

**Open Service Access (OSA);
Application Programming Interface (API);
Part 5: User Interaction SCF
(Parlay 6)**



iTeh STANDARD PREVIEW
(standards.iteh.ai)

Full standard:
<https://standards.iteh.ai/catalog/standards/sist/a262801d-6748-4205-8d04-19cc897d64a8/etsi-es-204-915-5-v1.1.1-2008-05>



Reference

DES/TISPAN-01032-5-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

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™, PLUGTESTS™, UMTS™, TIPHON™, the TIPHON logo and the ETSI logo are Trade Marks of ETSI registered 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	7
Foreword.....	7
1 Scope	8
2 References	8
3 Definitions and abbreviations.....	8
3.1 Definitions	8
3.2 Abbreviations	8
4 Generic and Call User Interaction and Administration SCF	9
4.1 Generic and Call User Interaction SCF	9
4.2 Generic User Interaction Administration SCF	10
4.3 Generic User Interaction SCF Design Aspects.....	10
4.4 General requirements on support of methods.....	10
5 Sequence Diagrams	11
5.1 Generic and Call User Interaction Sequence Diagrams.....	11
5.1.1 Alarm Call	11
5.1.2 Call Barring 1	12
5.1.3 Network Controlled Notifications.....	14
5.1.4 Prepaid	15
5.1.5 Pre-Paid with Advice of Charge (AoC)	16
5.2 Generic User Interaction Administration Sequence Diagrams.....	19
5.2.1 Message Administration	19
6 Class Diagrams.....	20
6.1 Generic and Call User Interaction Class Diagrams	20
6.2 Generic User Interaction Administration Class Diagrams	21
7 The Service Interface Specifications	22
7.1 Interface Specification Format	22
7.1.1 Interface Class	22
7.1.2 Method descriptions.....	22
7.1.3 Parameter descriptions.....	22
7.1.4 State Model.....	22
7.2 Base Interface.....	22
7.2.1 Interface Class IpInterface	22
7.3 Service Interfaces	23
7.3.1 Overview	23
7.4 Generic Service Interface	23
7.4.1 Interface Class IpService	23
7.4.1.1 Method setCallback().....	23
7.4.1.2 Method setCallbackWithSessionID().....	23
8 Generic User Interaction Interface Classes	24
8.1 Generic and Call User Interaction Interface Classes	24
8.1.1 Interface Class IpUIManager.....	24
8.1.1.1 Method createUI().....	25
8.1.1.2 Method createUICall()	25
8.1.1.3 Method createNotification().....	25
8.1.1.4 Method destroyNotification().....	26
8.1.1.5 Method changeNotification()	26
8.1.1.6 Method getNotification()	27
8.1.1.7 Method enableNotifications()	27
8.1.1.8 Method disableNotifications().....	27
8.1.2 Interface Class IpAppUIManager.....	28
8.1.2.1 Method userInteractionAborted().....	28

8.1.2.2	Method userInteractionNotificationInterrupted()	28
8.1.2.3	Method userInteractionNotificationContinued()	29
8.1.2.4	Method reportEventNotification()	29
8.1.2.5	Method abortMultipleUserInteractions()	29
8.1.2.6	Method <<new>> reportEventReq()	30
8.1.3	Interface Class IpUI	30
8.1.3.1	Method sendInfoReq()	31
8.1.3.2	Method sendInfoAndCollectReq()	32
8.1.3.3	Method release()	33
8.1.3.4	Method setOriginatingAddress()	33
8.1.3.5	Method getOriginatingAddress()	33
8.1.3.6	Method <<new>> reportEventRes()	34
8.1.3.7	Method <<new>> reportEventErr()	34
8.1.4	Interface Class IpAppUI	34
8.1.4.1	Method sendInfoRes()	35
8.1.4.2	Method sendInfoErr()	35
8.1.4.3	Method sendInfoAndCollectRes()	36
8.1.4.4	Method sendInfoAndCollectErr()	36
8.1.4.5	Method userInteractionFaultDetected()	36
8.1.5	Interface Class IpUICall	37
8.1.5.1	Method recordMessageReq()	37
8.1.5.2	Method deleteMessageReq()	38
8.1.5.3	Method abortActionReq()	38
8.1.5.4	Method getMessageReq()	38
8.1.6	Interface Class IpAppUICall	39
8.1.6.1	Method recordMessageRes()	39
8.1.6.2	Method recordMessageErr()	40
8.1.6.3	Method deleteMessageRes()	40
8.1.6.4	Method deleteMessageErr()	40
8.1.6.5	Method abortActionRes()	40
8.1.6.6	Method abortActionErr()	41
8.1.6.7	Method getMessageRes()	41
8.1.6.8	Method getMessageErr()	41
8.2	Generic User Interaction Administration Interface Classes	41
8.2.1	Interface Class IpUIAdminManager	41
8.2.1.1	Method getMessageReq()	42
8.2.1.2	Method putMessageReq()	42
8.2.1.3	Method deleteMessageReq()	43
8.2.1.4	Method getMessageListReq()	43
8.2.2	Interface Class IpAppUIAdminManager	43
8.2.2.1	Method getMessageRes()	44
8.2.2.2	Method getMessageErr()	44
8.2.2.3	Method deleteMessageRes()	44
8.2.2.4	Method deleteMessageErr()	45
8.2.2.5	Method putMessageRes()	45
8.2.2.6	Method putMessageErr()	45
8.2.2.7	Method getMessageListRes()	45
8.2.2.8	Method getMessageListErr()	46
9	State Transition Diagrams	46
9.1	Generic and Call User Interaction State Transition Diagrams	46
9.1.1	State Transition Diagrams for IpUIManager	46
9.1.1.1	Active State	46
9.1.1.2	Notification Terminated State	47
9.1.2	State Transition Diagrams for IpUI	47
9.1.2.1	Active State	47
9.1.2.2	Release Pending State	47
9.1.2.3	Finished State	47
9.1.3	State Transition Diagrams for IpUICall	48
9.1.3.1	Active State	48
9.1.3.2	Release Pending State	48
9.1.3.3	Finished State	49

9.2	Generic User Interaction Administration State Transition Diagrams	49
9.2.1	State Transition Diagrams for IpUIAdminManager	49
9.2.1.1	Active State	49
10	Service Properties	49
10.1	User Interaction Service Properties	49
11	Data Definitions	50
11.1	TpUIFault	50
11.2	IpUI	50
11.3	IpUIRef	50
11.4	IpAppUI	50
11.5	IpAppUIRef	50
11.6	IpAppUIManager	51
11.7	IpAppUIManagerRef	51
11.8	TpUICallIdentifier	51
11.9	TpUICollectCriteria	51
11.10	TpUIError	52
11.11	TpUIEventCriteria	53
11.12	TpUIEventCriteriaResultSet	53
11.13	TpUIEventCriteriaResult	53
11.14	TpUIEventInfoDataType	53
11.15	TpUIIdentifier	53
11.16	TpUIIdentifierSet	53
11.17	TpUIInfo	54
11.18	TpUIInfoType	55
11.19	TpUIMessageCriteria	55
11.20	TpUIReport	55
11.21	TpUIResponseRequest	56
11.22	TpUITargetObjectType	56
11.23	TpUITargetObject	56
11.24	TpUIVariableInfo	57
11.25	TpUIVariableInfoSet	57
11.26	TpUIVariablePartType	57
11.27	TpUIEventNotificationInfo	57
11.28	TpUISynthesisInfoData	57
11.29	TpUISynthesisGender	58
11.30	TpUISynthesisAge	58
11.31	TpUISynthesisRate	58
11.32	TpUISynthesisRange	58
11.33	TpUIWordOverrideSet	58
11.34	TpUIWordOverride	59
11.35	TpUIPronounceType	59
11.36	TpUICollectMode	59
11.37	TpUIRecognitionCriteria	59
11.38	TpUIRecognitionSpeakerID	60
11.39	TpUIRecognitionPropertySet	60
11.40	TpUIRecognitionProperty	60
11.41	TpUIRecognitionGrammar	60
11.42	TpMessageIDList	61
12	Exception Classes	62
Annex A (normative):	OMG IDL Description of User Interaction SCF	63
Annex B (informative):	W3C WSDL Description of User Interaction SCF	64
Annex C (informative):	Java™ API Description of the User Interaction SCF	65
Annex D (informative):	Contents of 3GPP OSA R7 User Interaction	66
Annex E (informative):	Description of Generic User Interaction SCF for 3GPP2 cdma2000 networks	67

E.1	General Exceptions.....	67
E.2	Specific Exceptions	67
E.2.1	Clause 1: Scope	67
E.2.2	Clause 2: References	67
E.2.3	Clause 3: Definitions and abbreviations.....	67
E.2.4	Clause 4: Generic and Call User Interaction SCF	67
E.2.5	Clause 5: Class Diagrams.....	67
E.2.6	Clause 6: Class Diagrams.....	67
E.2.7	Clause 7: The Service Interface Specifications	68
E.2.8	Clause 8: Generic User Interaction Interface Classes Definitions	68
E.2.9	Clause 9: State Transition Diagrams	68
E.2.10	Clause 10: Service Properties.....	68
E.2.11	Clause 11: Data Definitions.....	68
E.2.12	Clause 12: Exception Classes.....	68
E.2.13	Annex A (normative): OMG IDL Description of Generic User Interaction SCF.....	68
E.2.14	Annex B (informative): W3C WSDL Description of Generic User Interaction SCF.....	68
E.2.15	Annex C (informative): Java™ API Description of Generic User Interaction SCF.....	68
Annex F (informative):	Record of changes	69
F.1	Interfaces	69
F.1.1	New	69
F.1.2	Deprecated.....	69
F.1.3	Removed.....	69
F.2	Methods.....	69
F.2.1	New	69
F.2.2	Deprecated.....	69
F.2.3	Modified.....	70
F.2.4	Removed.....	70
F.3	Data Definitions	70
F.3.1	New	70
F.3.2	Modified.....	70
F.3.3	Removed.....	70
F.4	Service Properties.....	70
F.4.1	New	70
F.4.2	Deprecated.....	70
F.4.3	Modified.....	71
F.4.4	Removed.....	71
F.5	Exceptions	71
F.5.1	New	71
F.5.2	Modified.....	71
F.5.3	Removed.....	71
F.6	Others	71
History		72

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), and is now submitted for the ETSI standards Membership Approval Procedure.

The present document is part 5 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 29.198-5 V7.1.0 (Release 7).

1 Scope

The present document is part 5 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 User Interaction (UI) Service Capability Feature (SCF) aspects of the interface. All aspects of the User Interaction 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.
- Reference to the Java™ API 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 Generic and Call User Interaction and Administration SCF

4.1 Generic and Call User Interaction SCF

The Generic User Interaction service capability feature is used by applications to interact with end users. It consists of three interfaces:

- 1) User Interaction Manager, containing management functions for User Interaction related issues.
- 2) Generic User Interaction, containing methods to interact with an end-user.
- 3) Call User Interaction, containing methods to interact with an end-user engaged in a call.

The Generic User Interaction service capability feature is described in terms of the methods in the Generic User Interaction interfaces.

The following table gives an overview of the Generic User Interaction methods and to which interfaces these methods belong.

Table 1: Overview of Generic User Interaction interfaces and their methods

User Interaction Manager	Generic User Interaction
createUI	sendInfoReq
createUICall	sendInfoRes
createNotification	sendInfoErr
destroyUINotification	sendInfoAndCollectReq
reportEventNotification	sendInfoAndCollectRes
userInteractionAborted	sendInfoAndCollectErr
userInteractionNotificationInterrupted	release
userInteractionNotificationContinued	userInteractionFaultDetected
changeNotification	setOriginatingAddress
getNotification	getOriginatingAddress
enableNotifications	reportEventRes
disableNotifications	reportEventErr
abortMultipleUserInteractions	
reportEventReq	

The following table gives an overview of the Call User Interaction methods and to which interfaces these methods belong.

Table 2: Overview of Call User Interaction interfaces and their methods

User Interaction Manager	Call User Interaction
As defined for the Generic User Interaction SCF	Inherits from Generic User Interaction and adds:
	recordMessageReq
	recordMessageRes
	recordMessageErr
	deleteMessageReq
	deleteMessageRes
	deleteMessageErr
	abortActionReq
	abortActionRes
	abortActionErr
	getMessageReq
	getMessageRes
	getMessageErr

The IpUI Interface provides functions to send information to, or gather information from the user, i.e. this interface allows applications to send SMS and USSD messages. An application can use this interface independently of other SCFs. The IpUICall Interface provides functions to send information to, or gather information from the user (or call party) attached to a call.

4.2 Generic User Interaction Administration SCF

The Generic User Interaction Administration service capability feature is used by application to interact with the service to manage the user announcement and recorded messages. It consists of one interface:

- 1) User Interaction Administration Manager, containing message management functions for User Interaction.

Table 3: Overview of Generic User Interaction Administration interfaces and their methods

User Interaction Administration Manager
getMessageReq
putMessageReq
deleteMessageReq
getMessageListReq

4.3 Generic User Interaction SCF Design Aspects

The following clauses describe each aspect of the Generic User Interaction and Generic User Interaction Administration Service Capability Features (SCF).

The order is as follows:

- The Sequence diagrams give the reader a practical idea of how each of the SCFs 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. This clause also includes Call User interaction.
- 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.

4.4 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.

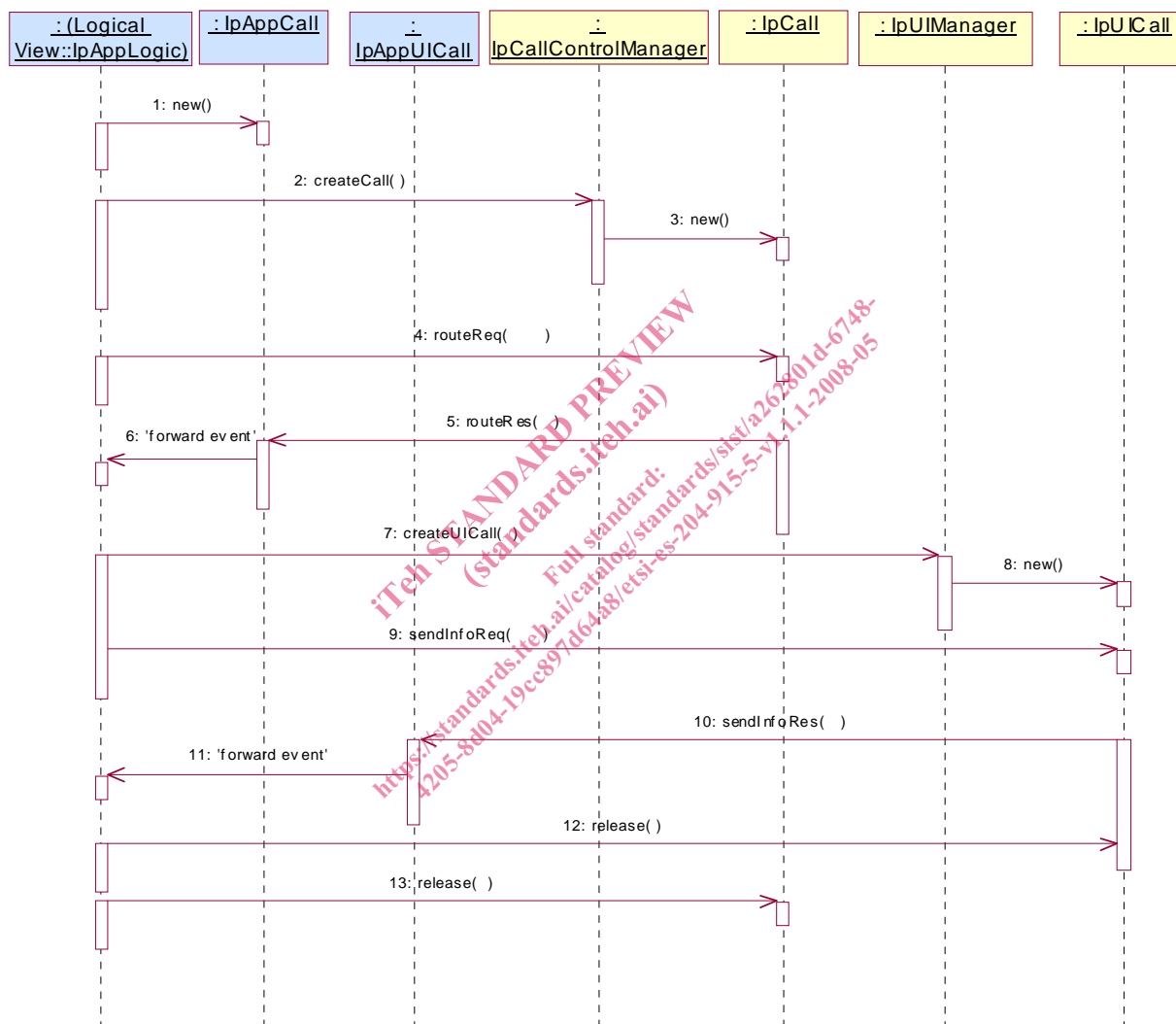
Where a method is not supported by an implementation of an Application interface, a call to that method shall be possible, and no exception shall be returned.

5 Sequence Diagrams

5.1 Generic and Call User Interaction Sequence Diagrams

5.1.1 Alarm Call

The following sequence diagram shows a 'reminder message', in the form of an alarm, being delivered to a customer as a result of a trigger from an application. Typically, the application would be set to trigger at a certain time, however, the application could also trigger on events.

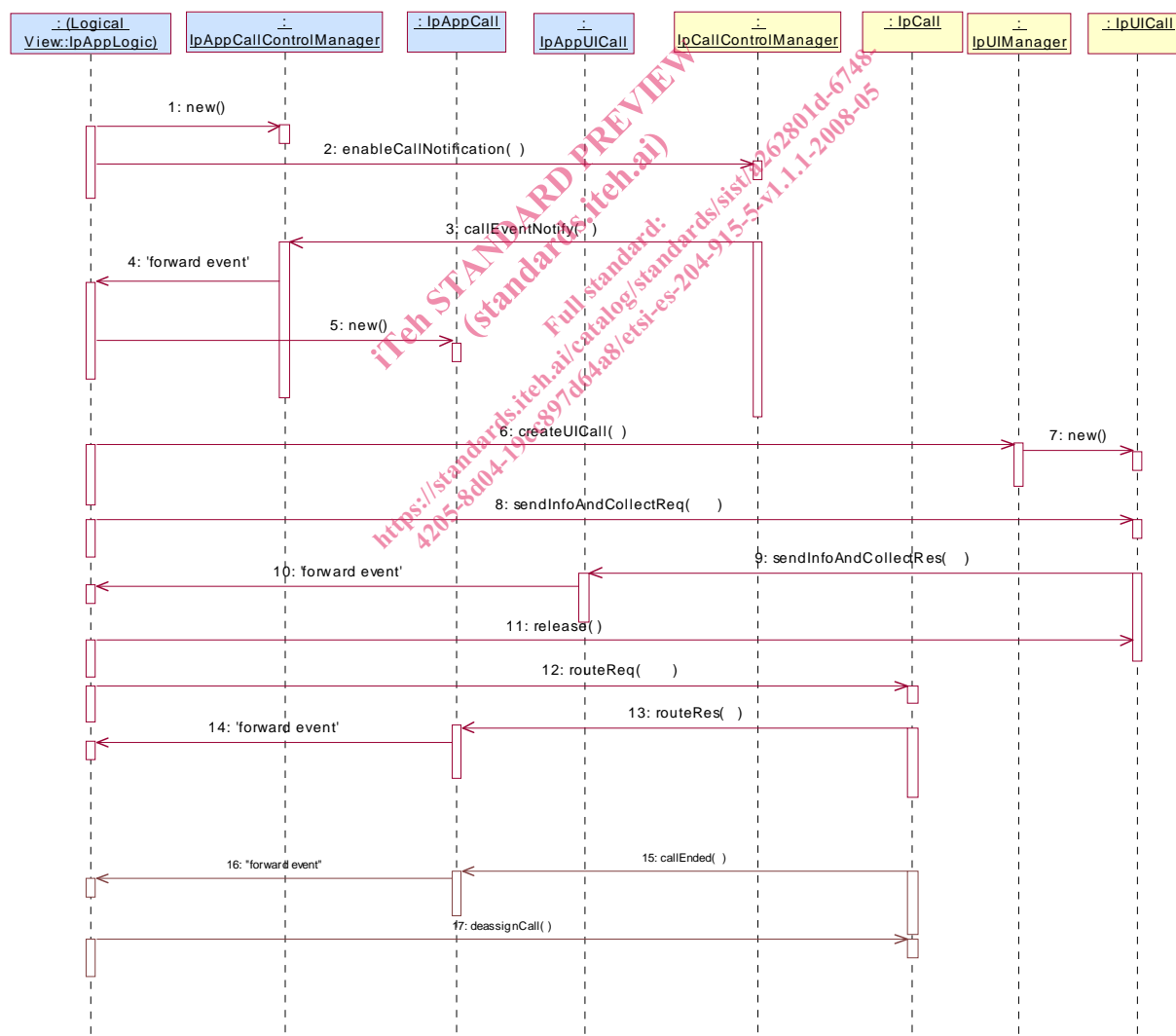


- 1: This message is used to create an object implementing the IpAppCall interface.
- 2: This message requests the object implementing the IpCallControlManager interface to create an object implementing the IpCall interface.
- 3: Assuming that the criteria for creating an object implementing the IpCall interface (e.g. load control values not exceeded) are met it is created.
- 4: This message instructs the object implementing the IpCall interface to route the call to the customer destined to receive the 'reminder message'.
- 5: This message passes the result of the call being answered to its callback object.
- 6: This message is used to forward the previous message to the IpAppLogic.

- 7: The application requests a new UICall object that is associated with the call object.
- 8: Assuming all criteria are met, a new UICall object is created by the service.
- 9: This message instructs the object implementing the IpUICall interface to send the alarm to the customer's call.
- 10: When the announcement ends this is reported to the call back interface.
- 11: The event is forwarded to the application logic.
- 12: The application releases the UICall object, since no further announcements are required. Alternatively, the application could have indicated P_FINAL_REQUEST in the sendInfoReq in which case the UICall object would have been implicitly released after the announcement was played.
- 13: The application releases the call and all associated parties.

5.1.2 Call Barring 1

The following sequence diagram shows a call barring service, initiated as a result of a prearranged event being received by the call control service. Before the call is routed to the destination number, the calling party is asked for a PIN code. The code is accepted and the call is routed to the original called party.



- 1: This message is used by the application to create an object implementing the IpAppCallControlManager interface.

- 2: This message is sent by the application to enable notifications on new call events. As this sequence diagram depicts a call barring service, it is likely that all new call events destined for a particular address or address range prompted for a password before the call is allowed to progress. When a new call, that matches the event criteria set, arrives, a message (not shown) is directed to the object implementing the IpCallControlManager. Assuming that the criteria for creating an object implementing the IpCall interface (e.g. load control values not exceeded) are met, other messages (not shown) are used to create the call and associated call leg object.
- 3: This message is used to pass the new call event to the object implementing the IpAppCallControlManager interface.
- 4: This message is used to forward the previous message to the IpAppLogic.
- 5: This message is used by the application to create an object implementing the IpAppCall interface. The reference to this object is passed back to the object implementing the IpCallControlManager using the return parameter of the callEventNotify.
- 6: This message is used to create a new UICall object. The reference to the call object is given when creating the UICall.
- 7: Provided all the criteria are fulfilled, a new UICall object is created.
- 8: The call barring service dialogue is invoked.
- 9: The result of the dialogue, which in this case is the PIN code, is returned to its callback object.
- 10: This message is used to forward the previous message to the IpAppLogic.
- 11: This message releases the UICall object.
- 12: Assuming the correct PIN is entered, the call is forward routed to the destination party.
- 13: This message passes the result of the call being answered to its callback object.
- 14: This message is used to forward the previous message to the IpAppLogic.
- 15: When the call is terminated in the network, the application will receive a notification. This notification will always be received when the call is terminated by the network in a normal way, the application does not have to request this event explicitly.
- 16: The event is forwarded to the application.
- 17: The application must free the call-related resources in the gateway by calling deassignCall.