

**Open Service Access (OSA);
Application Programming Interface (API);
Test Suite Structure and Test Purposes (TSS&TP);
Part 9: Generic messaging SCF
(Parlay 4)**



Reference

DES/TISPAN-06004-09-OSA

Keywords

API, OSA, TSS&TP

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 2005.
All rights reserved.

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

Contents

Intellectual Property Rights	4
Foreword.....	4
1 Scope	5
2 References	5
3 Definitions and abbreviations.....	5
3.1 Definitions	5
3.2 Abbreviations	6
4 Test Suite Structure (TSS).....	6
5 Test Purposes (TP)	6
5.1 Introduction	6
5.1.1 TP naming convention	6
5.1.2 Source of TP definition.....	6
5.1.3 Test strategy.....	6
5.2 TPs for the Generic Messaging SCF	7
5.2.1 Generic Messaging, SCF side	7
5.2.2 Generic Messaging, application side	42
History	47

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. Full details of the entire series can be found in part 1 [6].

To evaluate conformance of a particular implementation, it is necessary to have a set of test purposes to evaluate the dynamic behaviour of the Implementation Under Test (IUT). The specification containing those test purposes is called a Test Suite Structure and Test Purposes (TSS&TP) specification.

1 Scope

The present document provides the Test Suite Structure and Test Purposes (TSS&TP) specification for the Generic Messaging SCF of the Application Programming Interface (API) for Open Service Access (OSA) defined in ES 202 915-9 [1] in compliance with the relevant requirements, and in accordance with the relevant guidance given in ISO/IEC 9646-2 [4] and ETS 300 406 [5].

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

Referenced documents which are not found to be publicly available in the expected location might be found at <http://docbox.etsi.org/Reference>.

- [1] ETSI ES 202 915-9: "Open Service Access (OSA); Application Programming Interface (API); Part 9: Generic Messaging SCF (Parlay 4)".
- [2] ETSI ES 202 363: "Open Service Access (OSA); Application Programming Interface (API); Implementation Conformance Statement (ICS) proforma specification; (Parlay 4)".
- [3] ISO/IEC 9646-1: "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 1: General concepts".
- [4] ISO/IEC 9646-2: "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 2: Abstract Test Suite specification".
- [5] ETSI ETS 300 406: "Methods for Testing and Specification (MTS); Protocol and profile conformance testing specifications; Standardization methodology".
- [6] ETSI ES 202 388-1: "Open Service Access (OSA); Application Programming Interface (API); Test Suite Structure and Test Purposes (TSS&TP); Part 1: Overview (Parlay 4)".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ES 202 915-9 [1], ISO/IEC 9646-1 [3], ISO/IEC 9646-2 [4] and the following apply:

abstract test case: Refer to ISO/IEC 9646-1 [3].

Abstract Test Method (ATM): Refer to ISO/IEC 9646-1 [3].

Abstract Test Suite (ATS): Refer to ISO/IEC 9646-1 [3].

Implementation Under Test (IUT): Refer to ISO/IEC 9646-1 [3].

Lower Tester (LT): Refer to ISO/IEC 9646-1 [3].

Implementation Conformance Statement (ICS): Refer to ISO/IEC 9646-1 [3].

ICS proforma: Refer to ISO/IEC 9646-1 [3].

Implementation eXtra Information for Testing (IXIT): Refer to ISO/IEC 9646-1 [3].

IXIT proforma: Refer to ISO/IEC 9646-1 [3].

Test Purpose (TP): Refer to ISO/IEC 9646-1 [3].

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

API	Application Programming Interface
ATM	Abstract Test Method
ATS	Abstract Test Suite
GM	Generic Messaging
ICS	Implementation Conformance Statement
IUT	Implementation Under Test
IXIT	Implementation eXtra Information for Testing
LT	Lower Tester
OSA	Open Service Access
SCF	Service Capability Feature
TP	Test Purpose
TSS	Test Suite Structure

4 Test Suite Structure (TSS)

- Generic Messaging

5 Test Purposes (TP)

5.1 Introduction

For each test requirement a TP is defined.

5.1.1 TP naming convention

Tps are numbered, starting at 01, within each group. Groups are organized according to the TSS. Additional references are added to identify the actual test suite (see table 1).

Table 1: TP identifier naming convention scheme

Identifier: <suite_id>_<group>_<nnn>	
<suite_id>	= SCG name: "GM" for Generic Messaging part of Generic Messaging SCF 'GM_APP' for Generic Messaging part of Generic Messaging Application
<nn>	= sequential number: (01 to 99)

5.1.2 Source of TP definition

The TPs are based on ES 202 915-9 [1].

5.1.3 Test strategy

As the base standard ES 202 915-9 [1] contains no explicit requirements for testing, the TPs were generated as a result of an analysis of the base standard and the ICS specification ES 202 363 [2].

The TPs are only based on conformance requirements related to the externally observable behaviour of the IUT and are limited to conceivable situations to which a real implementation is likely to be faced (see ETS 300 406 [5]).

5.2 TPs for the Generic Messaging SCF

All ICS items referred to in this clause are as specified in ES 202 363 [2] unless indicated otherwise by another numbered reference.

All parameters specified in method calls are valid unless specified.

The procedures to trigger the SCF to call methods in the application are dependant on the underlying network architecture and are out of the scope of the present document. Those method calls are preceded by the words "Triggered action".

5.2.1 Generic Messaging, SCF side

Test GM_01

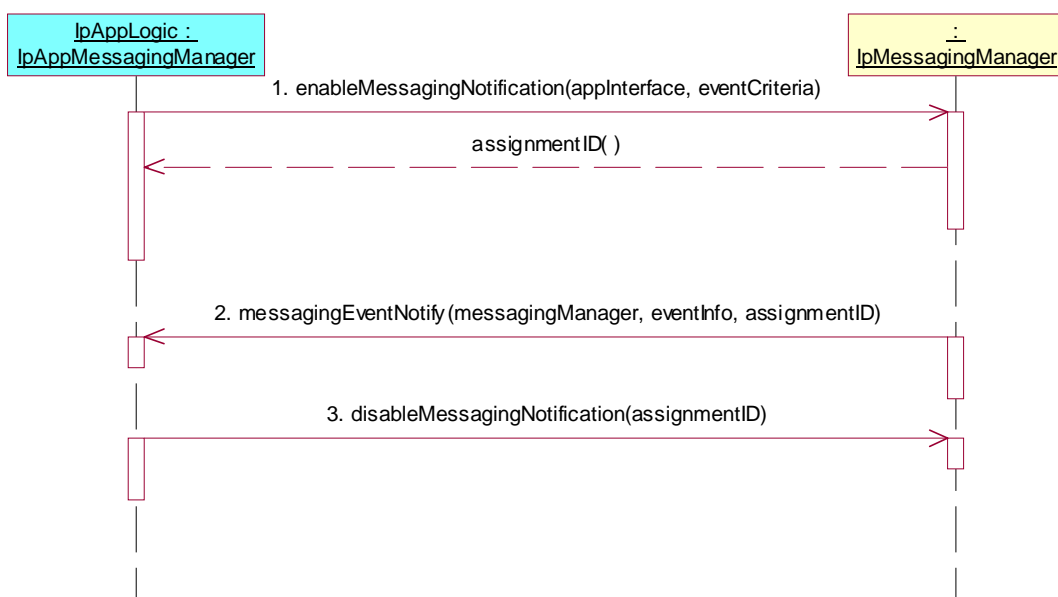
Summary: IpMessagingManager: all methods, successful.

Reference: ES 202 915-9 [1], clause 8.1.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call **enableMessagingNotification()**
Parameters: appInterface, eventCriteria
Check: valid value of TpAssignmentID is returned
2. Triggered action: cause IUT to call **messagingEventNotify()** method on the tester's (Application) **IpAppMessagingManager** interface.
Parameters: messagingManager, eventInfo, assignmentID
3. Method call **disableMessagingNotification()**
Parameters: assignmentId given in 1.
Check: no exception is returned



Test GM_02

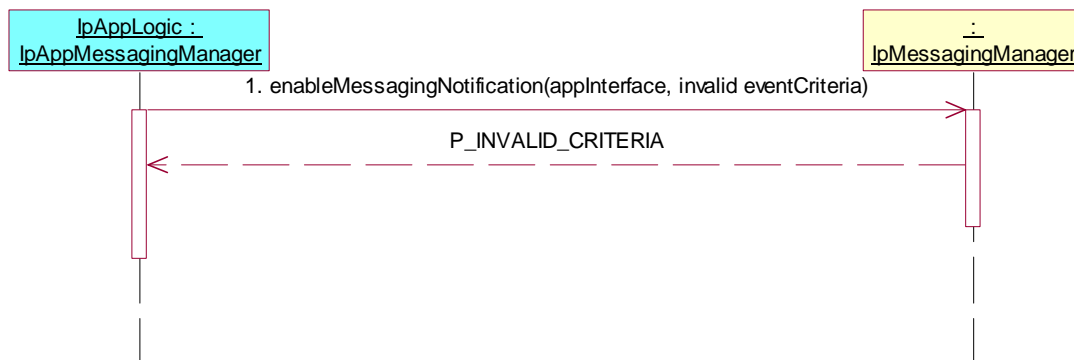
Summary: enableMessagingNotification, P_INVALID_CRITERIA.

Reference: ES 202 915-9 [1], clause 8.1.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call **enableMessagingNotification()**
Parameters: appInterface, invalid eventCriteria
Check: P_INVALID_CRITERIA is returned



Test GM_03

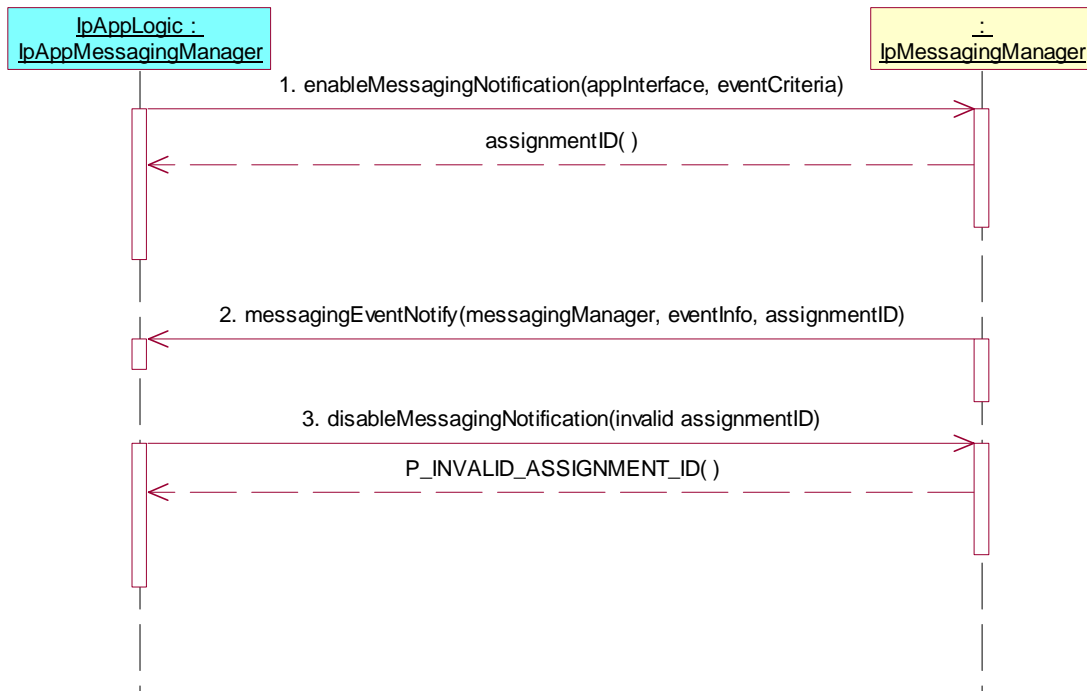
Summary: disableMessagingNotification, P_INVALID_ASSIGNMENT_ID.

Reference: ES 202 915-9 [1], clause 8.1.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call **enableMessagingNotification()**
 Parameters: appInterface, eventCriteria
 Check: valid value of TpAssignmentID is returned
2. Triggered action: cause IUT to call **messagingEventNotify()** method on the tester's (Application) **IpAppMessagingManager** interface.
 Parameters: messagingManager, eventInfo, assignmentID
3. Method call **disableMessagingNotification()**
 Parameters: invalid assignmentId
 Check: P_INVALID_ASSIGNMENT_ID is returned



Test GM_04

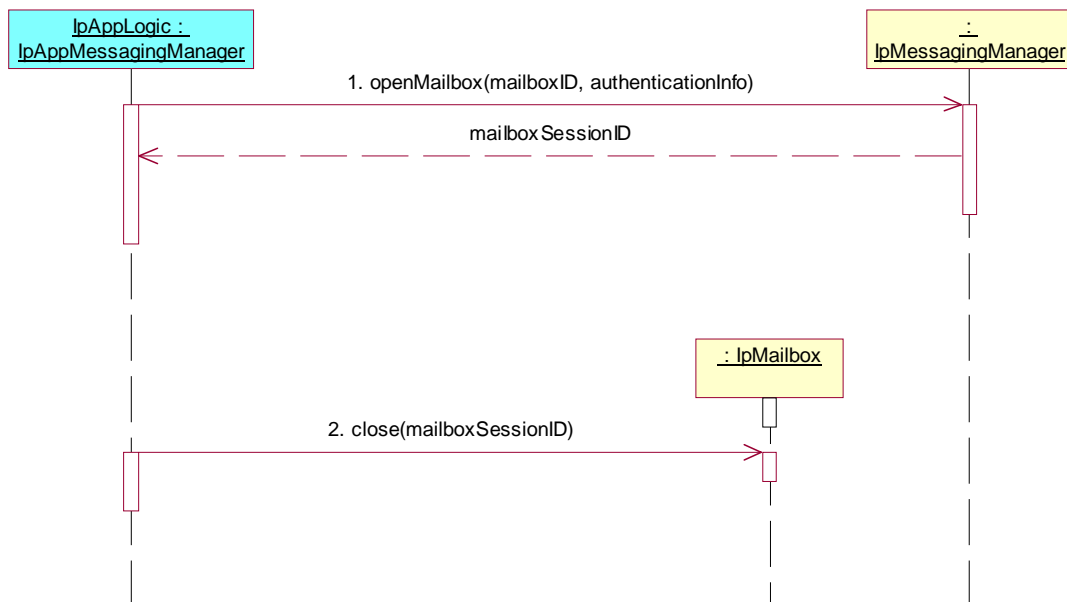
Summary: IpMessagingManager: all methods, successful.

Reference: ES 202 915-9 [1], clause 8.1.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call **openMailbox()** on IpMessagingManager
 Parameters: mailboxID, authenticationInfo
 Check: valid value of TpMailboxIdentifier is returned
2. Method call **close()** on IpMailbox
 Parameters: mailboxSessionID given in 1.
 Check: no exception is returned



Test GM_05

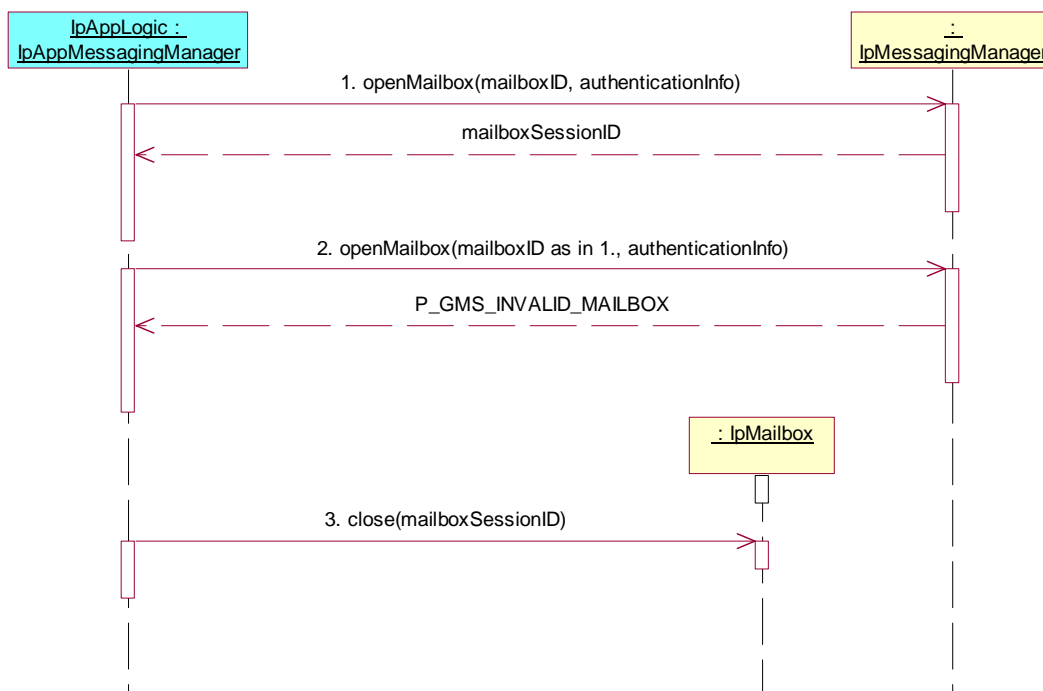
Summary: openMailbox, P_GMS_INVALID_MAILBOX.

Reference: ES 202 915-9 [1], clause 8.1.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call **openMailbox()** on IpMessagingManager
Parameters: mailboxID, authenticationInfo
Check: valid value of TpMailboxIdentifier is returned
2. Method call **openMailbox()**
Parameters: mailboxID given in 1., authenticationInfo
Check: P_GMS_INVALID_MAILBOX is returned
3. Method call **close()** on IpMailbox
Parameters: mailboxSessionID given in 1.
Check: no exception is returned



Test GM_06

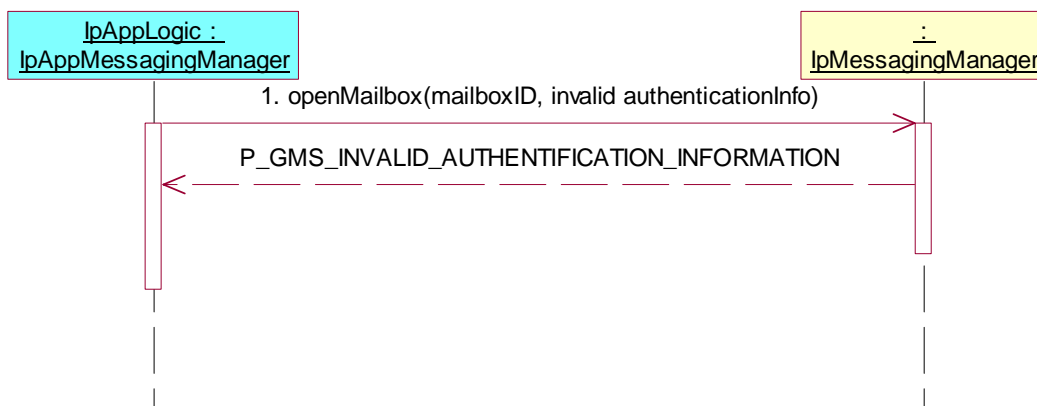
Summary: openMailbox, P_GMS_INVALID_AUTHENTICATION_INFORMATION.

Reference: ES 202 915-9 [1], clause 8.1.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call **openMailbox()**
Parameters: mailboxID, invalid authenticationInfo
Check: P_GMS_INVALID_AUTHENTICATION_INFORMATION is returned

**Test GM_07**

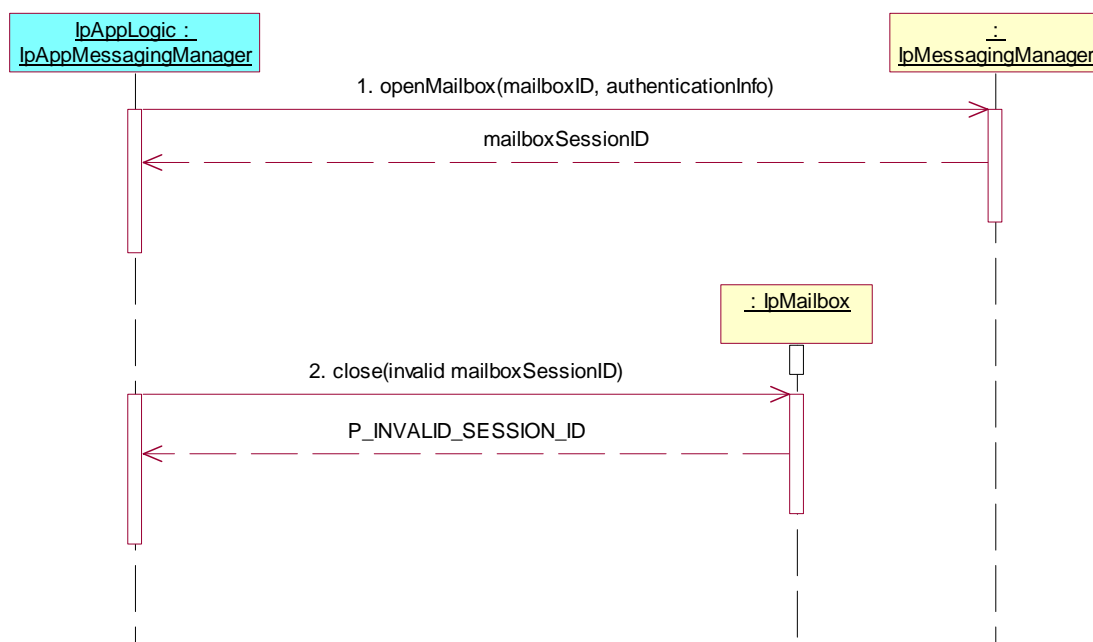
Summary: close, P_INVALID_SESSION_ID.

Reference: ES 202 915-9 [1], clause 8.3.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call **openMailbox()** on IpMessagingManager
Parameters: mailboxID, authenticationInfo
Check: valid value of TpMailboxIdentifier is returned
2. Method call **close()** on IpMailbox
Parameters: invalid mailboxSessionID
Check: P_INVALID_SESSION_ID is returned

**Test GM_08**

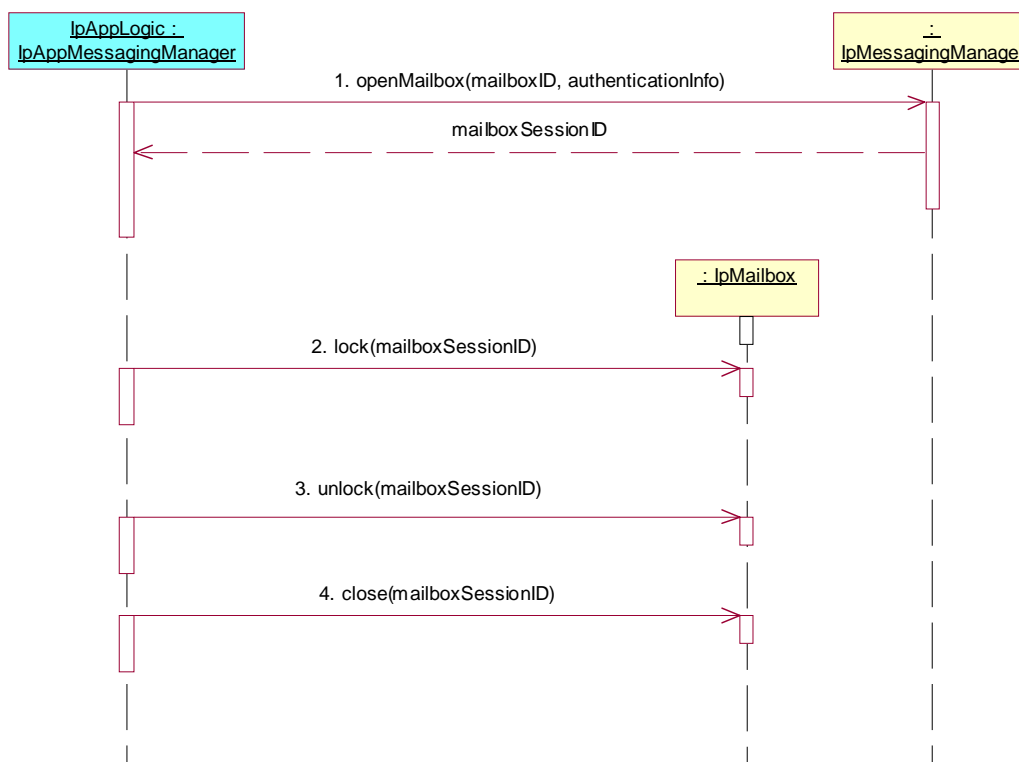
Summary: IpMailbox all methods, successful.

Reference: ES 202 915-9 [1], clause 8.1.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call **openMailbox()** on IpMessagingManager
Parameters: mailboxID, authenticationInfo
Check: valid value of TpMailboxIdentifier is returned
2. Method call **lock()** on IpMailbox
Parameters: mailboxSessionID given in 1.
Check: no exception is returned
3. Method call **unlock()**
Parameters: mailboxSessionID given in 1.
Check: no exception is returned
4. Method call **close()**
Parameters: mailboxSessionID given in 1.
Check: no exception is returned



Test GM_09

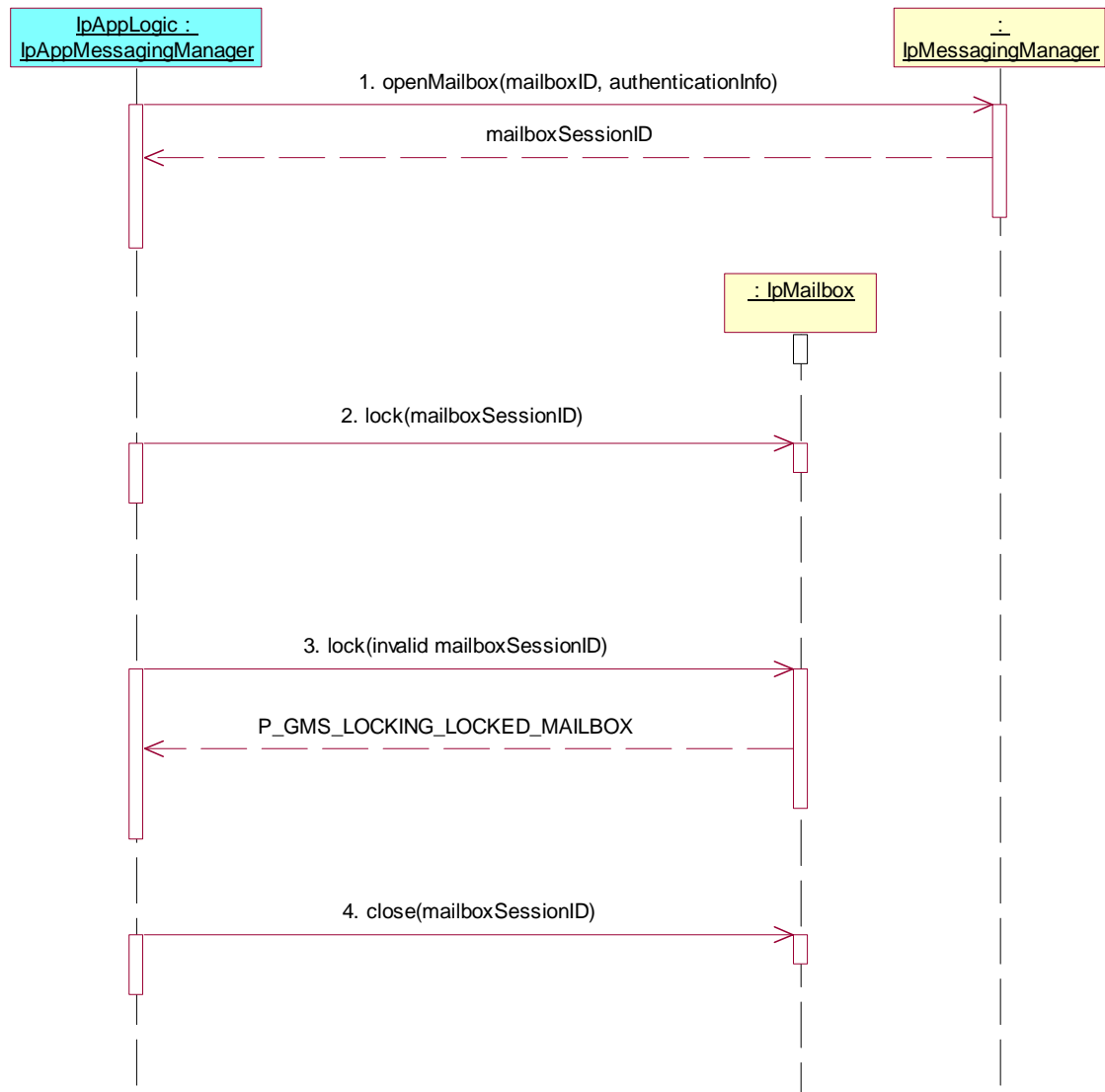
Summary: lock, P_GMS_LOCKING_LOCKED_MAILBOX.

Reference: ES 202 915-9 [1], clause 8.3.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call **openMailbox()** on IpMessagingManager
Parameters: mailboxID, authenticationInfo
Check: valid value of TpMailboxIdentifier is returned
2. Method call **lock()** on IpMailbox
Parameters: mailboxSessionID given in 1.
Check: no exception is returned
3. Method call **lock()**
Parameters: mailboxSessionID given in 1.
Check: P_GMS_LOCKING_LOCKED_MAILBOX is returned
4. Method call **close()**
Parameters: mailboxSessionID given in 1.
Check: no exception is returned



Test GM_10

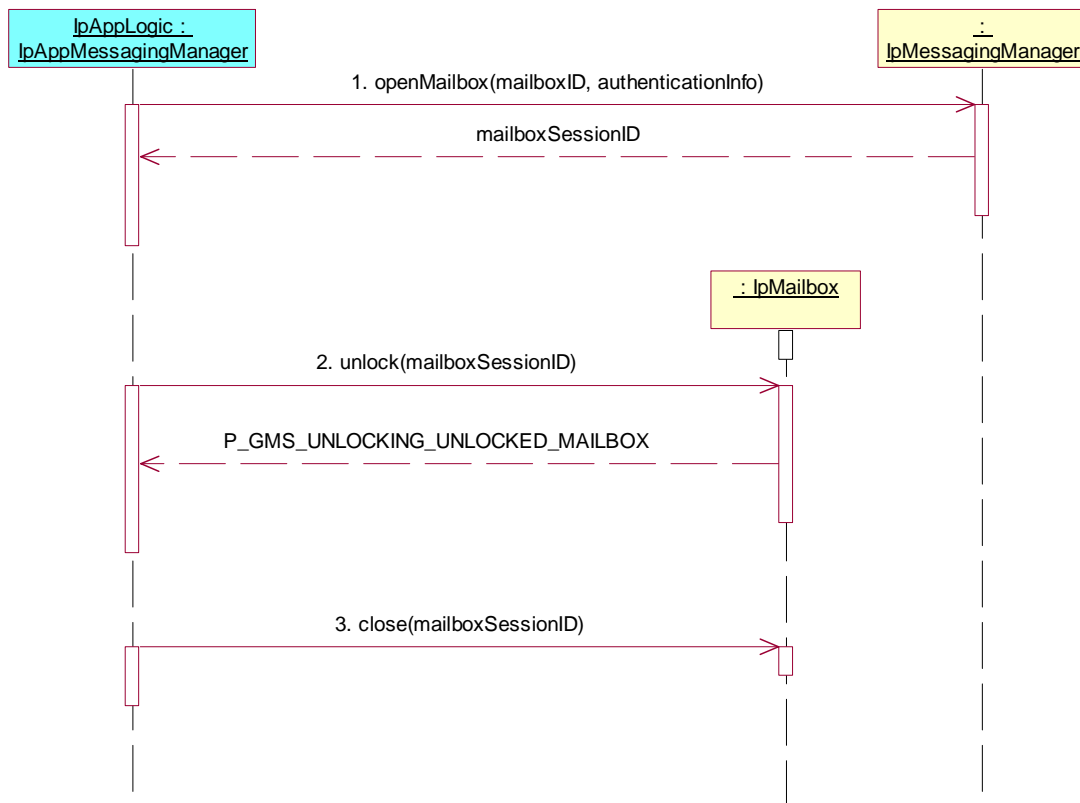
Summary: unlock, P_GMS_UNLOCKING_UNLOCKED_MAILBOX.

Reference: ES 202 915-9 [1], clause 8.3.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call **openMailbox()** on IpMessagingManager
 Parameters: mailboxID, authenticationInfo
 Check: valid value of TpMailboxIdentifier is returned
2. Method call **unlock()** on IpMailbox
 Parameters: mailboxSessionID given in 1.
 Check: P_GMS_UNLOCKING_UNLOCKED_MAILBOX is returned
3. Method call **close()**
 Parameters: mailboxSessionID given in 1.
 Check: no exception is returned



Test GM_11

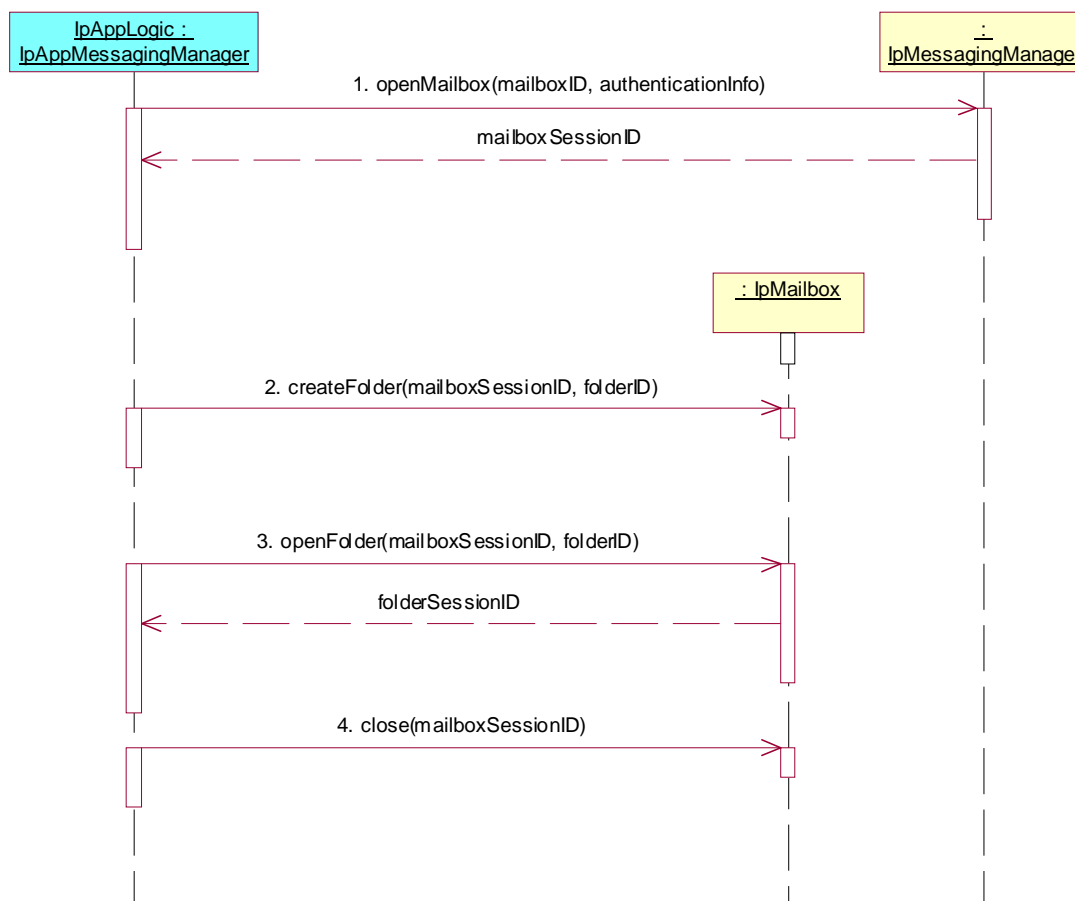
Summary: IpMailbox, all methods, successful.

Reference: ES 202 915-9 [1], clause 8.1.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call **openMailbox()** on IpMessagingManager
Parameters: mailboxID, authenticationInfo
Check: valid value of TpMailboxIdentifier is returned
2. Method call **createFolder()** on IpMailbox interface
Parameters: mailboxSessionID given in 1., folderID
Check: no exception is returned
3. Method call **openFolder()**
Parameters: mailboxSessionID given in 1., folderID given in 2.
Check: valid value of TpMailboxFolderIdentifier is returned
4. Method call **close()**
Parameters: mailboxSessionID given in 1.
Check: no exception is returned



Test GM_12

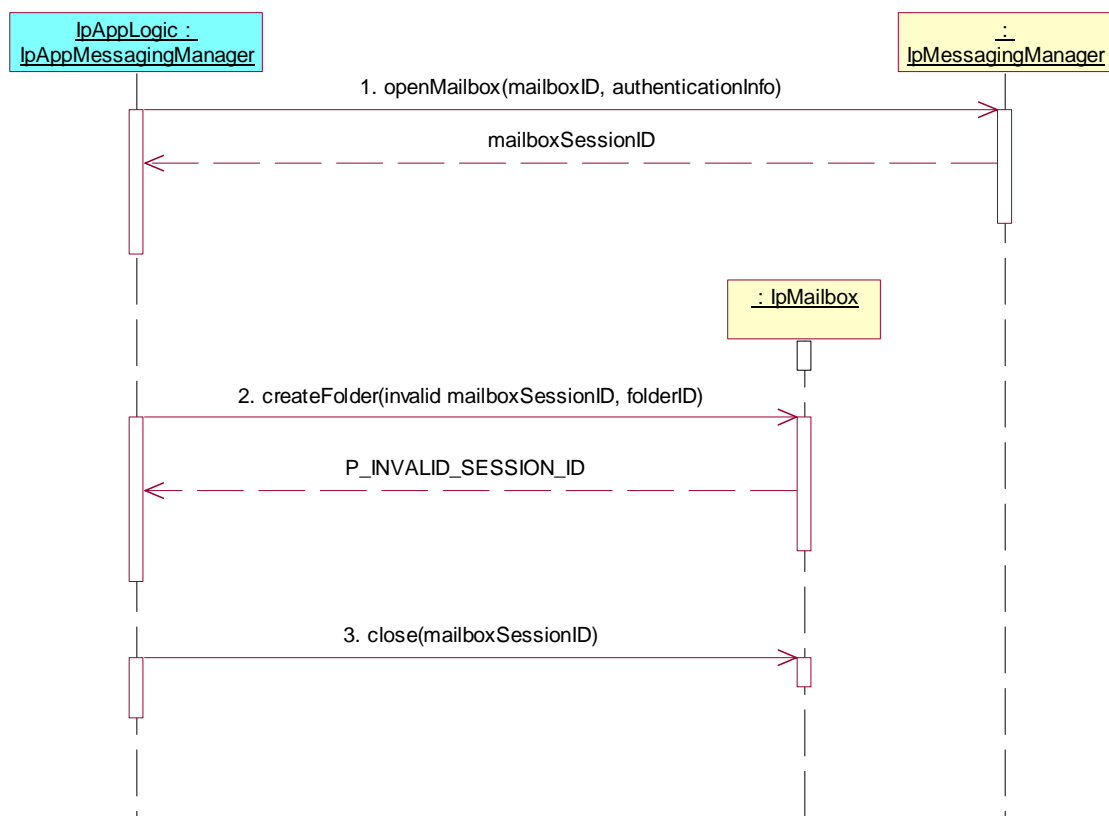
Summary: createFolder, P_INVALID_SESSION_ID.

Reference: ES 202 915-9 [1], clause 8.3.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call **openMailbox()** on IpMessagingManager
 Parameters: mailboxID, authenticationInfo
 Check: valid value of TpMailboxIdentifier is returned
2. Method call **createFolder()** on IpMailbox interface
 Parameters: invalid mailboxSessionID, folderID
 Check: P_INVALID_SESSION_ID is returned
3. Method call **close()**
 Parameters: mailboxSessionID given in 1.
 Check: no exception is returned

**Test GM_13**

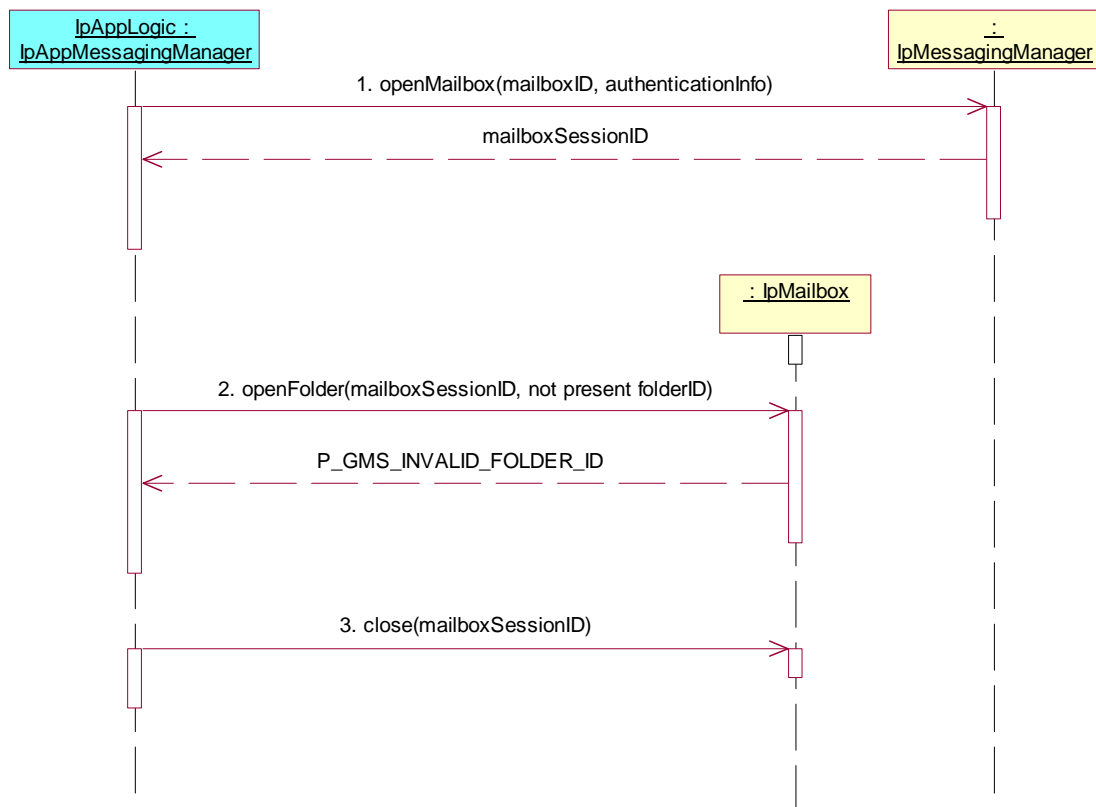
Summary: openFolder, P_GMS_INVALID_FOLDER_ID.

Reference: ES 202 915-9 [1], clause 8.3.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

- Method call **openMailbox()** on IpMessagingManager
Parameters: mailboxID, authenticationInfo
Check: valid value of TpMailboxIdentifier is returned
- Method call **openFolder()** on IpMailbox interface
Parameters: mailboxSessionID given in 1., invalid folderID
Check: P_GMS_INVALID_FOLDER_ID is returned
- Method call **close()**
Parameters: mailboxSessionID given in 1.
Check: no exception is returned

**Test GM_14**

Summary: openFolder, P_GMS_FOLDER_IS_OPEN.

Reference: ES 202 915-9 [1], clause 8.3.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

- Method call **openMailbox()** on IpMessagingManager
Parameters: mailboxID, authenticationInfo
Check: valid value of TpMailboxIdentifier is returned
- Method call **openFolder()** on IpMailbox interface
Parameters: mailboxSessionID given in 1., folderID
Check: valid value of TpMailboxFolderIdentifier is returned
- Method call **openFolder()** on IpMailbox interface
Parameters: mailboxSessionID given in 1., folderID given in 2.
Check: P_GMS_FOLDER_IS_OPEN is returned
- Method call **close()**
Parameters: mailboxSessionID given in 1.
Check: no exception is returned



Test GM_15

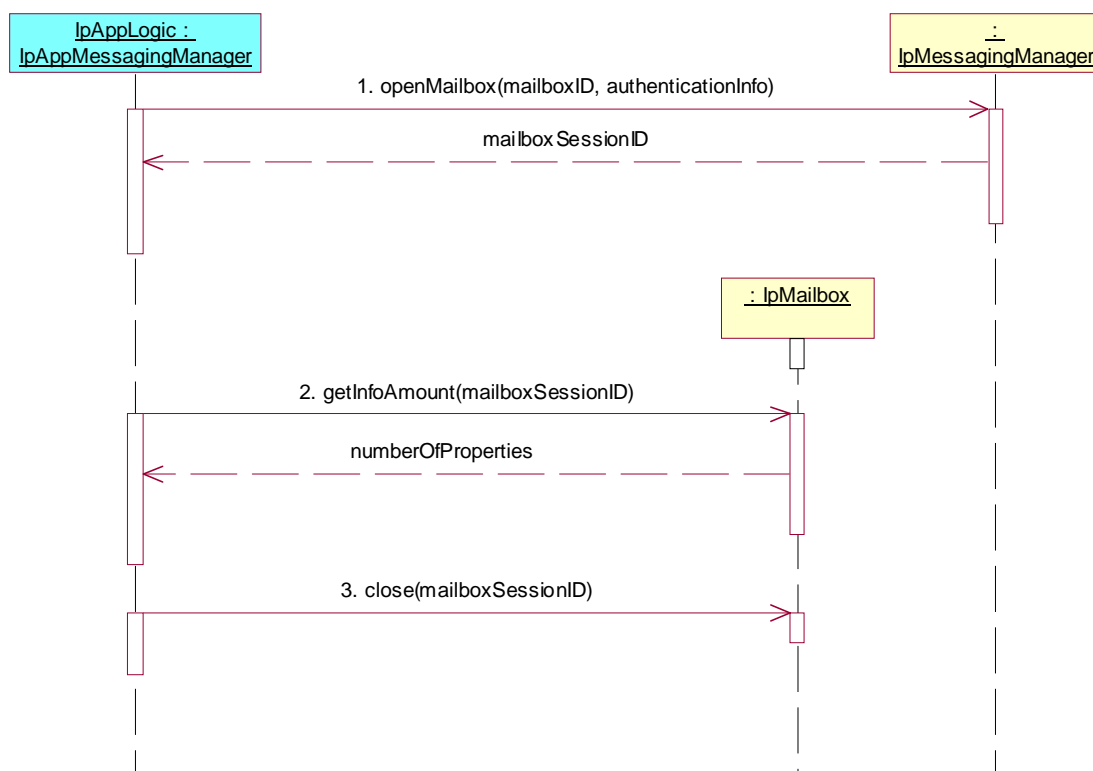
Summary: IpMailbox, all methods, successful.

Reference: ES 202 915-9 [1], clause 8.1.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call **openMailbox()** on IpMessagingManager
Parameters: mailboxID, authenticationInfo
Check: valid value of TpMailboxIdentifier is returned
2. Method call **getInfoAmount()** on IpMailbox
Parameters: mailboxSessionID given in 1.
Check: valid value of Tpint32 is returned
3. Method call **close()**
Parameters: mailboxSessionID given in 1.
Check: no exception is returned

**Test GM_16**

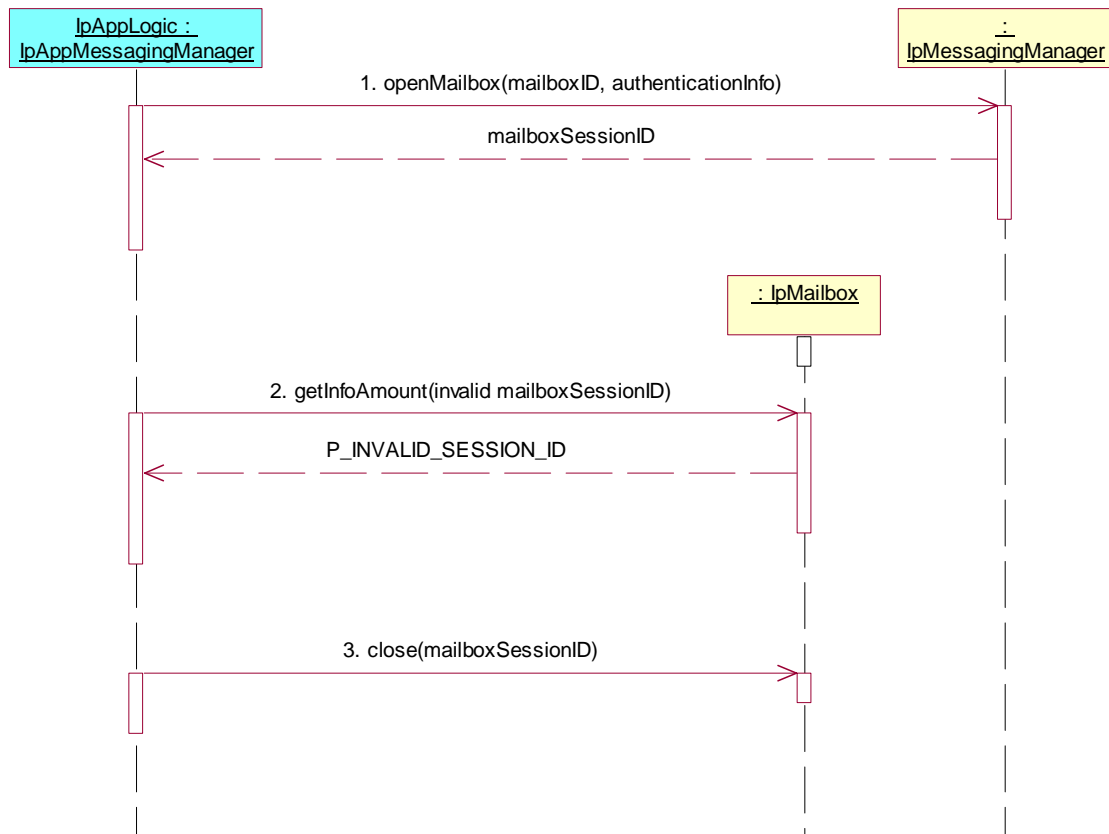
Summary: `getInfoAmount`, : P_INVALID_SESSION_ID.

Reference: ES 202 915-9 [1], clause 8.1.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

- Method call **openMailbox()** on IpMessagingManager
Parameters: mailboxID, authenticationInfo
Check: valid value of TpMailboxIdentifier is returned
- Method call **getInfoAmount()** on IpMailbox
Parameters: invalid mailboxSessionID
Check: P_INVALID_SESSION_ID is returned
- Method call **close()**
Parameters: mailboxSessionID given in 1.
Check: no exception is returned

**Test GM_17**

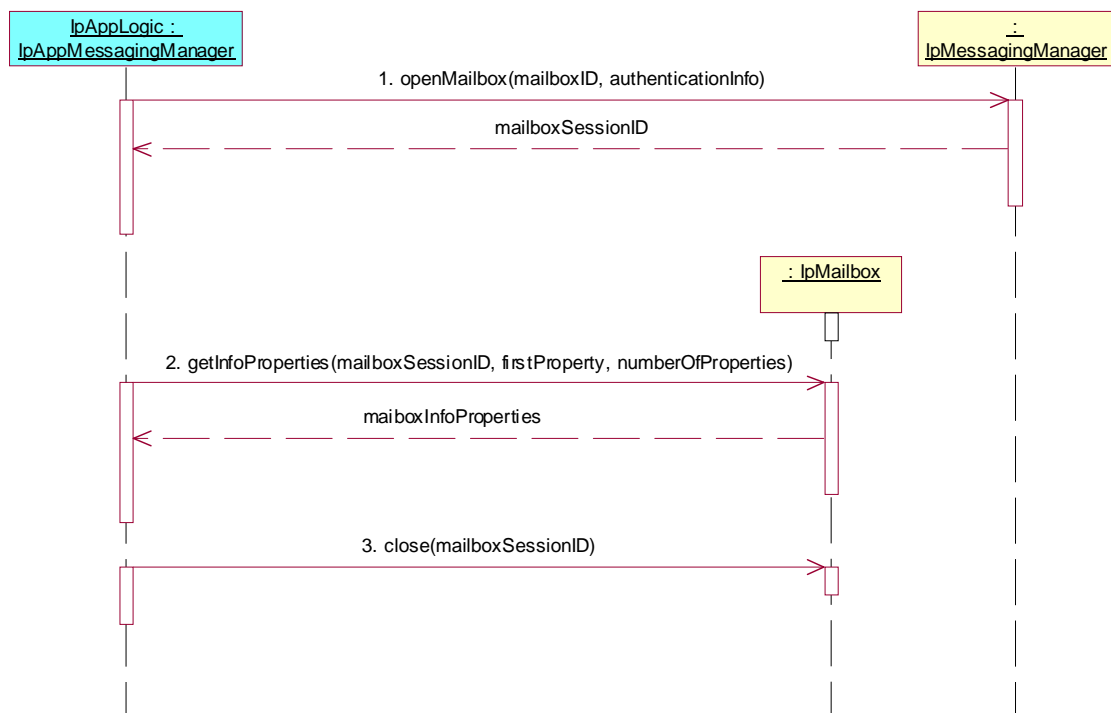
Summary: all methods, successful.

Reference: ES 202 915-9 [1], clause 8.1.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

- Method call **openMailbox()** on IpMessagingManager
 Parameters: mailboxID, authenticationInfo
 Check: valid value of TpMailboxIdentifier is returned
- Method call **getInfoProperties()** on IpMailbox
 Parameters: mailboxSessionID given in 1., firstProperty, numberOfProperties
 Check: valid value of TpMailboxInfoPropertySet is returned
- Method call **close()**
 Parameters: mailboxSessionID given in 1.
 Check: no exception is returned

**Test GM_18**

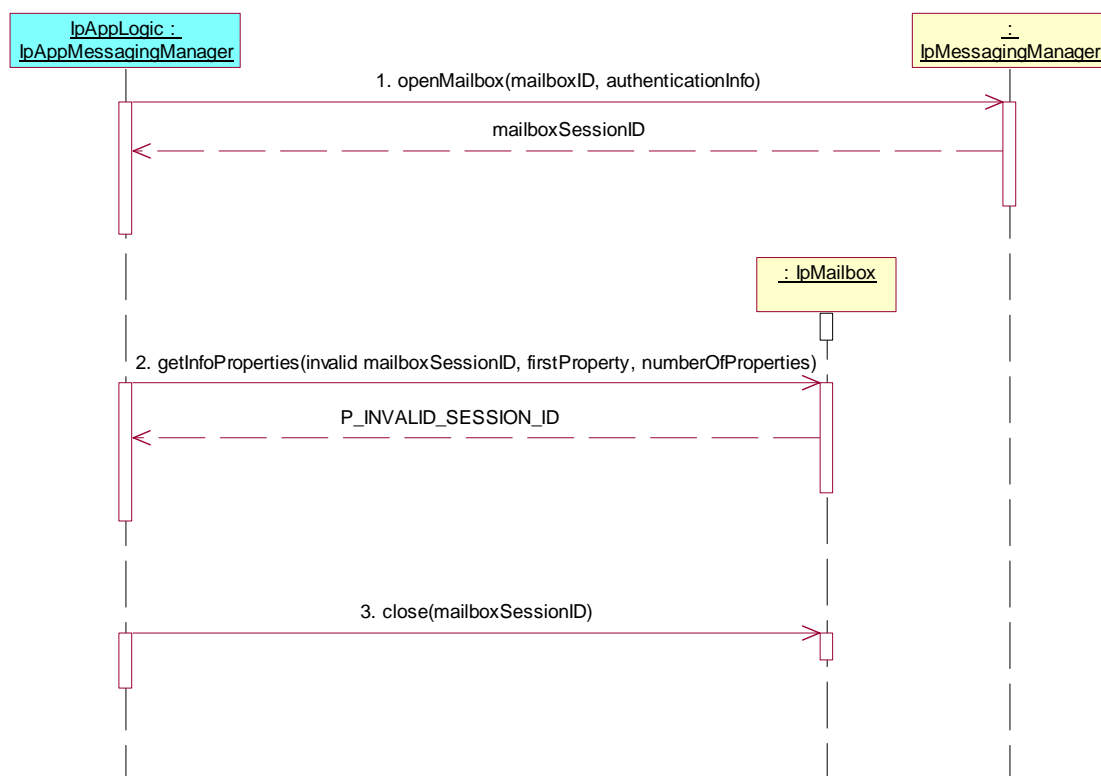
Summary: getInfoProperties, P_INVALID_SESSION_ID.

Reference: ES 202 915-9 [1], clause 8.4.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

- Method call **openMailbox()** on IpMessagingManager
Parameters: mailboxID, authenticationInfo
Check: valid value of TpMailboxIdentifier is returned
- Method call **getInfoProperties()** on IpMailbox
Parameters: invalid mailboxSessionID, firstProperty, numberOfProperties
Check: P_INVALID_SESSION_ID is returned
- Method call **close()**
Parameters: mailboxSessionID given in 1.
Check: no exception is returned

**Test GM_19**

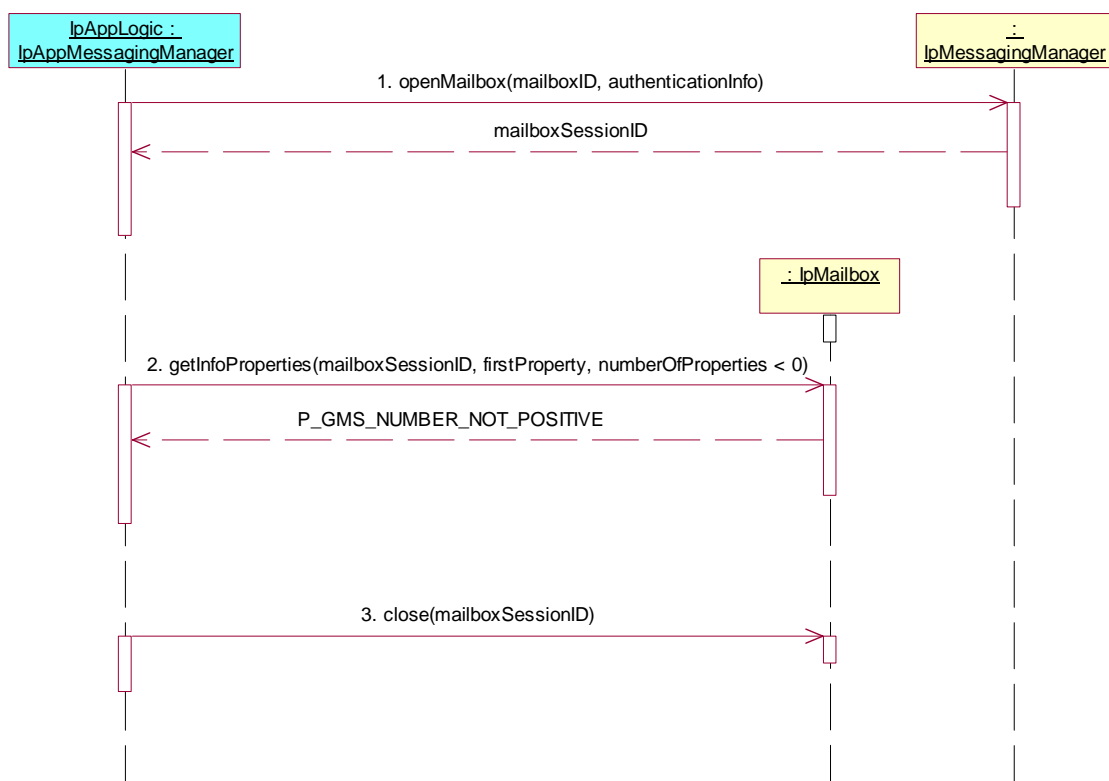
Summary: getInfoProperties, P_GMS_NUMBER_NOT_POSITIVE.

Reference: ES 202 915-9 [1], clause 8.4.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

- Method call **openMailbox()** on IpMessagingManager
Parameters: mailboxID, authenticationInfo
Check: valid value of TpMailboxIdentifier is returned
- Method call **getInfoProperties()** on IpMailbox
Parameters: mailboxSessionID, firstProperty, negative numberOfProperties
Check: P_GMS_NUMBER_NOT_POSITIVE is returned
- Method call **close()**
Parameters: mailboxSessionID given in 1.
Check: no exception is returned



Test GM_20

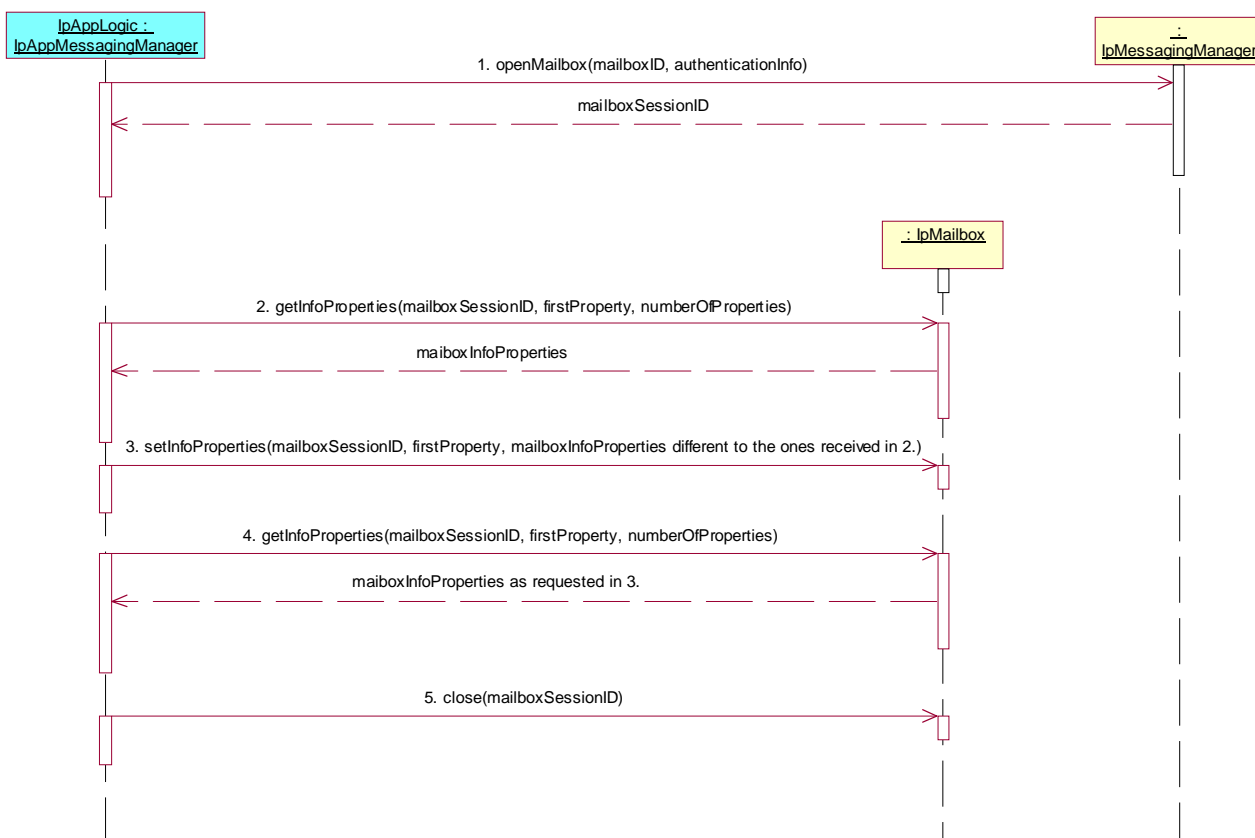
Summary: IpMailbox, all methods, successful.

Reference: ES 202 915-9 [1], clause 8.1.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

- Method call **openMailbox()** on IpMessagingManager
Parameters: mailboxID, authenticationInfo
Check: valid value of TpMailboxIdentifier is returned
- Method call **getInfoProperties()** on IpMailbox
Parameters: mailboxSessionID given in 1, firstProperty, numberOfProperties
Check: valid value of TpMailboxInfoPropertySet is returned
- Method call **setInfoProperties()**
Parameters: mailboxSessionID given in 1., firstProperty, mailboxInfoProperties different from result in 2.
Check: no exception is returned
- Method call **getInfoProperties()**
Parameters: mailboxSessionID given in 1., same firstProperty given in 3., numberOfProperties
Check: valid value of TpMailboxInfoPropertySet is returned with properties match mailboxInfoProperties given in 3.
- Method call **close()**
Parameters: mailboxSessionID given in 1.
Check: no exception is returned



Test GM_21

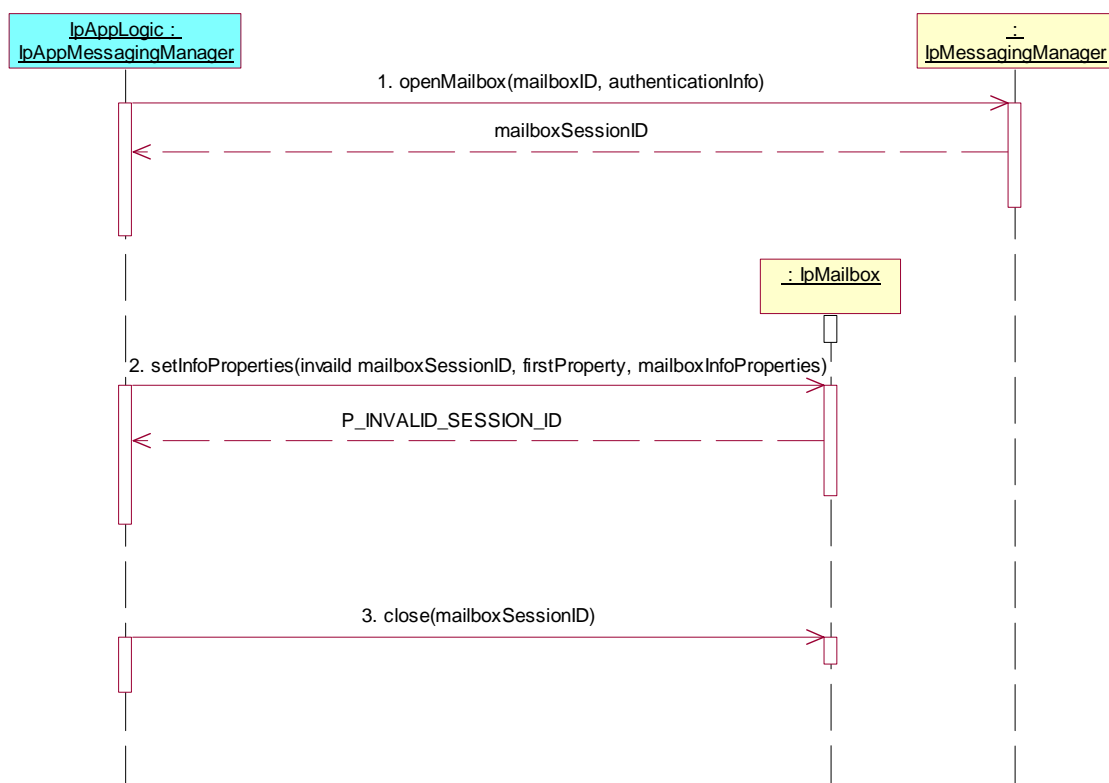
Summary: setInfoProperties, P_INVALID_SESSION_ID.

Reference: ES 202 915-9 [1], clause 8.4.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call **openMailbox()** on IpMessagingManager
Parameters: mailboxID, authenticationInfo
Check: valid value of TpMailboxIdentifier is returned
2. Method call **setInfoProperties()** on IpMailbox
Parameters: invalid mailboxSessionID, firstProperty, mailboxInfoProperties
Check: P_INVALID_SESSION_ID is returned
3. Method call **close()**
Parameters: mailboxSessionID given in 1.
Check: no exception is returned

**Test GM_22**

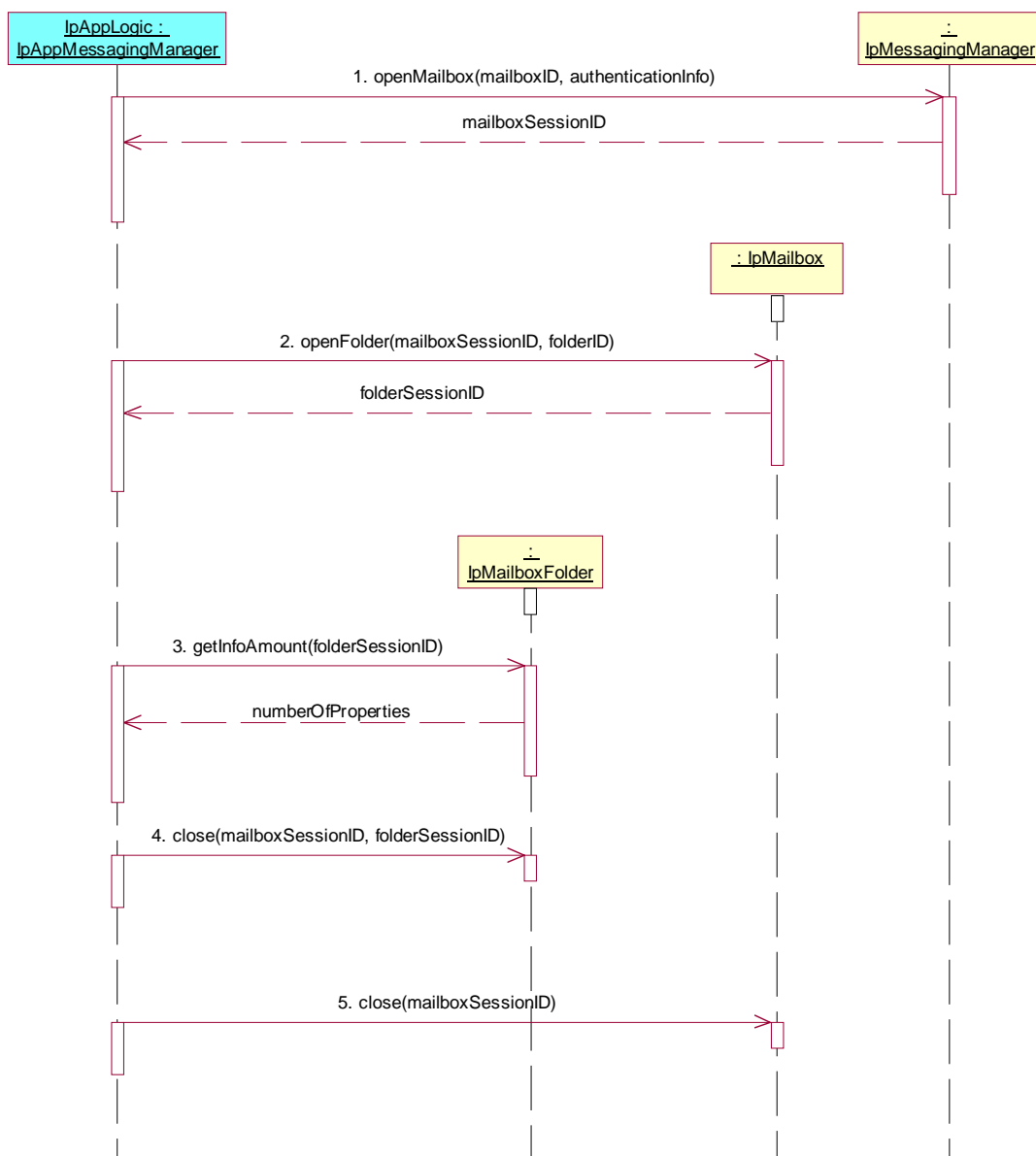
Summary: IpMailbox, all methods, successful.

Reference: ES 202 915-9 [1], clause 8.1.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

- Method call **openMailbox()** on IpMessagingManager
Parameters: mailboxID, authenticationInfo
Check: valid value of TpMailboxIdentifier is returned
- Method call **openFolder()** on IpMailbox
Parameters: mailboxSessionID given in 1., folderID
Check: valid value of TpMailboxFolderIdentifier is returned
- Method call **getInfoAmount()** on IpMailboxFolder
Parameters: folderSessionID given in 2.
Check: valid value of TpInt32 is returned
- Method call **close()** on IpMailboxFolder
Parameters: mailboxSessionID given in 1., folderSessionID given in 2.
Check: no exception is returned
- Method call **close()** on IpMailbox
Parameters: mailboxSessionID given in 1.
Check: no exception is returned

**Test GM_23**

Summary: **getInfoAmount, P_INVALID_SESSION_ID.**

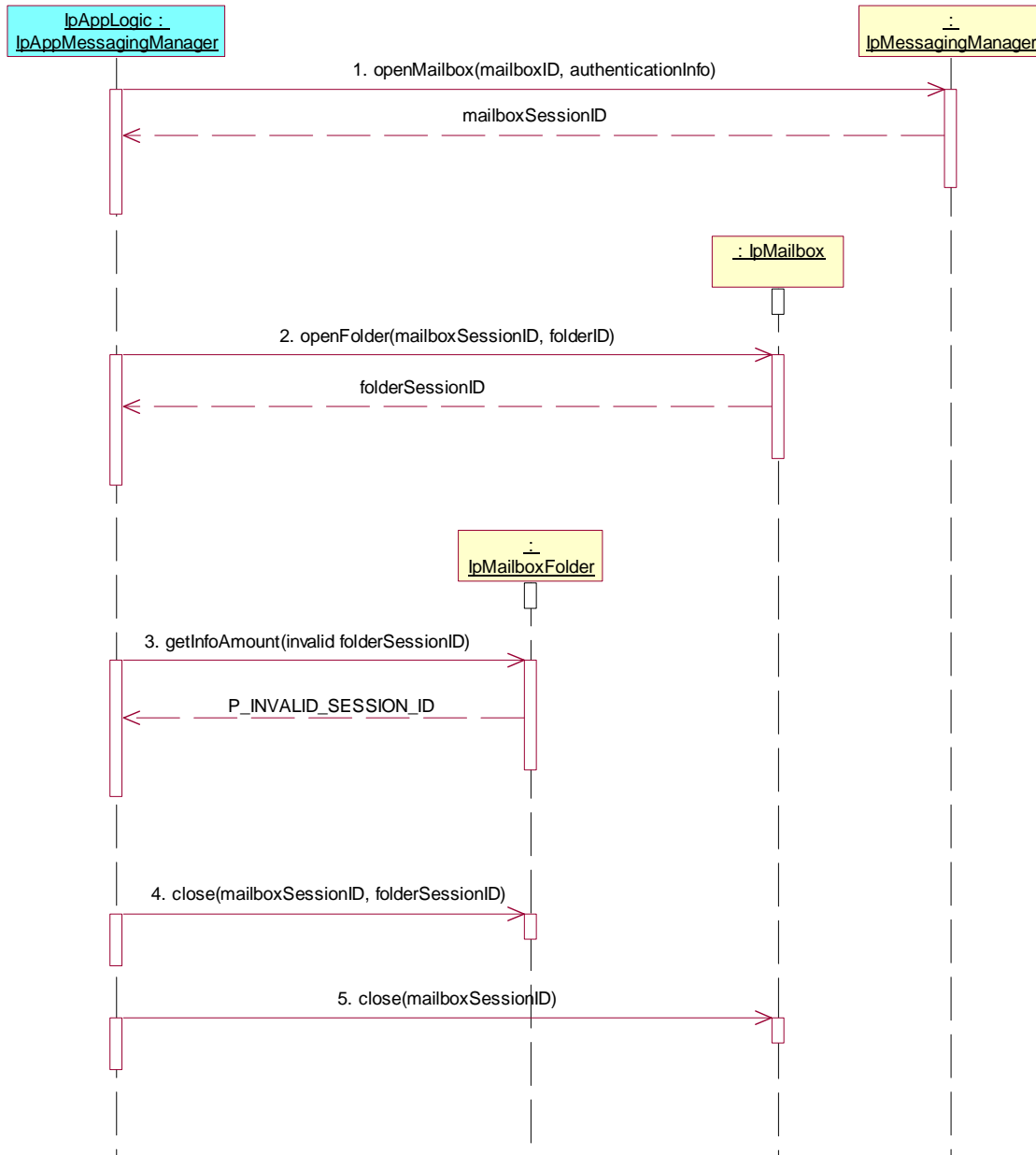
Reference: ES 202 915-9 [1], clause 8.4.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

- Method call **openMailbox()** on IpMessagingManager
Parameters: mailboxID, authenticationInfo
Check: valid value of TpMailboxIdentifier is returned
- Method call **openFolder()** on IpMailbox
Parameters: mailboxSessionID given in 1., folderID
Check: valid value of TpMailboxFolderIdentifier is returned
- Method call **getInfoAmount()** on IpMailboxFolder
Parameters: invalid folderSessionID
Check: P_INVALID_SESSION_ID is returned

4. Method call **close()** on IpMailboxFolder
Parameters: mailboxSessionID given in 1., folderSessionID given in 2.
Check: no exception is returned
5. Method call **close()** on IpMailbox
Parameters: mailboxSessionID given in 1.
Check: no exception is returned



Test GM_24

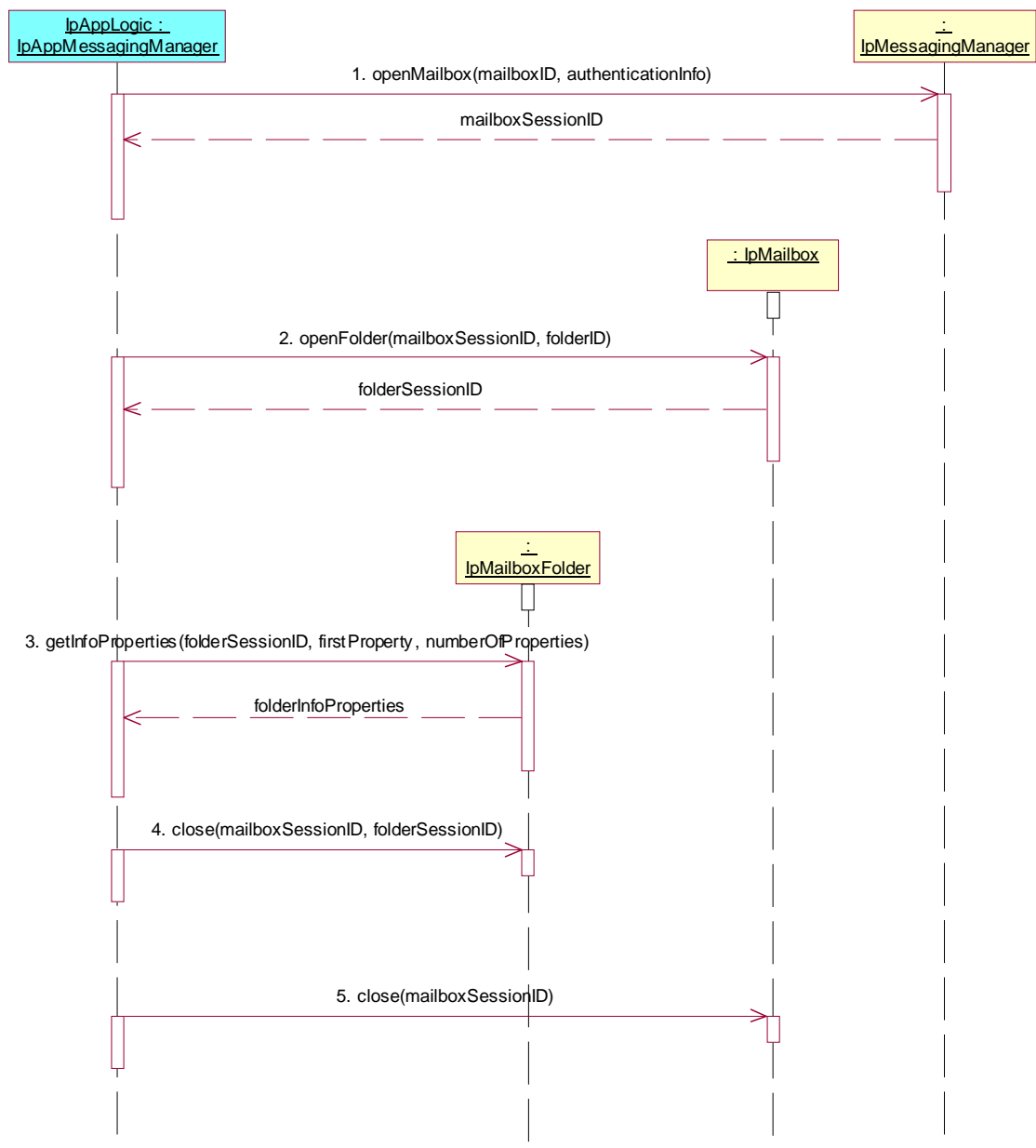
Summary: IpMailboxFolder all methods, successful.

Reference: ES 202 915-9 [1], clause 8.5.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call **openMailbox()** on IpMessagingManager
Parameters: mailboxID, authenticationInfo
Check: valid value of TpMailboxIdentifier is returned
2. Method call **openFolder()** on IpMailbox
Parameters: mailboxSessionID given in 1., folderID
Check: valid value of TpMailboxFolderIdentifier is returned
3. Method call **getInfoProperties()** on IpMailboxFolder
Parameters: folderSessionID given in 2., firstProperty, numberOfProperties
Check: valid value of TpFolderInfoPropertySet is returned
4. Method call **close()** on IpMailboxFolder
Parameters: mailboxSessionID given in 1., folderSessionID given in 2.
Check: no exception is returned
5. Method call **close()** on IpMailbox
Parameters: mailboxSessionID given in 1.
Check: no exception is returned



Test GM_25

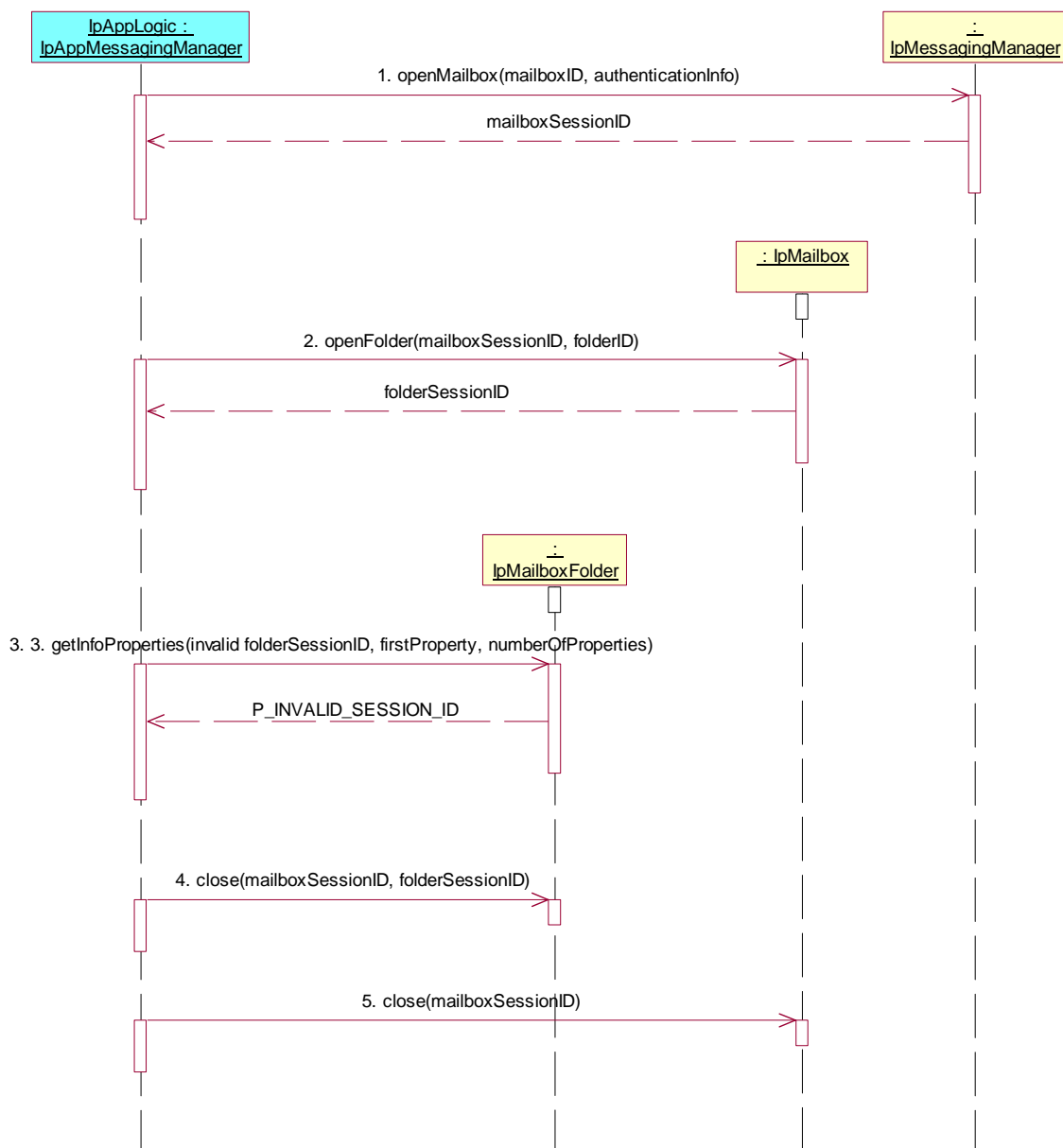
Summary: getInfoProperties, P_INVALID_SESSION_ID.

Reference: ES 202 915-9 [1], clause 8.5.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call **openMailbox()** on IpMessagingManager
Parameters: mailboxID, authenticationInfo
Check: valid value of TpMailboxIdentifier is returned
2. Method call **openFolder()** on IpMailbox
Parameters: mailboxSessionID given in 1., folderID
Check: valid value of TpMailboxFolderIdentifier is returned
3. Method call **getInfoProperties()** on IpMailboxFolder
Parameters: invalid folderSessionID, firstProperty, numberOfProperties
Check: P_INVALID_SESSION_ID is returned
4. Method call **close()** on IpMailboxFolder
Parameters: mailboxSessionID given in 1., folderSessionID given in 2.
Check: no exception is returned
5. Method call **close()** on IpMailbox
Parameters: mailboxSessionID given in 1.
Check: no exception is returned

**Test GM_26**

Summary: getInfoProperties, P_GMS_NUMBER_NOT_POSITIVE.

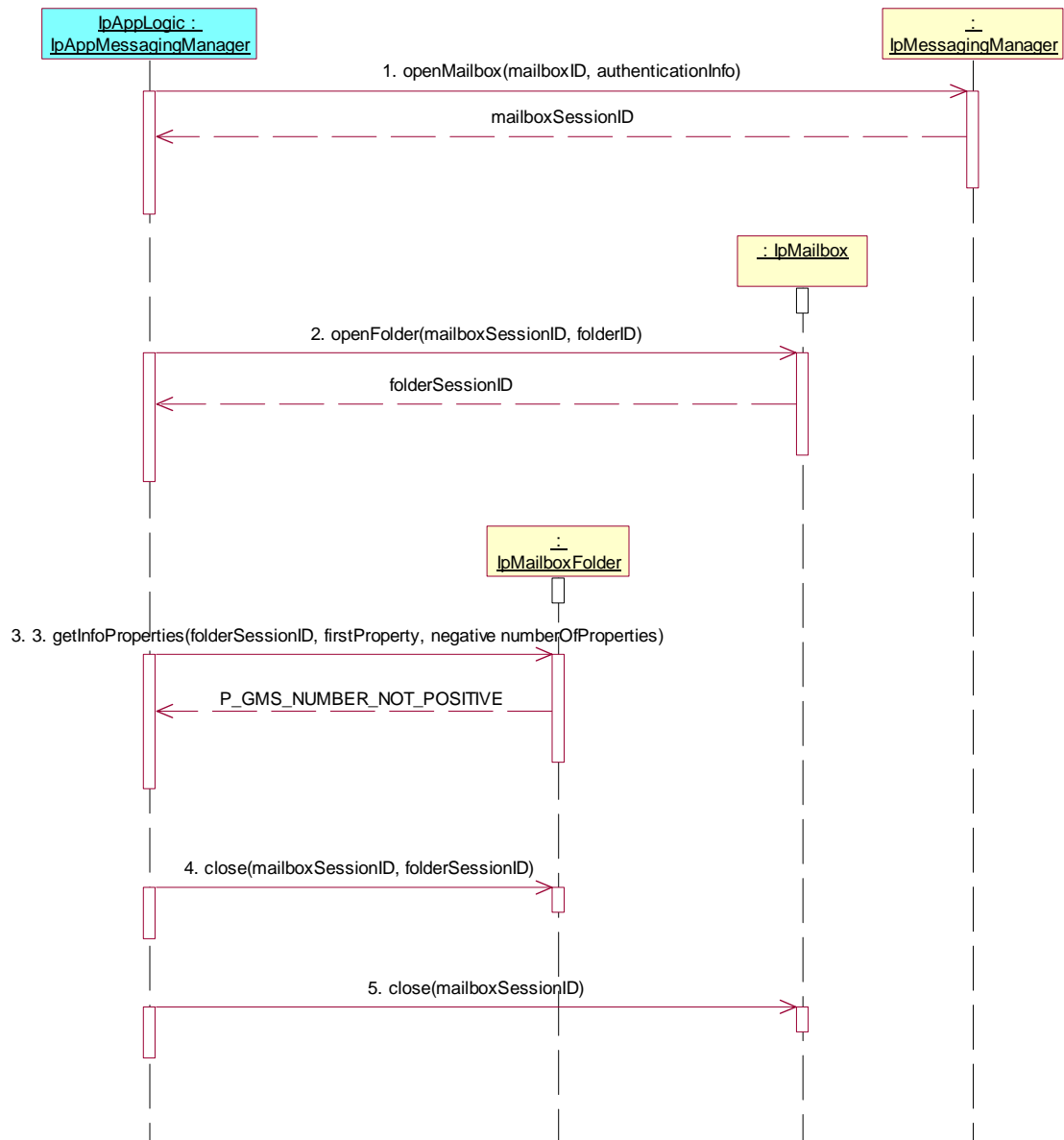
Reference: ES 202 915-9 [1], clause 8.5.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

- Method call **openMailbox()** on IpMessagingManager
 Parameters: mailboxID, authenticationInfo
 Check: valid value of TpMailboxIdentifier is returned
- Method call **openFolder()** on IpMailbox
 Parameters: mailboxSessionID given in 1., folderID
 Check: valid value of TpMailboxFolderIdentifier is returned

3. Method call **getInfoProperties()** on IpMailboxFolder
Parameters: folderSessionID given in 2, firstProperty, negative numberOfProperties
Check: P_GMS_NUMBER_NOT_POSITIVE is returned
4. Method call **close()** on IpMailboxFolder
Parameters: mailboxSessionID given in 1., folderSessionID given in 2.
Check: no exception is returned
5. Method call **close()** on IpMailbox
Parameters: mailboxSessionID given in 1.
Check: no exception is returned



Test GM_27

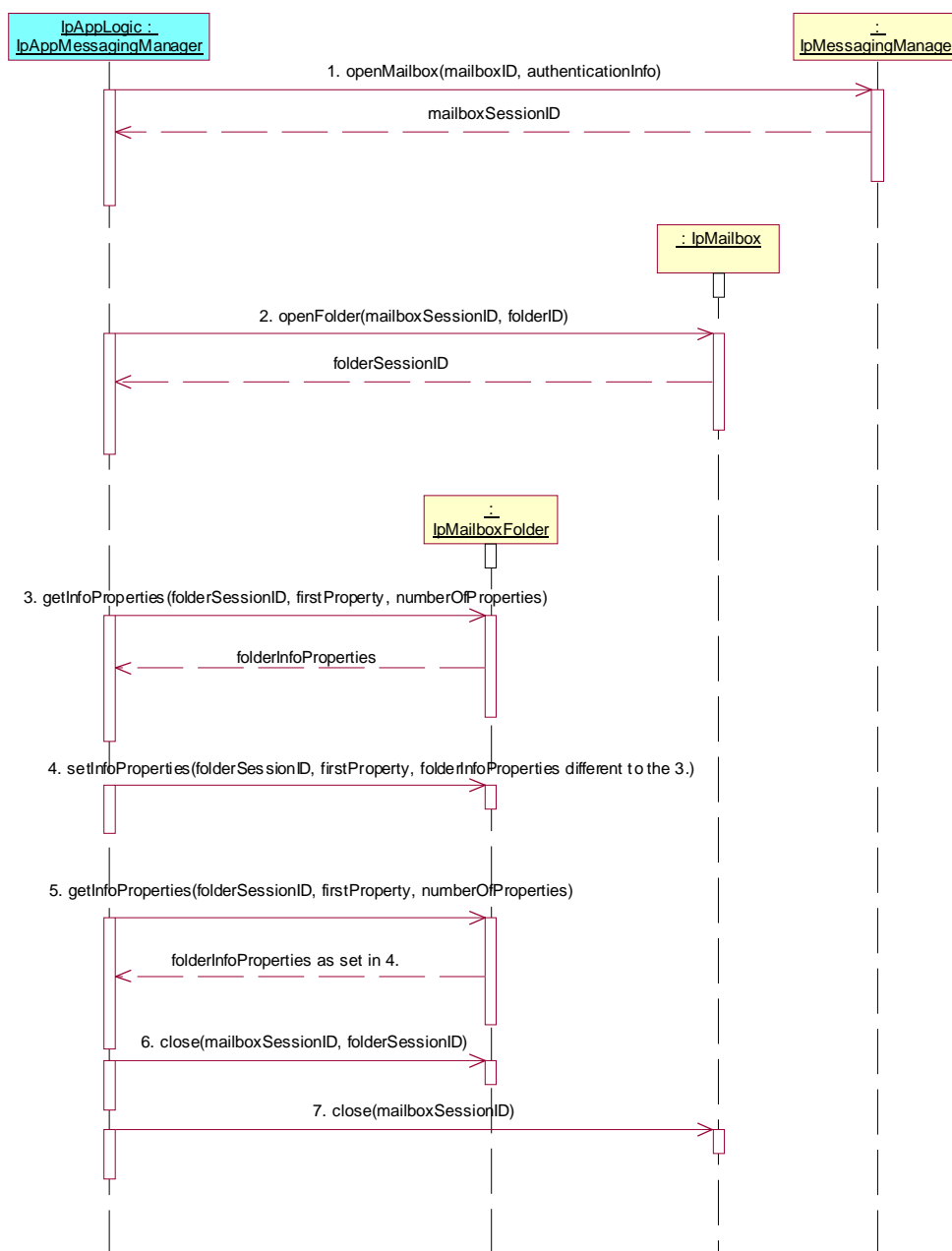
Summary: IpMailboxFolder, all methods, successful.

Reference: ES 202 915-9 [1], clause 8.5.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call **openMailbox()** on IpMessagingManager
Parameters: mailboxID, authenticationInfo
Check: valid value of TpMailboxIdentifier is returned
2. Method call **openFolder()** on IpMailbox
Parameters: mailboxSessionID given in 1., folderID
Check: valid value of TpMailboxFolderIdentifier is returned
3. Method call **getInfoProperties()** on IpMailboxFolder
Parameters: folderSessionID given in 2., firstProperty, numberOfProperties
Check: valid value of TpFolderInfoPropertySet is returned
4. Method call **setInfoProperties()**
Parameters: folderSessionID given in 2., firstProperty, folderInfoProperties different from TpFolderInfoPropertySet returned in 3.
Check: no exception is returned
5. Method call **getInfoProperties()**
Parameters: folderSessionID given in 2., firstProperty given in 4., numberOfProperties
Check: valid value of TpFolderInfoPropertySet is returned with values given in 4.
6. Method call **close()** on IpMailboxFolder
Parameters: mailboxSessionID given in 1., folderSessionID given in 2.
Check: no exception is returned
7. Method call **close()** on IpMailbox
Parameters: mailboxSessionID given in 1.
Check: no exception is returned



Test GM_28

Summary: setInfoProperties, P_INVALID_SESSION_ID.

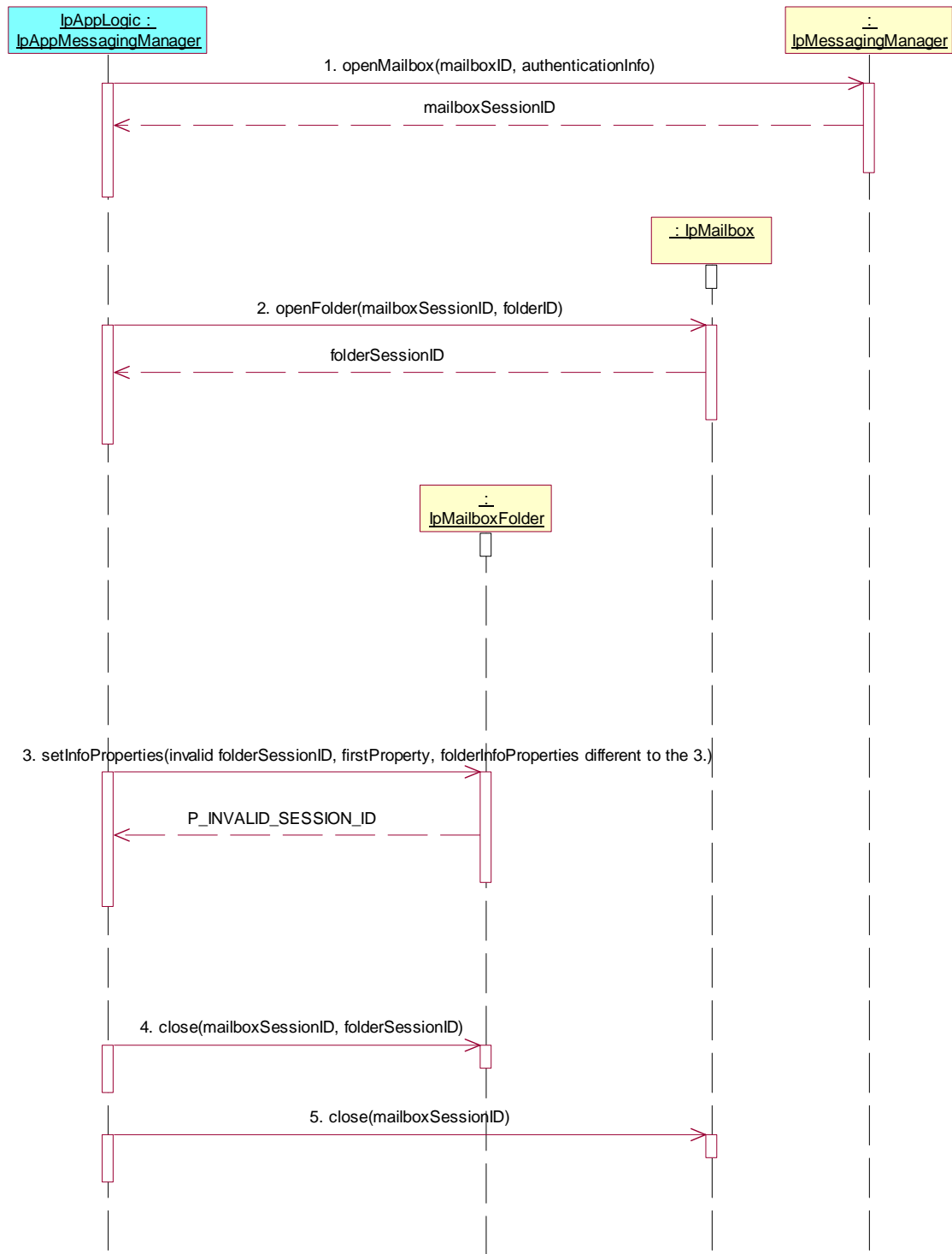
Reference: ES 202 915-9 [1], clause 8.5.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call **openMailbox()** on IpMessagingManager
Parameters: mailboxID, authenticationInfo
Check: valid value of TpMailboxIdentifier is returned
2. Method call **openFolder()** on IpMailbox
Parameters: mailboxSessionID given in 1., folderID
Check: valid value of TpMailboxFolderIdentifier is returned

3. Method call **setInfoProperties()** on IpMailboxFolder
Parameters: invalid folderSessionID, firstProperty, folderInfoProperties
Check: P_INVALID_SESSION_ID exception is returned
4. Method call **close()** on IpMailboxFolder
Parameters: mailboxSessionID given in 1., folderSessionID given in 2.
Check: no exception is returned
5. Method call **close()** on IpMailbox
Parameters: mailboxSessionID given in 1.
Check: no exception is returned



Test GM_29

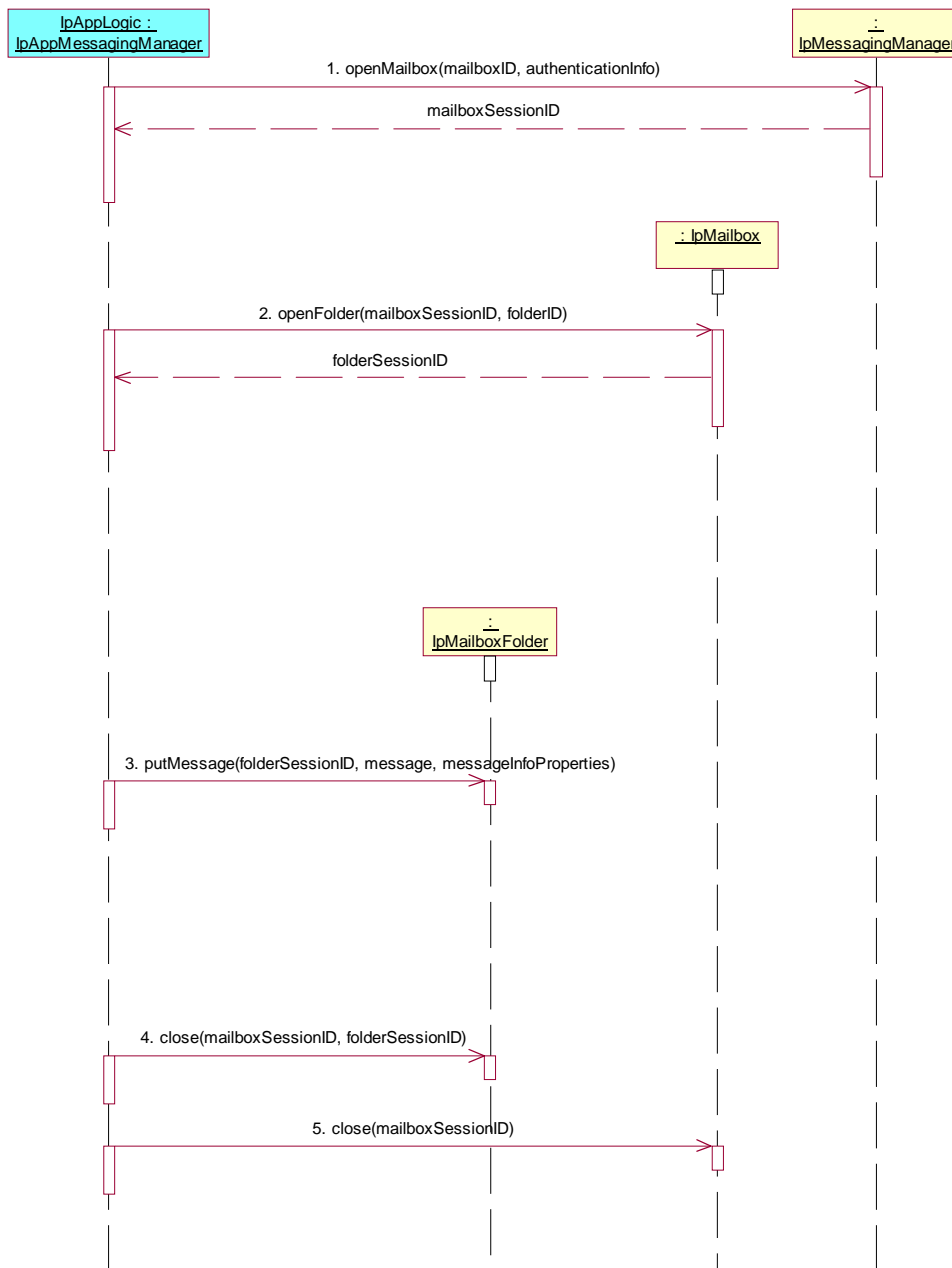
Summary: IpMailboxFolder, all methods, successful.

Reference: ES 202 915-9 [1], clause 8.1.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call **openMailbox()** on IpMessagingManager
Parameters: mailboxID, authenticationInfo
Check: valid value of TpMailboxIdentifier is returned
2. Method call **openFolder()** on IpMailbox
Parameters: mailboxSessionID given in 1., folderID
Check: valid value of TpMailboxFolderIdentifier is returned
3. Method call **putMessage()** on IpMailboxFolder
Parameters: folderSessionID given in 2., message, messageInfoProperties
Check: no exception is returned
4. Method call **close()** on IpMailboxFolder
Parameters: mailboxSessionID given in 1., folderSessionID given in 2.
Check: no exception is returned
5. Method call **close()** on IpMailbox
Parameters: mailboxSessionID given in 1.
Check: no exception is returned



Test GM_30

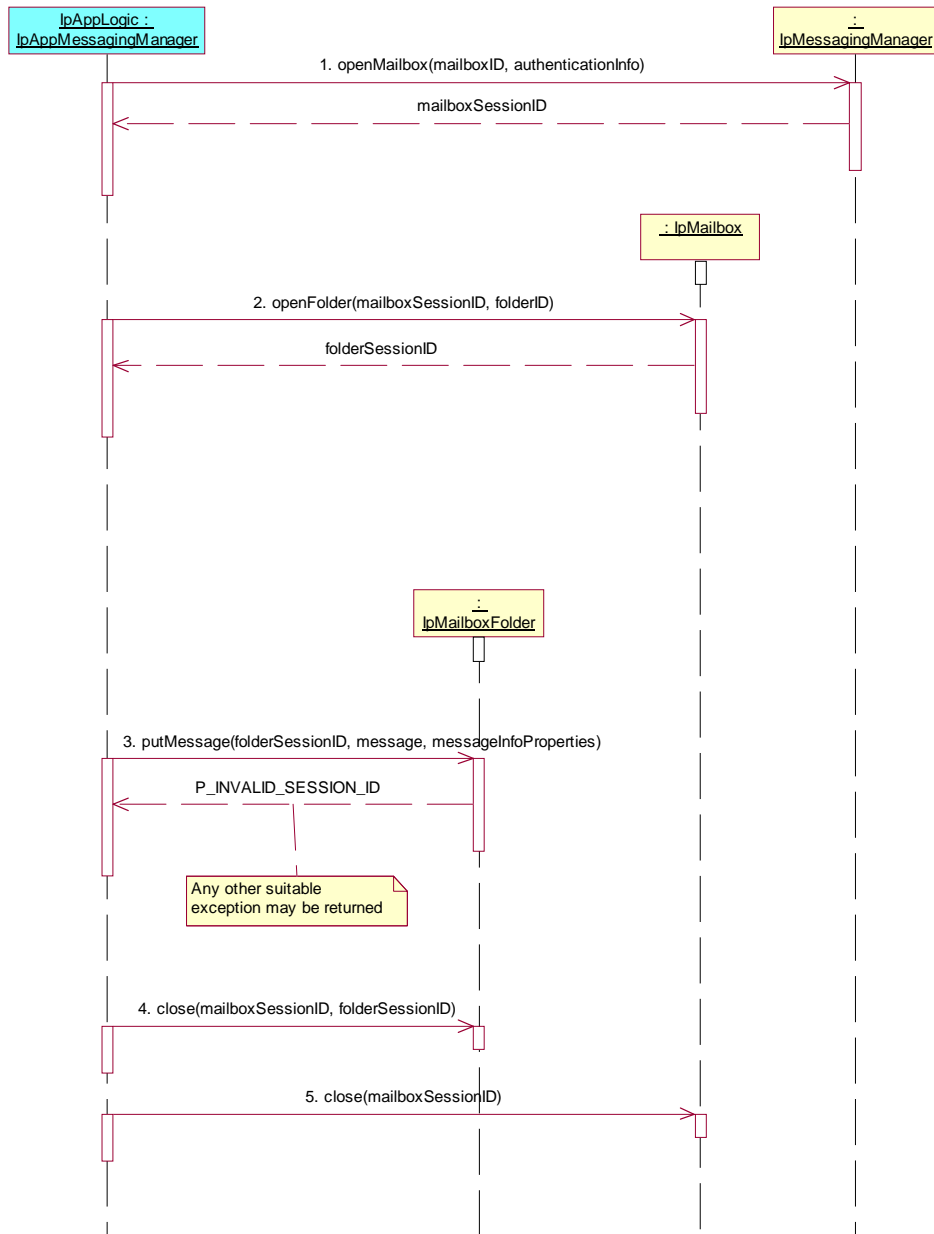
Summary: putMessage, P_INVALID_SESSION_ID.

Reference: ES 202 915-9 [1], clause 8.4.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call **openMailbox()** on IpMessagingManager
Parameters: mailboxID, authenticationInfo
Check: valid value of TpMailboxIdentifier is returned
2. Method call **openFolder()** on IpMailbox
Parameters: mailboxSessionID given in 1., folderID
Check: valid value of TpMailboxFolderIdentifier is returned
3. Method call **putMessage()** on IpMailboxFolder
Parameters: invalid folderSessionID, message, messageInfoProperties
Check: P_INVALID_SESSION_ID is returned
4. Method call **close()** on IpMailboxFolder
Parameters: mailboxSessionID given in 1., folderSessionID given in 2.
Check: no exception is returned
5. Method call **close()** on IpMailbox
Parameters: mailboxSessionID given in 1.
Check: no exception is returned



Test GM_31

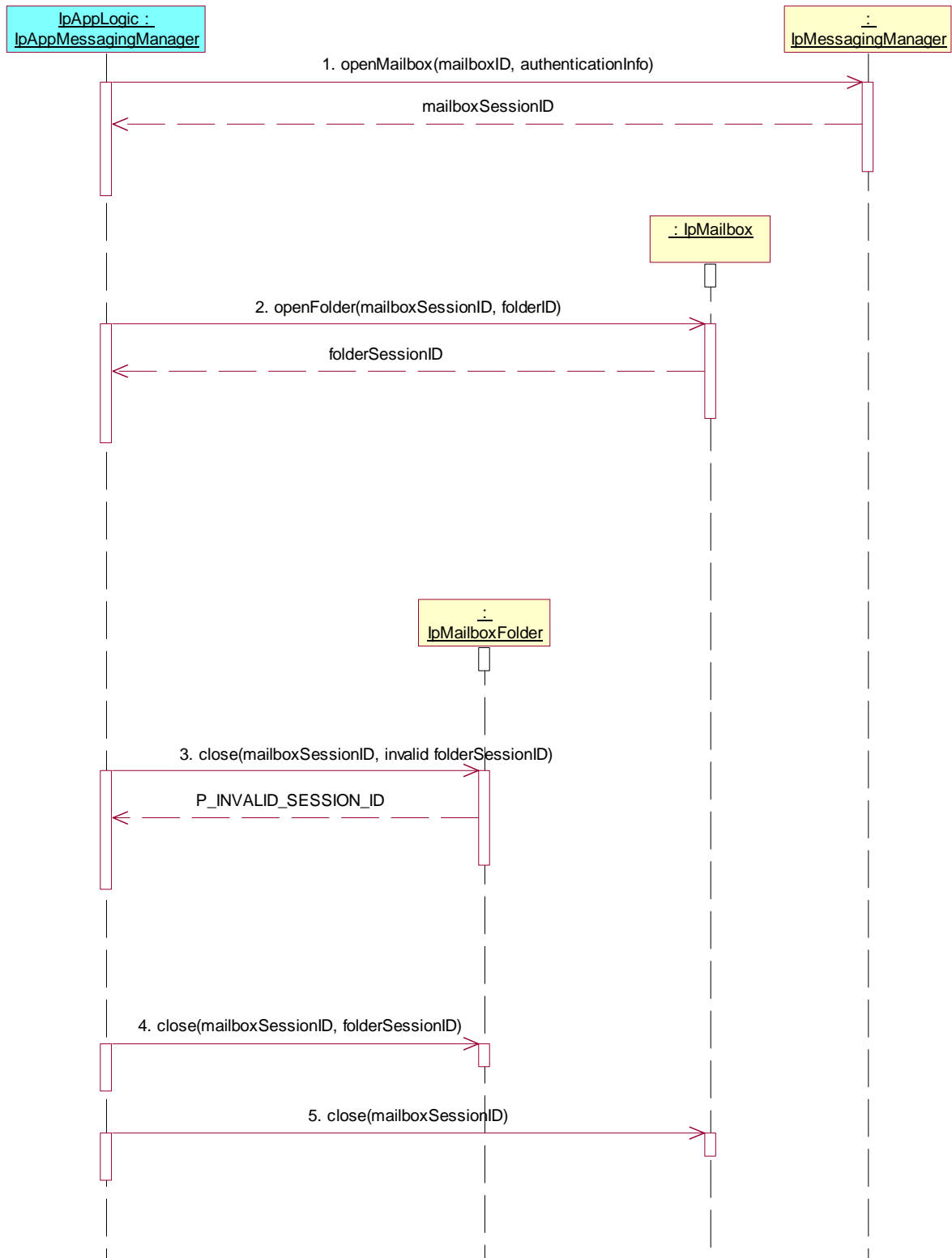
Summary: close on IpMailboxFolder, P_INVALID_SESSION_ID.

Reference: ES 202 915-9 [1], clause 8.4.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call **openMailbox()** on IpMessagingManager
Parameters: mailboxID, authenticationInfo
Check: valid value of TpMailboxIdentifier is returned
2. Method call **openFolder()** on IpMailbox
Parameters: mailboxSessionID given in 1., folderID
Check: valid value of TpMailboxFolderIdentifier is returned
3. Method call **close()** on IpMailboxFolder
Parameters: invalid mailboxSessionID, folderSessionID
Check: P_INVALID_SESSION_ID exception is returned
4. Method call **close()** on IpMailboxFolder
Parameters: mailboxSessionID given in 1., folderSessionID given in 2.
Check: no exception is returned
5. Method call **close()** on IpMailbox
Parameters: mailboxSessionID given in 1.
Check: no exception is returned



5.2.2 Generic Messaging, application side

In performing the tests for the application, it may be necessary to permit the application to perform any valid set of method exchanges with the tester in between the triggered actions or method calls indicated in the tests below. The tester shall respond to the application's method calls in conformance to the API specification and as required by the application. The requirements of the application should be made known to the tester's operator in advance.

Test GM_APP_01

Summary: IpAppMessagingManager: mailboxTerminated methods, successful.

Reference: ES 202 915-9 [1], clause 8.2.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Selection: mailboxTerminated method supported - PICS item: [2] AMM1

Test Sequence:

1. Triggered action: cause IUT to call **openMailbox()** method on the tester's (SCF) **IpMessagingManager** interface.

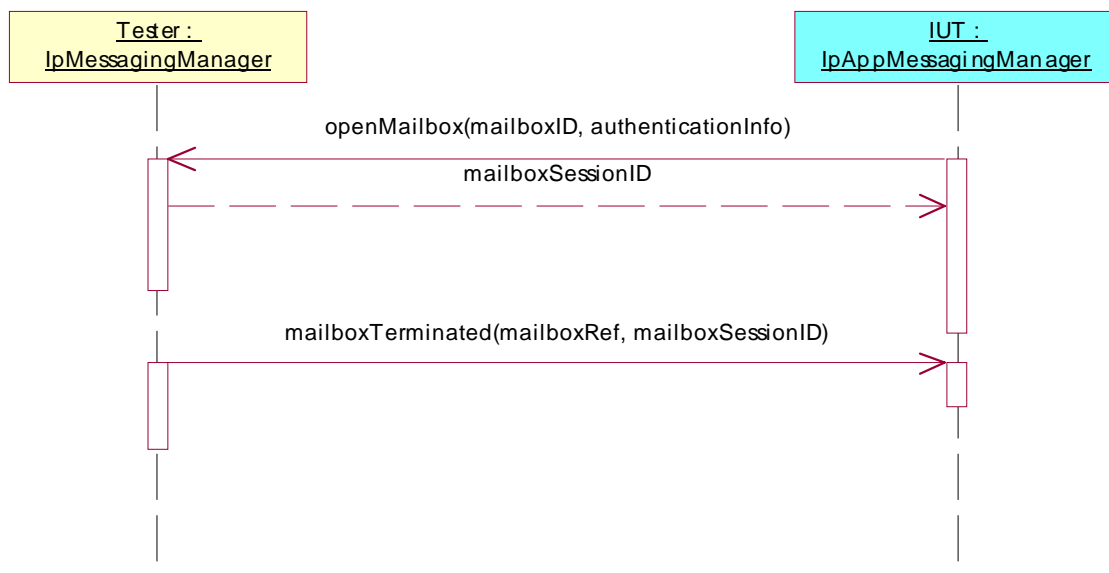
Check: valid value of mailboxID and authenticationInfo is transmitted,

Return: valid value of TpMailboxIdentifier is returned

2. Method call **mailboxTerminated()**

Parameters: mailbox and mailboxSessionID as given in 1. is transmitted.

Check: no exception is returned

**Test GM_APP_02**

Summary: IpAppMessagingManager: mailboxFaultDetected methods, successful.

Reference: ES 202 915-9 [1], clause 8.2.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Selection: mailboxFaultDetected method supported - PICS item: [2] AMM2

Test Sequence:

1. Triggered action: cause IUT to call **openMailbox()** method on the tester's (SCF) **IpMessagingManager** interface.

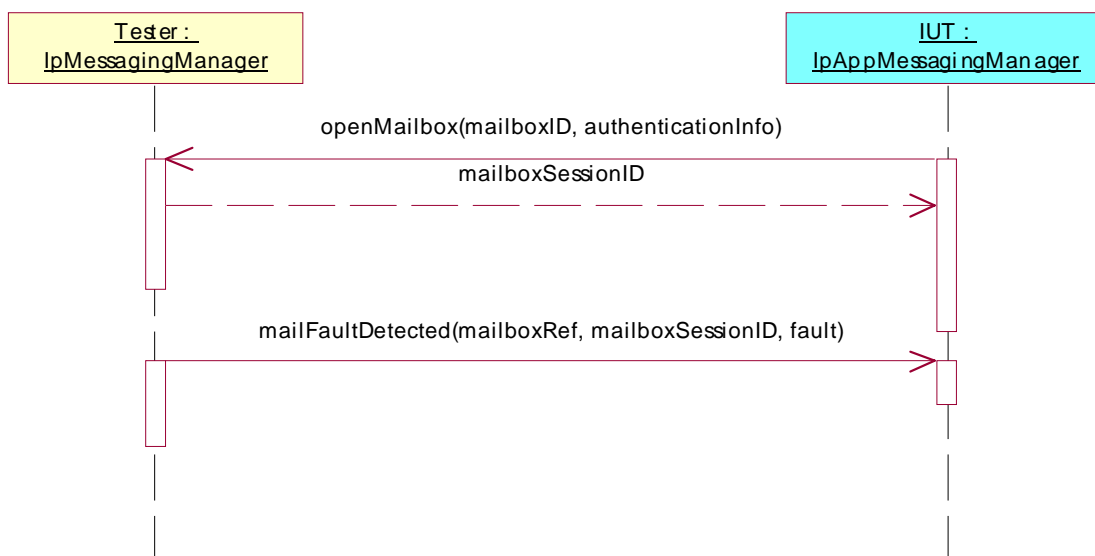
Check: valid value of mailboxID and authenticationInfo is transmitted,

Return: valid value of TpMailboxIdentifier is returned

2. Method call **mailboxFaultDetected()**

Parameters: mailbox and mailboxSessionID as given in 1. and fault is transmitted.

Check: no exception is returned

**Test GM_APP_03**

Summary: IpAppMessagingManager: messagingEventNotify methods, successful.

Reference: ES 202 915-9 [1], clause 8.2.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Selection: messagingEventNotify method supported - PICS item: [2] AMM3

Test Sequence:

1. Triggered action: cause IUT to call **openMailbox()** method on the tester's (SCF) **IpMessagingManager** interface.

Check: valid value of mailboxID and authenticationInfo is transmitted,

Return: valid value of TpMailboxIdentifier is returned

2. Triggered action: cause IUT to call **enableMessagingNotification()** method on the tester's (SCF) **IpMessagingManager** interface.

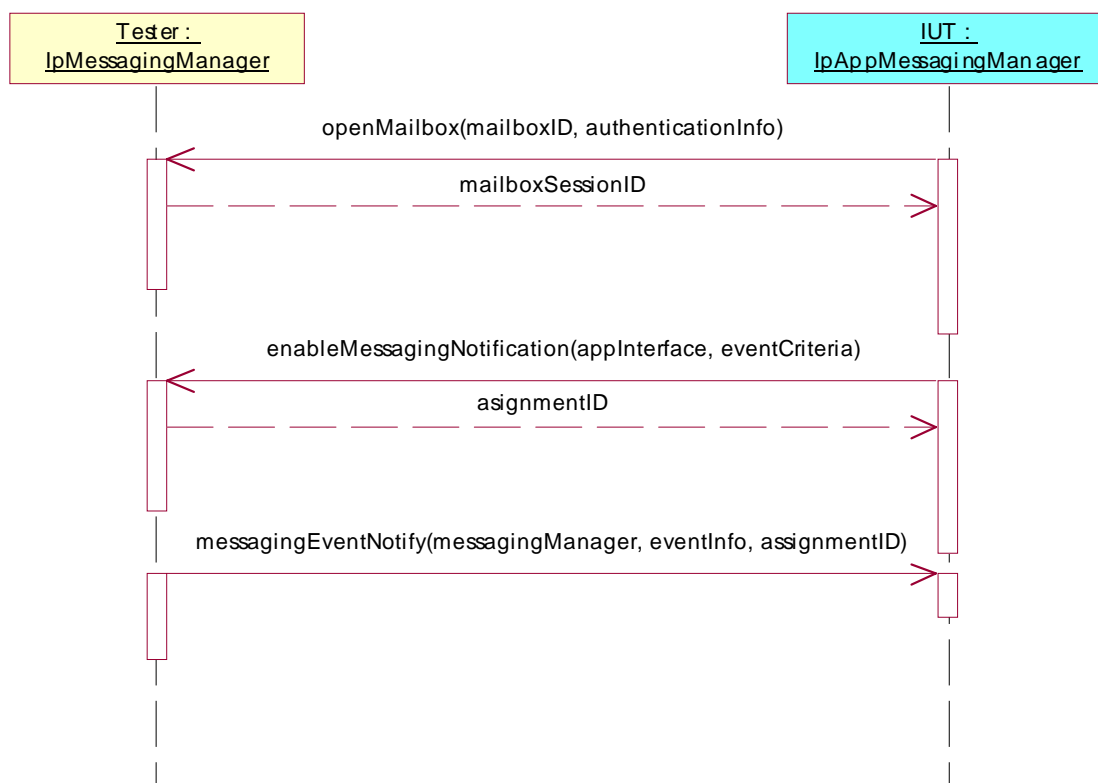
Check: valid value of appInterface and eventCriteria is transmitted,

Return: valid value of TpAssignmentID is returned

3. Method call **messagingEventNotify()**

Parameters: messagingManager, eventInfo and assignmentID as given in 2. is transmitted.

Check: no exception is returned



Test GM_APP_04

Summary: IpAppMessagingManager: messagingNotificationTerminated methods, successful.

Reference: ES 202 915-9 [1], clause 8.2.

Preamble: Registration of the IUT (User Location SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpMessagingManager interface through selecting that service and signing the required service agreement.

Selection: messagingNotificationTerminated method supported - PICS item: [2] AMM4

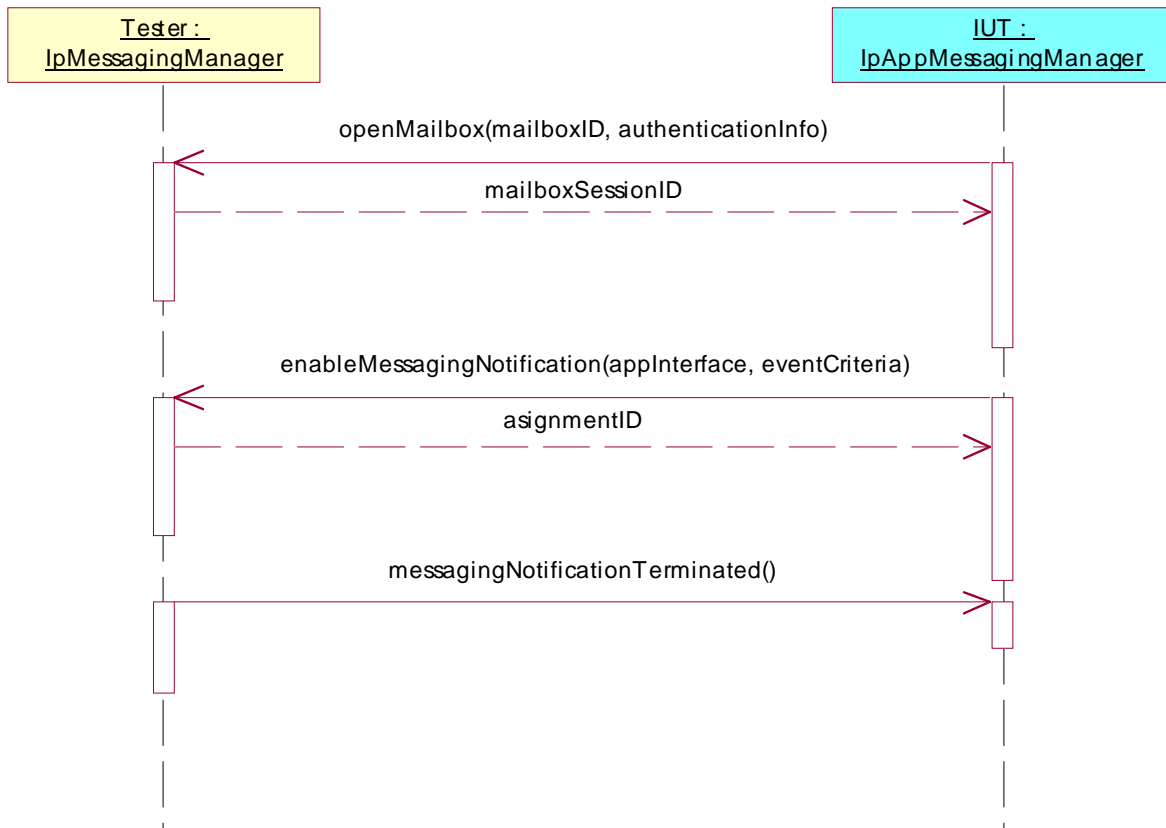
Test Sequence:

1. Triggered action: cause IUT to call **openMailbox()** method on the tester's (SCF) **IpMessagingManager** interface.

Check: valid value of mailboxID and authenticationInfo is transmitted,
Return: valid value of TpMailboxIdentifier is returned
2. Triggered action: cause IUT to call **enableMessagingNotification()** method on the tester's (SCF) **IpMessagingManager** interface.

Check: valid value of appInterface and eventCriteria is transmitted,
Return: valid value of TpAssignmentID is returned
3. Method call **messagingNotificationTerminated()**

Parameters: none.
Check: no exception is returned



History

Document history		
V1.1.1	January 2005	Membership Approval Procedure MV 20050311: 2005-01-11 to 2005-03-11
V1.1.1	March 2005	Publication