



SLOVENSKI STANDARD

SIST ES 202 915-5 V1.2.1:2005

01-januar-2005

**Odpri dostop do storitve (OSA) – Vmesnik za aplikacijsko programiranje (API) – 5.
del: Medsebojno vplivanje uporabnikov SCF**

Open Service Access (OSA); Application Programming Interface (API); Part 5: User
Interaction SCF (Parlay 4)

iTeh STANDARD PREVIEW
(standards.iteh.ai)

Ta slovenski standard je istoveten z: **ES 202 915-5 Version 1.2.1**

SIST ES 202 915-5 V1.2.1:2005
<https://standards.iteh.ai/catalog/standards/sist/27c59e2c-f1f8-4c44-ba12-e813226b4bac/sist-es-202-915-5-v1-2-1-2005>

ICS:

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

SIST ES 202 915-5 V1.2.1:2005 **en**

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[SIST ES 202 915-5 V1.2.1:2005](https://standards.iteh.ai/catalog/standards/sist/27c59e2c-fff8-4c44-ba12-e813226b4bac/sist-es-202-915-5-v1-2-1-2005)

<https://standards.iteh.ai/catalog/standards/sist/27c59e2c-fff8-4c44-ba12-e813226b4bac/sist-es-202-915-5-v1-2-1-2005>

ETSI ES 202 915-5 V1.2.1 (2003-08)

ETSI Standard

Open Service Access (OSA); Application Programming Interface (API); Part 5: User Interaction SCF (Parlay 4)



iTeh STANDARD PREVIEW (standards.iteh.ai)

[SIST ES 202 915-5 V1.2.1:2005](https://standards.iteh.ai/catalog/standards/sist/27c59e2c-fff8-4c44-ba12-e813226b4bac/sist-es-202-915-5-v1-2-1-2005)

<https://standards.iteh.ai/catalog/standards/sist/27c59e2c-fff8-4c44-ba12-e813226b4bac/sist-es-202-915-5-v1-2-1-2005>



Reference

RES/SPAN-120096-5

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

iTeh STANDARD PREVIEW
(standards.iteh.ai)

SIST ES 202 915-5 V1.2.1:2005

<https://standards.iteh.ai/catalog/standards/sist/27c59e2c-fff8-4c44-ba12-e813226b417c/sist-202-915-5-v1-2-1-2005>

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

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	6
Foreword.....	6
1 Scope	7
2 References	7
3 Definitions and abbreviations.....	7
3.1 Definitions	7
3.2 Abbreviations	7
4 Generic and Call User Interaction SCF.....	8
4.1 General requirements on support of methods.....	9
5 Sequence Diagrams	9
5.1 Alarm Call	9
5.2 Call Barring 1	11
5.3 Network Controlled Notifications	12
5.4 Prepaid.....	13
5.5 Pre-Paid with Advice of Charge (AoC).....	15
6 Class Diagrams.....	18
7 The Service Interface Specifications	19
7.1 Interface Specification Format	19
7.1.1 Interface Class	19
7.1.2 Method descriptions.....	19
7.1.3 Parameter descriptions	19
7.1.4 State Model	19
7.2 Base Interface	19
7.2.1 Interface Class IpInterface	19
7.3 Service Interfaces	20
7.3.1 Overview	20
7.4 Generic Service Interface	20
7.4.1 Interface Class IpService	20
7.4.1.1 Method setCallback()	20
7.4.1.2 Method setCallbackWithSessionID().....	20
8 Generic User Interaction Interface Classes	21
8.1 Interface Class IpUIManager	21
8.1.1 Method createUI().....	21
8.1.2 Method createUICall()	22
8.1.3 Method createNotification().....	22
8.1.4 Method destroyNotification()	23
8.1.5 Method changeNotification().....	23
8.1.6 Method getNotification()	24
8.1.7 Method <<new>> enableNotifications().....	24
8.1.8 Method <<new>> disableNotifications().....	25
8.2 Interface Class IpAppUIManager.....	25
8.2.1 Method userInteractionAborted()	25
8.2.2 Method <<deprecated>> reportNotification()	25
8.2.3 Method userInteractionNotificationInterrupted().....	26
8.2.4 Method userInteractionNotificationContinued().....	26
8.2.5 Method <<new>> reportEventNotification().....	26
8.3 Interface Class IpUI.....	27
8.3.1 Method sendInfoReq().....	27
8.3.2 Method sendInfoAndCollectReq().....	28
8.3.3 Method release()	29
8.4 Interface Class IpAppUI.....	29

8.4.1	Method sendInfoRes()	30
8.4.2	Method sendInfoErr()	30
8.4.3	Method sendInfoAndCollectRes()	30
8.4.4	Method sendInfoAndCollectErr()	31
8.4.5	Method userInteractionFaultDetected()	31
8.5	Interface Class IpUICall	31
8.5.1	Method recordMessageReq()	32
8.5.2	Method deleteMessageReq()	32
8.5.3	Method abortActionReq()	33
8.6	Interface Class IpAppUICall	33
8.6.1	Method recordMessageRes()	33
8.6.2	Method recordMessageErr()	34
8.6.3	Method deleteMessageRes()	34
8.6.4	Method deleteMessageErr()	34
8.6.5	Method abortActionRes()	35
8.6.6	Method abortActionErr()	35
9	State Transition Diagrams	36
9.1	State Transition Diagrams for IpUIManager	36
9.1.1	Active State	36
9.1.2	Notification Terminated State	36
9.2	State Transition Diagrams for IpUI	37
9.2.1	Active State	37
9.2.2	Release Pending State	37
9.2.3	Finished State	37
9.3	State Transition Diagrams for IpUICall	38
9.3.1	Active State	38
9.3.2	Release Pending State	38
9.3.3	Finished State	39
10	Service Properties	39
10.1	User Interaction Service Properties	39
11	Data Definitions	39
11.1	TpUIFault	39
11.2	IpUI	39
11.3	IpUIRef	40
11.4	IpAppUI	40
11.5	IpAppUIRef	40
11.6	IpAppUIManager	40
11.7	IpAppUIManagerRef	40
11.8	TpUICallIdentifier	40
11.9	TpUICollectCriteria	40
11.10	TpUIError	41
11.11	TpUIEventCriteria	42
11.12	TpUIEventCriteriaResultSet	42
11.13	TpUIEventCriteriaResult	42
11.14	TpUIEventInfo	42
11.15	TpUIEventInfoDataType	42
11.16	TpUIIdentifier	43
11.17	TpUIInfo	43
11.18	TpUIInfoType	43
11.19	TpUIMessageCriteria	43
11.20	TpUIReport	44
11.21	TpUIResponseRequest	44
11.22	TpUITargetObjectType	44
11.23	TpUITargetObject	45
11.24	TpUIVariableInfo	45
11.25	TpUIVariableInfoSet	45
11.26	TpUIVariablePartType	45
11.27	TpUIEventNotificationInfo	46
12	Exception Classes	46

STANDARD PREVIEW
(standards.iteh.ai)

SIST ES 202 915-5 V1.2.1:2005

[https://standards.iteh.ai/catalog/standards/sist/27c59e2c-ff8-4c44-ba12-](https://standards.iteh.ai/catalog/standards/sist/27c59e2c-ff8-4c44-ba12-e813226b4bac/sist-es-202-915-5-v1-2-1-2005)

[e813226b4bac/sist-es-202-915-5-v1-2-1-2005](https://standards.iteh.ai/catalog/standards/sist/27c59e2c-ff8-4c44-ba12-e813226b4bac/sist-es-202-915-5-v1-2-1-2005)

Annex A (normative):	OMG IDL Description of User Interaction SCF	47
Annex B (informative):	W3C WSDL Description of User Interaction SCF	48
Annex C (informative):	Java API Description of User Interaction SCF	49
Annex D (informative):	Contents of 3GPP OSA R4 User Interaction.....	50
Annex E (informative):	Record of changes	51
E.1	Interfaces	51
E.1.1	New	51
E.1.2	Deprecated.....	51
E.1.3	Removed.....	51
E.2	Methods	51
E.2.1	New	51
E.2.2	Deprecated.....	52
E.2.3	Modified.....	52
E.2.4	Removed.....	52
E.3	Data Definitions	52
E.3.1	New	52
E.3.2	Modified.....	52
E.3.3	Removed.....	53
E.4	Service Properties.....	53
E.4.1	New	53
E.4.2	Deprecated.....	53
E.4.3	Modified.....	53
E.4.4	Removed.....	53
E.5	Exceptions	54
E.5.1	New	54
E.5.2	Modified.....	54
E.5.3	Removed.....	54
E.6	Others	54
History		55

iTech STANDARD PREVIEW
(standards.itech.ai)

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 5 of a multi-part deliverable covering Open Service Access (OSA); Application Programming Interface (API), as identified below. The API specification (ES 202 915) 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";** [SIST ES 202 915-5 V1.2.1:2005](https://standards.iteh.ai/catalog/standards/sist/27c59e2c-fff8-4c44-ba12-e813226b4bac/sist-es-202-915-5-v1-2-1-2005)
- Part 6: "Mobility SCF"; <https://standards.iteh.ai/catalog/standards/sist/27c59e2c-fff8-4c44-ba12-e813226b4bac/sist-es-202-915-5-v1-2-1-2005>
- 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".

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 4.1 set of specifications.

The present document is equivalent to 3GPP TS 29.198-5 V5.2.0 (Release 5).

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 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). **(standards.iteh.ai)**

2 References

[SIST ES 202 915-5 V1.2.1:2005](https://standards.iteh.ai/catalog/standards/sist/27c59e2c-fff8-4c44-ba12-e813226b4bac/sist-es-202-915-5-v1-2-1-2005)

standards.iteh.ai/catalog/standards/sist/27c59e2c-fff8-4c44-ba12-e813226b4bac/sist-es-202-915-5-v1-2-1-2005

The references listed in clause 2 of ES 202 915-1 contain provisions which, through reference in this text, constitute provisions of the present document.

ETSI ES 202 915-1: "Open Service Access (OSA); Application Programming Interface (API); Part 1: Overview (Parlay 4)".

ETSI ES 202 915-2: "Open Service Access (OSA); Application Programming Interface (API); Part 2: Common Data Definitions (Parlay 4)".

ETSI ES 202 915-4-2: "Open Service Access (OSA); Application Programming Interface (API); Part 4: Call Control; Sub-part 2: Generic Call Control SCF (Parlay 4)".

ETSI ES 202 915-4-3: "Open Service Access (OSA); Application Programming Interface (API); Part 4: Call Control; Sub-part 3: Multi-Party Call Control SCF (Parlay 4)".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ES 202 915-1 apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations defined in ES 202 915-1 apply.

4 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
reportNotification	sendInfoAndCollectRes
userInteractionAborted	sendInfoAndCollectErr
userInteractionNotificationInterrupted	release
userInteractionNotificationContinued	UserInteractionFaultDetected
changeNotification	
getNotification	
enableNotifications	
disableNotifications	

SIST ES 202 915-5 V1.2.1:2005

<https://standards.iteh.ai/catalog/standards/sist/27c59e2c-fff8-4c44-ba12-c013220746ac/sist-es-202-915-5-v1-2-1-2005>

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

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.

The following clauses describe each aspect of the Generic User Interaction Service Capability Feature (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 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. 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 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 ES 202 915-2.

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.

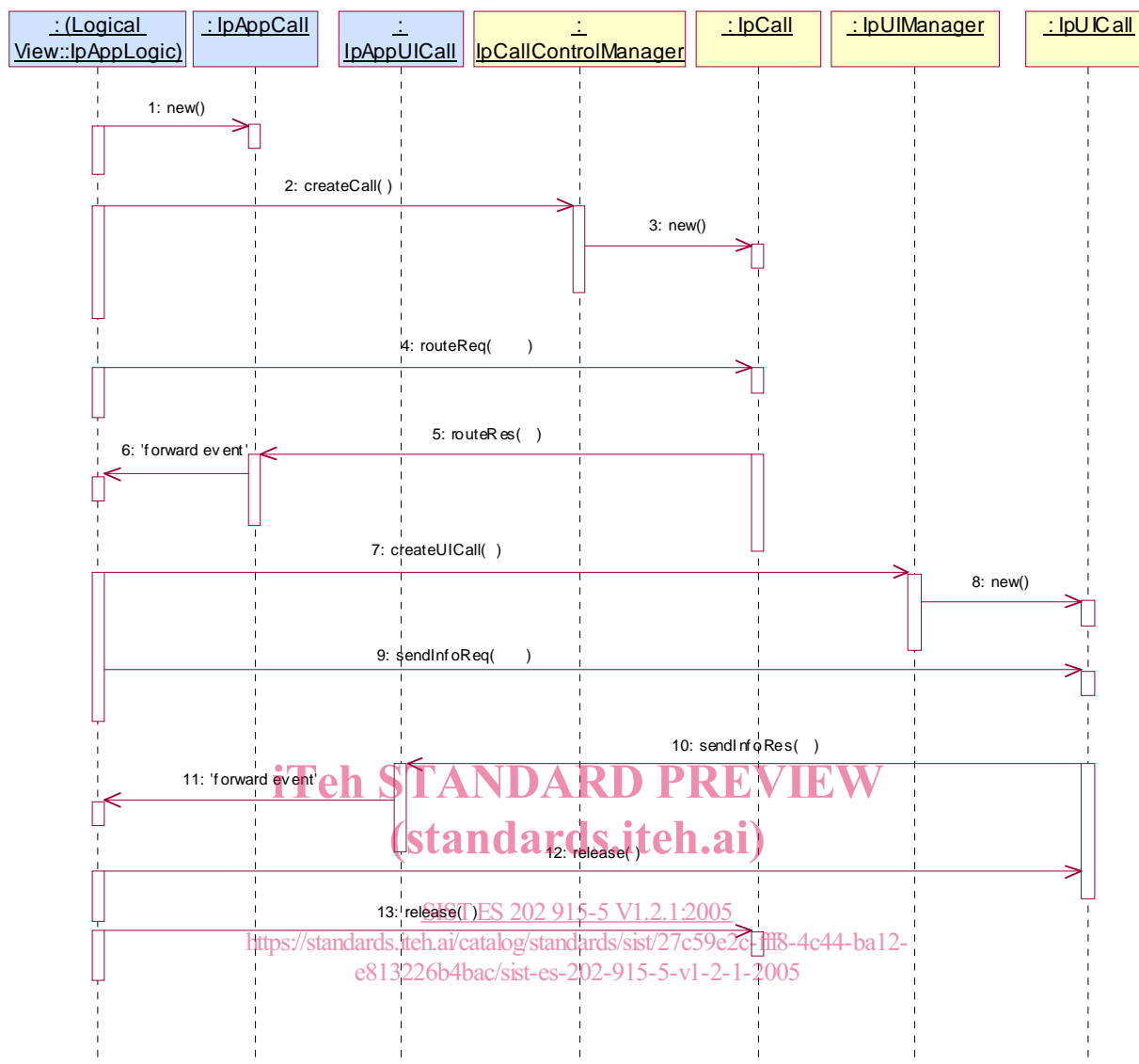
(standards.iteh.ai)

5 Sequence Diagrams

<https://standards.iteh.ai/catalog/standards/sist/27c59e2c-fff8-4c44-ba12-e813226b4bac/sist-es-202-915-5-v1-2-1-2005>

5.1 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.



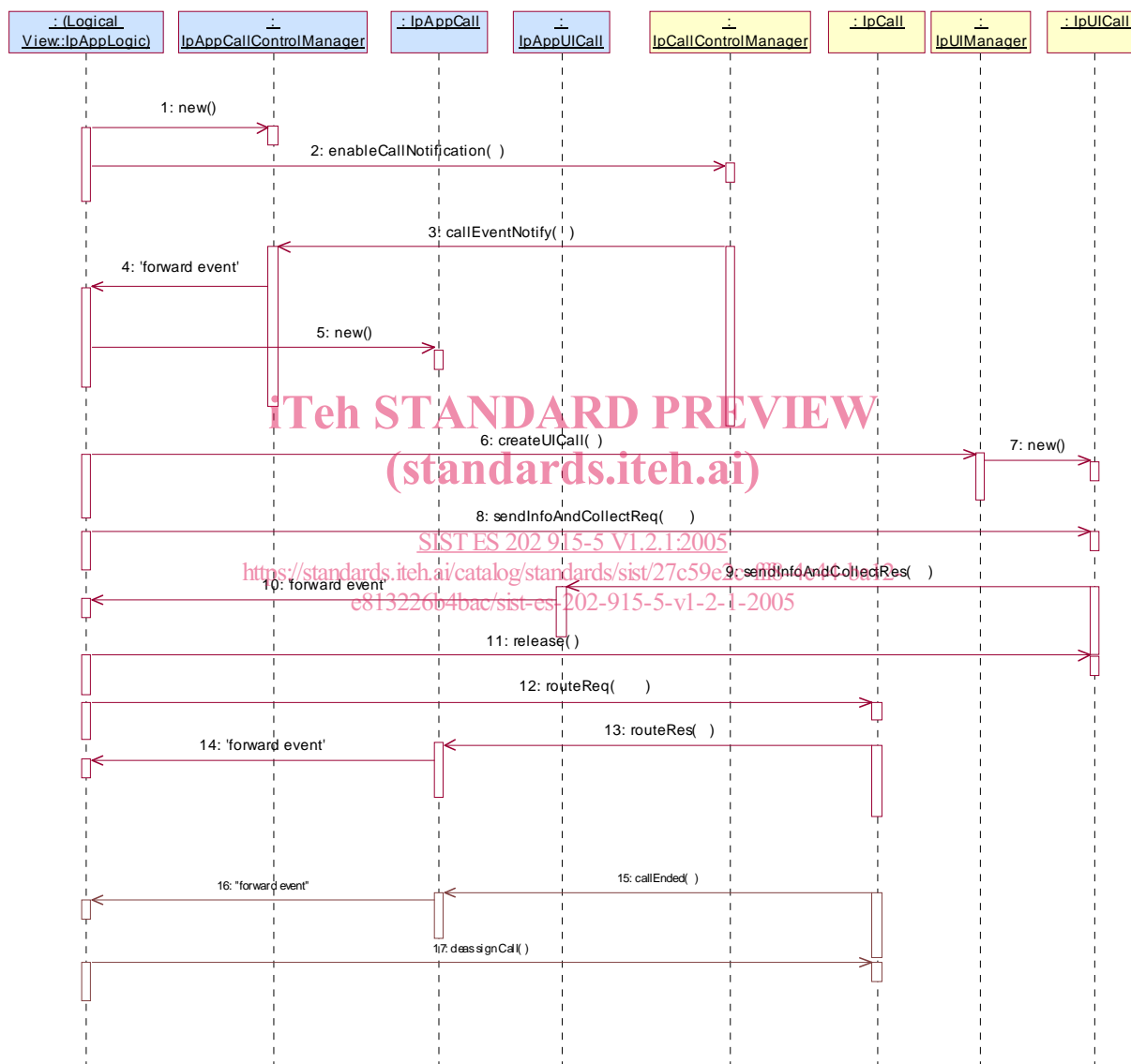
- 1: This message is used to create an object implementing the IpAppCall interface.
- 2: This message requests the object implementing the IpCallControlManager interface to create an object implementing the IpCall interface.
- 3: Assuming that the criteria for creating an object implementing the IpCall 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.

5.2 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.

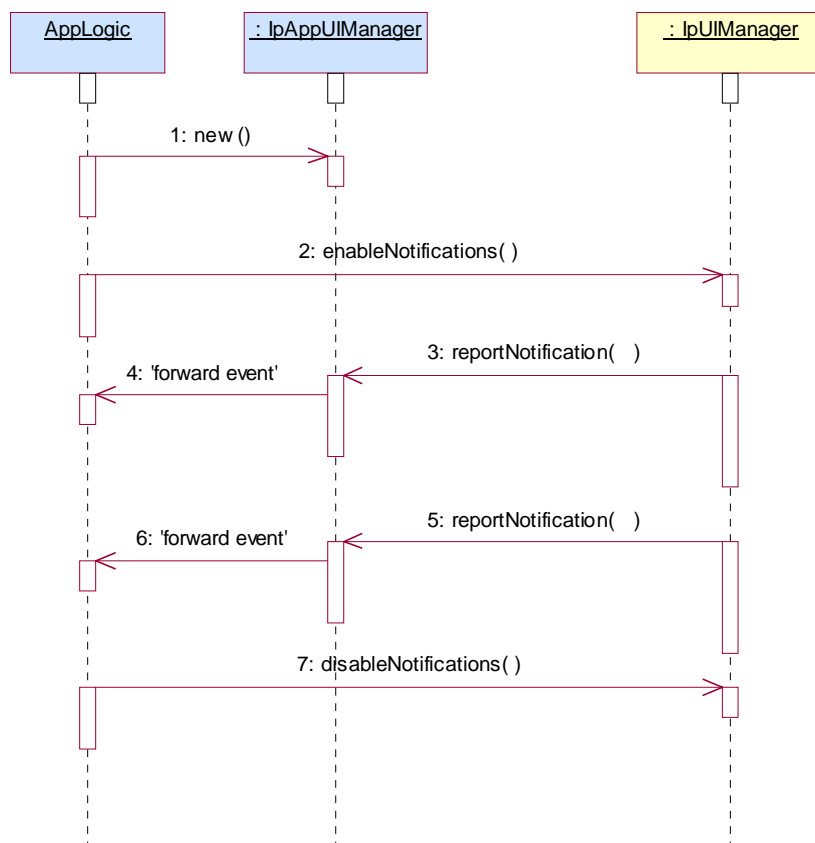
16: The event is forwarded to the application.

17: The application must free the call related resources in the gate way by calling deassignCall.

ITh STANDARD PREVIEW
(standards.iteh.ai)

5.3 Network Controlled Notifications

The following sequence diagram shows how an application can receive notifications that have not been created by the application, but are provisioned from within the network.



- 1: The application is started. The application creates a new IpAppUIManager to handle callbacks.
- 2: The enableNotifications method is invoked on the IpUIManager interface to indicate that the application is ready to receive notifications that are created in the network. For illustrative purposes we assume notifications of type "B" are created in the network.
- 3: When a network created trigger occurs the application is notified on the callback interface.
- 4: The event is forwarded to the application.
- 5: When a network created trigger occurs the application is notified on the callback interface.
- 6: The event is forwarded to the application.
- 7: When the application does not want to receive notifications created in the network anymore, it invokes disableNotifications on the IpMultiPartyCallControlManager interface. From now on the gateway will not send any notifications to the application that are created in the network.

5.4 Prepaid

This sequence shows a Pre-paid application. The subscriber is using a pre-paid card or credit card to pay for the call. The application each time allows a certain timeslice for the call. After the timeslice, a new timeslice can be started or the application can terminate the call. In the following sequence the end-user will received an announcement before his final timeslice.

iTeh STANDARD PREVIEW (standards.iteh.ai)

[SIST ES 202 915-5 V1.2.1:2005](https://standards.iteh.ai/catalog/standards/sist/27c59e2c-fff8-4c44-ba12-e813226b4bac/sist-es-202-915-5-v1-2-1-2005)

<https://standards.iteh.ai/catalog/standards/sist/27c59e2c-fff8-4c44-ba12-e813226b4bac/sist-es-202-915-5-v1-2-1-2005>