185
10.12	TpCallInfoType.....	186
10.13	TpCallLoadControlMechanism.....	186
10.14	TpCallLoadControlIntervalRate .....	186
10.15	TpCallLoadControlMechanismType.....	186
10.16	TpCallMonitorMode .....	187
10.17	TpCallNetworkAccessType .....	187
10.18	TpCallPartyCategory.....	187
10.19	TpCallServiceCode .....	188
10.20	TpCallServiceCodeSet .....	188
10.21	TpCallServiceCodeType .....	188
10.22	TpCallSuperviseReport .....	188
10.23	TpCallSuperviseTreatment.....	189
10.24	TpCallTeleService.....	189
10.25	TpCallTreatment .....	190
10.26	TpCallTreatmentType .....	190
10.27	TpCallAdditionalTreatmentInfo.....	190

ITih STANDARD PREVIEW

(standards.iteh.ai)

SIST ES 201 915-4 V1.4.1:2005

standards.iteh.ai/catalog/standards/sist/3ff22eaf-dcf4-4397-9b38-

06e77b7e70cb/sist-es-201-915-4-v1-4-1-2005

10.28	TpMediaType .....	190
<b>Annex A (normative):</b>	<b>OMG IDL Description of Call Control SCF .....</b>	<b>191</b>
<b>Annex B (informative):</b>	<b>Contents of 3GPP OSA R4 Call Control .....</b>	<b>192</b>
<b>Annex C (informative):</b>	<b>Record of changes .....</b>	<b>193</b>
C.1	Interfaces .....	193
C.1.1	New .....	193
C.1.2	Deprecated.....	193
C.1.3	Removed.....	193
C.2	Methods .....	193
C.2.1	New .....	193
C.2.2	Deprecated.....	194
C.2.3	Modified.....	194
C.2.4	Removed.....	194
C.3	Data Definitions .....	194
C.3.1	New .....	194
C.3.2	Modified.....	194
C.3.3	Removed.....	195
C.4	Service Properties.....	195
C.4.1	New .....	195
C.4.2	Deprecated.....	195
C.4.3	Modified.....	195
C.4.4	Removed.....	195
C.5	Exceptions .....	196
C.5.1	New .....	196
C.5.2	Modified.....	196
C.5.3	Removed.....	196
C.6	Others .....	196
History	.....	197

iTeh STANDARD PREVIEW  
(standards.iteh.ai)

[SIST ES 201 915-4 V1.4.1:2005](https://standards.iteh.ai/catalog/standards/sist/3ff22eaf-dcfa-4397-9b38-06e77b7e70cb/sist-es-201-915-4-v1-4-1-2005)

<https://standards.iteh.ai/catalog/standards/sist/3ff22eaf-dcfa-4397-9b38-06e77b7e70cb/sist-es-201-915-4-v1-4-1-2005>

---

## 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 4 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.**

**A subset of the present document is in 3GPP TS 29.198-4 4.6.0 (Release 4).**

---

# 1 Scope

The present document is part 4 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 Call Control Service Capability Feature (SCF) aspects of the interface. All aspects of the Call Control 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).

**iTeh STANDARD PREVIEW**

---

## 2 References (standards.iteh.ai)

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

<https://standards.iteh.ai/catalog/standards/sist/3ff22eaf-dcfa-4397-9b38-66c557b1-201915-4-v1.4.1-2005>

ETSI ES 201 915-1: "Open Service Access; Application Programming Interface; Part 1: Overview (Parlay 3)".

---

## 3 Definitions and abbreviations

### 3.1 Definitions

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

### 3.2 Abbreviations

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

---

## 4 Call Control SCF

Two flavours of call control APIs have been included in 3GPP Release 4. These are the generic call control and the multi-party call control. The generic call control is the same API as was already present in the previous specification for 3GPP Release 99 (TS 129 198 V3.4.0 [28]) and is in principle able to satisfy the requirements on Call Control APIs for 3GPP Release 4.

However, the joint work between 3GPP CN5, ETSI SPAN12 and the Parlay Call Control Working group with collaboration from JAIN has been focussed on the Multi-party call control API. A number of improvements on call control functionality have been made and are reflected in this API. For this it was necessary to break the inheritance that previously existed between Generic and Multi-party call control.

The joint call control group has furthermore decided that the multi-party call control is to be considered as the future base call control family and the technical work will not be continued on Generic Call control. Errors or technical flaws will of course be corrected.

The following clauses describe each aspect of the Call Control 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 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 the present document.

## 4.1 Call Model Description

The adopted call model has the following objects.

- a call object. A call is a relation between a number of parties. The call object relates to the entire call view from the application. E.g., the entire call will be released when a release is called on the call. Note that different applications can have different views on the same physical call, e.g., one application for the originating side and another application for the terminating side. The applications will not be aware of each other, all 'communication' between the applications will be by means of network signalling. The API currently does not specify any feature interaction mechanisms;
- a call leg object. The leg object represents a logical association between a call and an address. The relationship includes at least the signalling relation with the party. The relation with the address is only made when the leg is routed. Before that the leg object is IDLE and not yet associated with the address;
- an address. The address logically represents a party in the call;
- a terminal. A terminal is the end-point of the signalling and/or media for a party. This object type is currently not addressed.

The call object is used to establish a relation between a number of parties by creating a leg for each party within the call.

Associated with the signalling relationship represented by the call leg, there may also be a bearer connection (e.g. in the traditional voice only networks) or a number (zero or more) of media channels (in multi-media networks).

A leg can be attached to the call or detached from the call. When the leg is attached, this means that media or bearer channels related to the legs are connected to the media or bearer channels of the other legs that are attached to the same call. I.e., only legs that are attached can 'speak' to each other. A leg can have a number of states, depending on the signalling received from or sent to the party associated with the leg. Usually there is a limit to the number of legs that are in being routed (i.e., the connection is being established) or connected to the call (i.e., the connection is established). Also, there usually is a limit to the number of legs that can be simultaneously attached to the same call.

Some networks distinguish between controlling and passive legs. By definition the call will be released when the controlling leg is released. All other legs are called passive legs. There can be at most one controlling leg per call. However, there is currently no way the application can influence whether a Leg is controlling or not.

There are two ways for an application to get the control of a call. The application can request to be notified of calls that meet certain criteria. When a call occurs in the network that meets these criteria, the application is notified and can control the call. Some legs will already be associated with the call in this case. Another way is to create a new call from the application.

## 4.2 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 The Service Interface Specifications

## 5.1 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.

### 5.1.1 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>`.

### 5.1.2 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.

### 5.1.3 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.

### 5.1.4 State Model

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

## 5.2 Base Interface

### 5.2.1 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

## 5.3 Service Interfaces

### 5.3.1 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'.

## 5.4 Generic Service Interface

### 5.4.1 Interface Class IpService

Inherits from: IpInterface

All service interfaces inherit from the following interface.

<<Interface>> IpService
SIST ES 201 915-4 V1.4.1:2005 <a href="https://standards.iteh.ai/catalog/standards/sis/3ff22caf-dcfa-4397-9b38-06e77b7e70cb/sist-es-201-915-4-v1-4-1-2005">https://standards.iteh.ai/catalog/standards/sis/3ff22caf-dcfa-4397-9b38-06e77b7e70cb/sist-es-201-915-4-v1-4-1-2005</a>
setCallback (appInterface : in IpInterfaceRef) : void setCallbackWithSessionID (appInterface : in IpInterfaceRef, sessionID : in TpSessionID) : void

#### 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.

#### Parameters

**appInterface : in IpInterfaceRef**

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

#### Raises

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