

# ETSI ES 204 915-9 V1.1.1 (2008-05)

ETSI Standard

## Open Service Access (OSA); Application Programming Interface (API); Part 9: Generic Messaging SCF (Parlay 6)



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

Full standard:  
<https://standards.iteh.ai/catalog/standards/sist/2ff27b04-1d02-4348-911a-4062c7daa6aa/etsi-es-204-915-9-v1.1.1-2008-05>



---

Reference

DES/TISPAN-01032-9-OSA

---

Keywords

---

API, IDL, OSA, UML

**ETSI**

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

---

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

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

---

**Important notice**

---

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

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

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

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

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

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

---

**Copyright Notification**

---

No part may be reproduced except as authorized by written permission.  
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2008.

© The Parlay Group 2008.

All rights reserved.

**DECT™**, **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 .....	6
Foreword.....	6
1 Scope .....	7
2 References .....	7
3 Definitions and abbreviations.....	7
3.1 Definitions .....	7
3.2 Abbreviations .....	7
4 Generic Messaging SCF.....	7
5 Sequence Diagrams .....	8
5.1 Prepare Mailbox .....	8
5.2 Open Mailbox.....	9
5.3 Get Message .....	9
5.4 Get Folder Information.....	10
5.5 Close Mailbox .....	11
6 Class Diagrams.....	11
7 The Service Interface Specifications .....	12
7.1 Interface Specification Format .....	12
7.1.1 Interface Class .....	12
7.1.2 Method descriptions.....	12
7.1.3 Parameter descriptions.....	13
7.1.4 State Model.....	13
7.2 Base Interface.....	13
7.2.1 Interface Class IpInterface .....	13
7.3 Service Interfaces .....	13
7.3.1 Overview .....	13
7.4 Generic Service Interface .....	13
7.4.1 Interface Class IpService .....	13
7.4.1.1 Method setCallback().....	14
7.4.1.2 Method setCallbackWithSessionID().....	14
8 Generic Messaging Interface Classes.....	14
8.1 Interface Class IpMessagingManager .....	15
8.1.1 Method openMailbox().....	15
8.1.2 Method enableMessagingNotification() .....	16
8.1.3 Method disableMessagingNotification().....	16
8.2 Interface Class IpAppMessagingManager.....	16
8.2.1 Method mailboxTerminated().....	17
8.2.2 Method mailboxFaultDetected().....	17
8.2.3 Method messagingEventNotify().....	17
8.2.4 Method messagingNotificationTerminated().....	18
8.3 Interface Class IpMailbox .....	18
8.3.1 Method close() .....	18
8.3.2 Method lock() .....	19
8.3.3 Method unlock() .....	19
8.3.4 Method getInfoAmount().....	19
8.3.5 Method getInfoProperties().....	20
8.3.6 Method setInfoProperties() .....	20
8.3.7 Method openFolder().....	21
8.3.8 Method createFolder() .....	21
8.3.9 Method remove() .....	21
8.4 Interface Class IpMailboxFolder.....	22
8.4.1 Method getInfoAmount().....	22

8.4.2	Method getInfoProperties()	23
8.4.3	Method setInfoProperties()	23
8.4.4	Method putMessage()	24
8.4.5	Method getMessage()	24
8.4.6	Method close()	25
8.4.7	Method remove()	25
8.5	Interface Class IpMessage	25
8.5.1	Method getInfoAmount()	26
8.5.2	Method getInfoProperties()	26
8.5.3	Method setInfoProperties()	27
8.5.4	Method remove()	28
8.5.5	Method getContent()	28
9	State Transition Diagrams	28
10	Data Definitions	29
10.1	Event notification Definitions	29
10.1.1	TpMessagingEventName	29
10.1.2	TpMessagingEventCriteria	29
10.1.3	TpGMSNewMessageArrivedCriteria	29
10.1.4	TpMessagingEventInfo	29
10.1.5	TpGMSNewMessageArrivedInfo	30
10.2	Generic Messaging Data Definitions	30
10.2.1	IpMessagingManager	30
10.2.2	IpMessagingManagerRef	30
10.2.3	IpAppMessagingManager	30
10.2.4	IpAppMessagingManagerRef	30
10.2.5	IpMailbox	30
10.2.6	IpMailboxRef	30
10.2.7	IpMailboxFolder	30
10.2.8	IpMailboxFolderRef	30
10.2.9	IpMessage	30
10.2.10	IpMessageRef	30
10.2.11	TpFolderInfoProperty	31
10.2.12	TpFolderInfoPropertyName	31
10.2.13	TpFolderInfoPropertySet	31
10.2.14	TpMailboxFolderIdentifier	31
10.2.15	TpMailboxIdentifier	31
10.2.16	TpMailboxInfoProperty	32
10.2.17	TpMailboxInfoPropertyName	32
10.2.18	TpMailboxInfoPropertySet	32
10.2.19	TpMessage	32
10.2.20	TpMessageFormat	32
10.2.21	TpMessageInfoProperty	32
10.2.22	TpMessageInfoPropertyName	33
10.2.23	TpMessageInfoPropertySet	33
10.2.24	TpMessagePriority	33
10.2.25	TpMessageStatus	34
10.2.26	TpMessagingFault	34
11	Exception Classes	34
<b>Annex A (normative):</b>	<b>OMG IDL Description of Generic Messaging SCF</b>	<b>35</b>
<b>Annex B (informative):</b>	<b>W3C WSDL Description of Generic Messaging SCF</b>	<b>36</b>
<b>Annex C (informative):</b>	<b>Java™ API Description of the Generic Messaging SCF</b>	<b>37</b>
<b>Annex D (informative):</b>	<b>Record of changes</b>	<b>38</b>
D.1	Interfaces	38
D.1.1	New	38
D.1.1	Deprecated	38
D.1.3	Removed	38

D.2	Methods	38
D.2.1	New	38
D.2.2	Deprecated	38
D.2.3	Modified	39
D.2.4	Removed	39
D.3	Data Definitions	39
D.3.1	New	39
D.3.2	Modified	39
D.3.3	Removed	39
D.4	Service Properties	39
D.4.1	New	39
D.4.2	Deprecated	40
D.4.3	Modified	40
D.4.4	Removed	40
D.5	Exceptions	40
D.5.1	New	40
D.5.2	Modified	40
D.5.3	Removed	40
D.6	Others	40
	History	41

**iTeh STANDARD PREVIEW**  
 (standards.iteh.ai)  
 Full standard:  
<https://standards.iteh.ai/catalog/standards/sist/2ff27b04-1d02-4348-911a-4062c7daa6aa/etsi-es-204-915-9-v1.1.1-2008-05>

---

## Intellectual Property Rights

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

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

---

## Foreword

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

The present document is part 9 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.**

---

# 1 Scope

The present document is part 9 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 Generic Messaging Service Capability Feature (SCF) aspects of the interface. All aspects of the Generic Messaging SCF are defined here, these being:

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

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

---

# 2 References

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

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

---

# 3 Definitions and abbreviations

## 3.1 Definitions

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

## 3.2 Abbreviations

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

---

# 4 Generic Messaging SCF

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

The order is as follows:

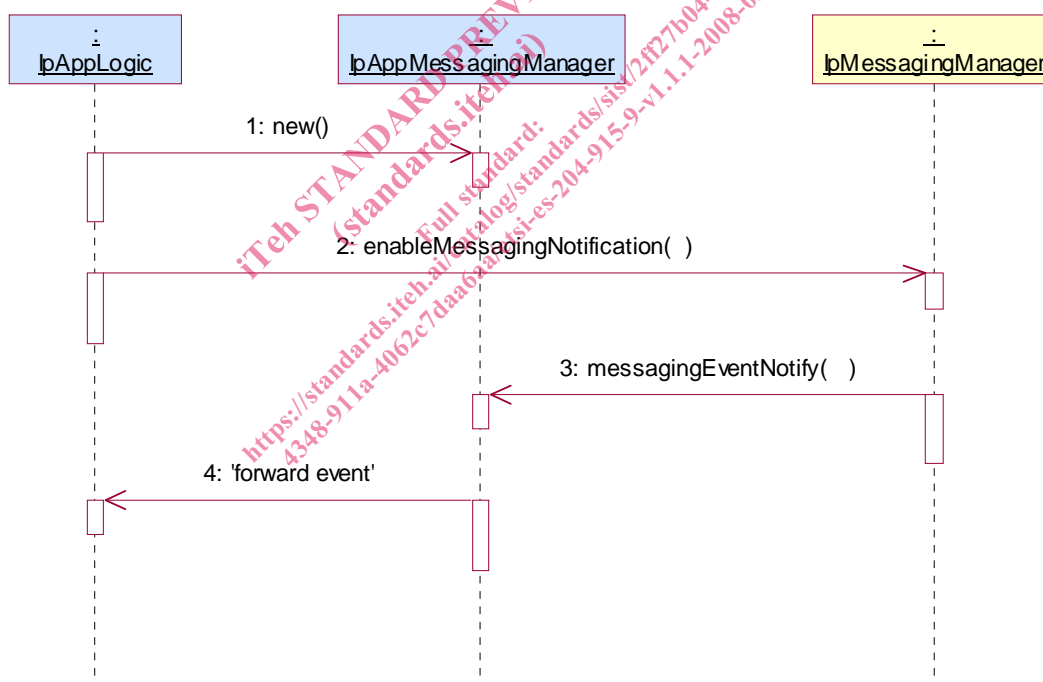
- The Sequence diagrams give the reader a practical idea of how each of the SCF is implemented.
- The Class relationships clause shows how each of the interfaces applicable to the SCF, relate to one another.

- The Interface specification clause describes in detail each of the interfaces shown within the Class diagram part.
- The State Transition Diagrams (STD) show the transition between states in the SCF. The states and transitions are well-defined; either methods specified in the Interface specification or events occurring in the underlying networks cause state transitions.
- The Data Definitions clause shows a detailed expansion of each of the data types associated with the methods within the classes. Note that some data types are used in other methods and classes and are therefore defined within the Common Data types part ES 204 915-2.

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.

## 5 Sequence Diagrams

### 5.1 Prepare Mailbox



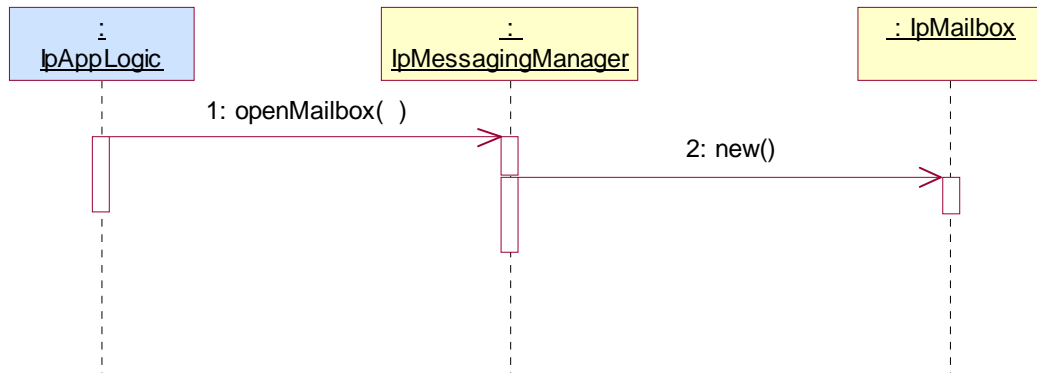
- 1: This message is used by the application to create an object implementing the IpAppMessagingManager interface.
- 2: This message is used to enable the notification mechanism so that events can be sent to the application.

When new mail, that matches the event criteria set in message 2, arrives a message indicating the presence of new mail (not shown) is directed to the object implementing the IpMessagingManager.

- 3: This message is used to pass the new mail event to the object implementing the IpAppMessagingManager interface.
- 4: This message is used to forward message 3 to the IpAppLogic.

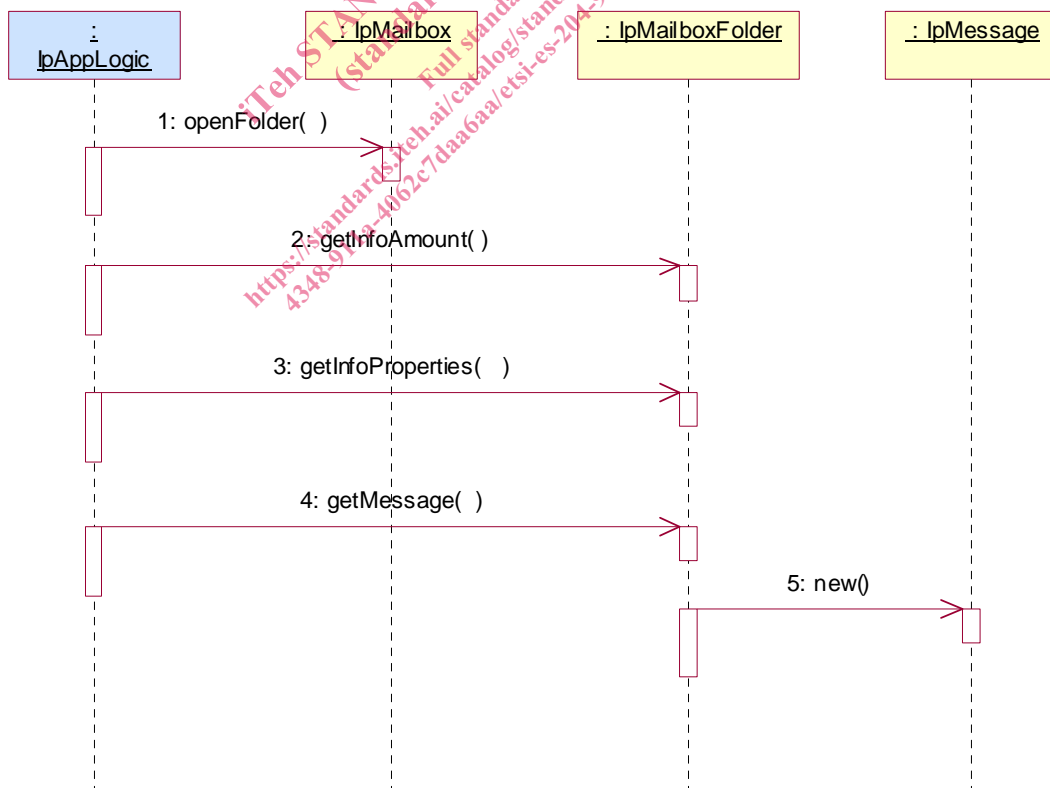


## 5.2 Open Mailbox



- 1: This message requests the object implementing the IpMessagingManager interface to create an object implementing the IpMailbox interface.
- 2: Assuming that the criteria for creating an object implementing the IpMailbox interface is met, message 2 is used to create it.

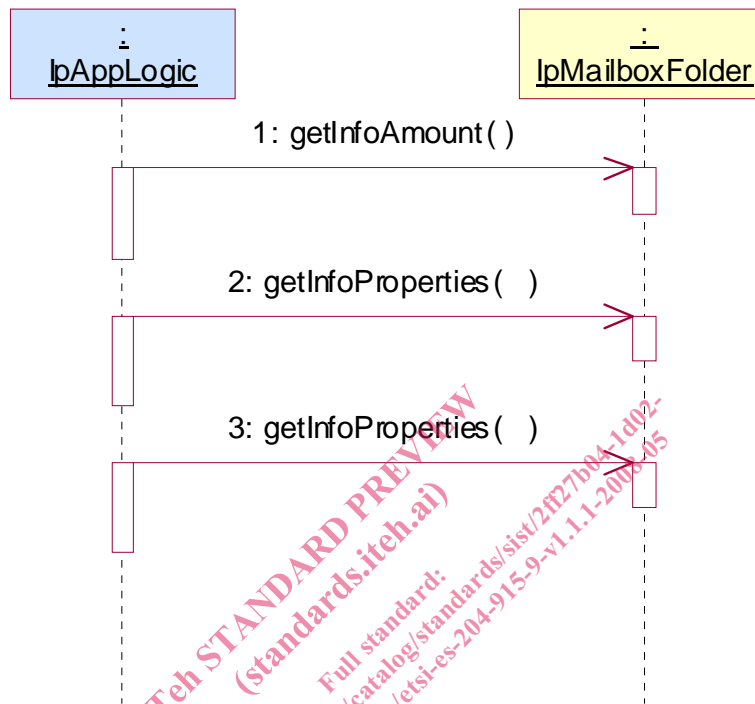
## 5.3 Get Message



- 1: This message requests a folder to be opened and returns a reference to that folder.
- 2: This message requests the number of folder information properties of the opened folder.
- 3: This message requests all of the folder information properties.

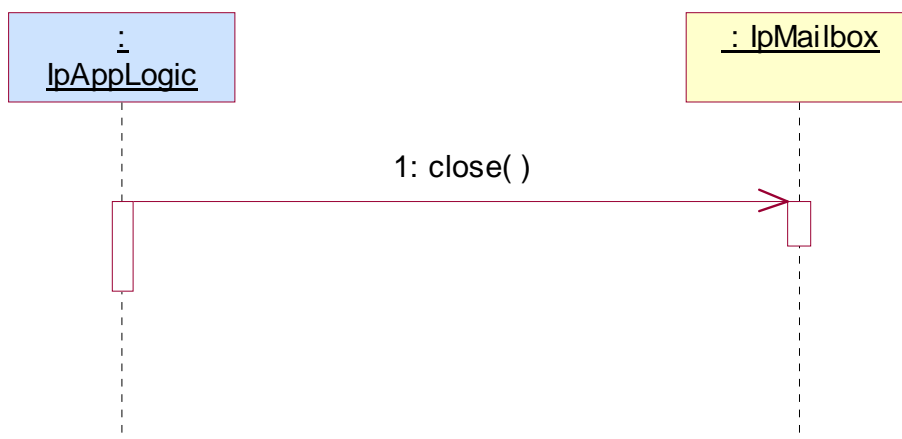
- 4: This message requests a message from the opened mailbox folder.
- 5: Assuming that the criteria for creating an object implementing the IpMessage interface are met, the (internal) message 5 is used to create it.

## 5.4 Get Folder Information



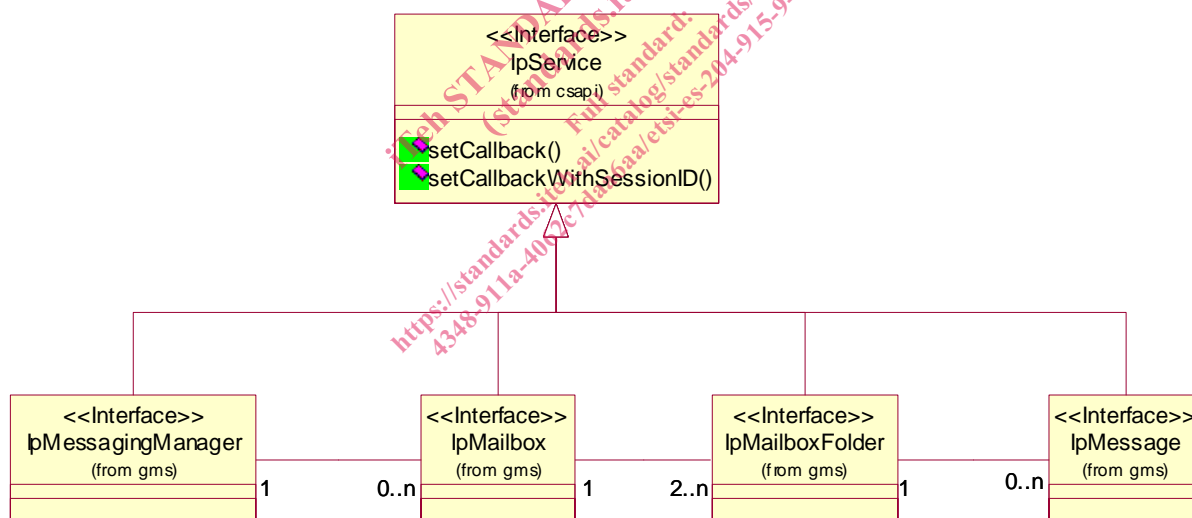
- 1: This message requests the number of folder information properties of the specified folder.
- 2: This message requests the first set of folder information properties.
- 3: This message requests the second set of folder information properties.

## 5.5 Close Mailbox



1: This message requests the object implementing the IpMailbox interface to de-assign.

## 6 Class Diagrams



**Figure 1: Package Overview : Service Interfaces**

The application generic messaging service package consists of only one IpAppMessagingManager interface.

The generic messaging service package consists of one IpMessagingManager interface, zero or more IpMailbox interfaces, zero or more IpMailboxFolder and zero or more IpMessage interfaces.

The class diagram in the following figure shows the interfaces that make up the application generic messaging service package and the generic messaging service package. Communication between these packages is done via the +uses the IpMessagingManager channels. Communication with the IpMailbox and IpMailboxFolder interfaces has to be done via the application logic (not shown).

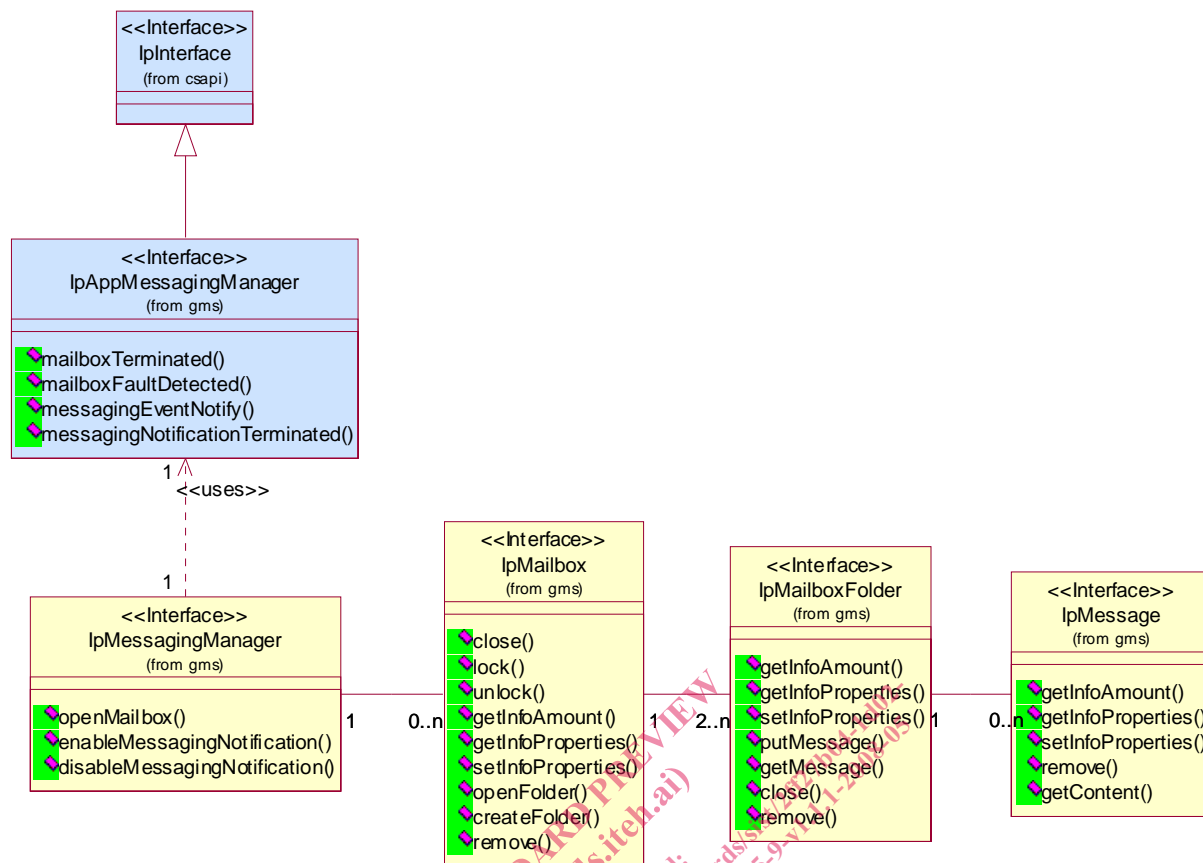


Figure 2: Package Overview: Application and Service Interfaces

## 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.