

Open Service Access (OSA); Application Programming Interface (API); Part 3: Framework (Parlay 5)



iTeh STANDARD PREVIEW
(standards.iteh.ai)

Full standard:
<https://standards.iteh.ai/catalog/standards/sist/229e1f97-1c83-44c5-b1a2-98a2b70fc86a/etsi-es-203-915-3-v1.3.1-2008-04>



Reference

RES/TISPAN-01055-03-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	14
Foreword.....	14
1 Scope	15
2 References	15
3 Definitions and abbreviations.....	15
3.1 Definitions	15
3.2 Abbreviations	15
4 Overview of the Framework.....	16
5 The Base Interface Specification.....	17
5.1 Interface Specification Format	17
5.1.1 Interface Class	17
5.1.2 Method descriptions.....	18
5.1.3 Parameter descriptions.....	18
5.1.4 State Model.....	18
5.2 Base Interface.....	18
5.2.1 Interface Class IpInterface	18
5.3 Service Interfaces	18
5.3.1 Overview	18
5.4 Generic Service Interface	19
5.4.1 Interface Class IpService	19
5.4.1.1 Method setCallback()	19
5.4.1.2 Method setCallbackWithSessionID().....	19
6 Framework Access Session API.....	20
6.1 Sequence Diagrams	20
6.1.1 Trust and Security Management Sequence Diagrams	20
6.1.1.1 Initial Access.....	20
6.1.1.2 Framework Terminates Access	22
6.1.1.3 Application Terminates Access.....	23
6.1.1.4 Non-API level Authentication.....	23
6.1.1.5 API Level Authentication	24
6.2 Class Diagrams.....	26
6.3 Interface Classes.....	27
6.3.1 Trust and Security Management Interface Classes	27
6.3.1.1 Interface Class IpClientAPILevelAuthentication.....	27
6.3.1.1.1 Method <<deprecated>> authenticate().....	27
6.3.1.1.2 Method abortAuthentication()	28
6.3.1.1.3 Method authenticationSucceeded()	28
6.3.1.1.4 Method challenge().....	28
6.3.1.2 Interface Class IpClientAccess.....	29
6.3.1.2.1 Method terminateAccess().....	30
6.3.1.3 Interface Class IpInitial	30
6.3.1.3.1 Method <<deprecated>> initiateAuthentication()	31
6.3.1.3.2 Method initiateAuthenticationWithVersion().....	32
6.3.1.4 Interface Class IpAuthentication	33
6.3.1.4.1 Method requestAccess()	33
6.3.1.5 Interface Class IpAPILevelAuthentication	34
6.3.1.5.1 Method <<deprecated>> selectEncryptionMethod().....	34
6.3.1.5.2 Method <<deprecated>> authenticate().....	35
6.3.1.5.3 Method abortAuthentication()	36
6.3.1.5.4 Method authenticationSucceeded()	36
6.3.1.5.5 Method selectAuthenticationMechanism().....	36
6.3.1.5.6 Method challenge().....	37
6.3.1.6 Interface Class IpAccess	38

6.3.1.6.1	Method obtainInterface()	38
6.3.1.6.2	Method obtainInterfaceWithCallback()	39
6.3.1.6.3	Method <<deprecated>> endAccess()	39
6.3.1.6.4	Method listInterfaces()	40
6.3.1.6.5	Method <<deprecated>> releaseInterface()	40
6.3.1.6.6	Method selectSigningAlgorithm()	40
6.3.1.6.7	Method terminateAccess()	41
6.3.1.6.8	Method relinquishInterface()	41
6.4	State Transition Diagrams	42
6.4.1	Trust and Security Management State Transition Diagrams	42
6.4.1.1	State Transition Diagrams for IpInitial	42
6.4.1.2	State Transition Diagrams for IpAPILevelAuthentication	43
6.4.1.2.1	Idle State	43
6.4.1.2.2	Authenticating Framework State	44
6.4.1.2.3	Framework Authenticated State	44
6.4.1.2.4	Authenticating Client State	44
6.4.1.2.5	Client Authenticated State	44
6.4.1.2.6	Idle State	45
6.4.1.2.7	Authenticating Framework State	46
6.4.1.2.8	Framework Authenticated State	46
6.4.1.2.9	Authenticating Client State	46
6.4.1.2.10	Client Authenticated State	46
6.4.1.2.11	Idle State	47
6.4.1.2.12	Authenticating Framework State	47
6.4.1.2.13	Framework Authenticated State	48
6.4.1.2.14	Authenticating Client State	48
6.4.1.2.15	Client Authenticated State	48
6.4.1.2.16	Idle State	49
6.4.1.2.17	Authenticating Framework State	49
6.4.1.2.18	Framework Authenticated State	50
6.4.1.2.19	Authenticating Client State	50
6.4.1.2.20	Client Authenticated State	50
6.4.1.3	State Transition Diagrams for IpAccess	51
6.4.1.3.1	Active State	51
7	Framework-to-Application API	52
7.1	Sequence Diagrams	52
7.1.1	Event Notification Sequence Diagrams	52
7.1.1.1	Enable Event Notification	52
7.1.2	Integrity Management Sequence Diagrams	53
7.1.2.1	Load Management: Suspend/resume notification from application	53
7.1.2.2	Load Management: Framework queries load statistics	54
7.1.2.3	Load Management: Framework callback registration and Application load control	55
7.1.2.4	Load Management: Application reports current load condition	56
7.1.2.5	Load Management: Application queries load statistics	56
7.1.2.6	Load Management: Application callback registration and load control	57
7.1.2.7	Heartbeat Management: Start/perform/end heartbeat supervision of the application	58
7.1.2.8	Fault Management: Framework detects a Service failure	59
7.1.2.9	Fault Management: Application requests a Framework activity test	60
7.1.3	Service Agreement Management Sequence Diagrams	60
7.1.3.1	Service Selection	60
7.1.4	Service Discovery Sequence Diagrams	62
7.1.4.1	Service Discovery	62
7.2	Class Diagrams	65
7.3	Interface Classes	67
7.3.1	Service Discovery Interface Classes	67
7.3.1.1	Interface Class IpServiceDiscovery	67
7.3.1.1.1	Method listServiceTypes()	68
7.3.1.1.3	Method discoverService()	69
7.3.1.1.4	Method listSubscribedServices()	70
7.3.2	Service Agreement Management Interface Classes	70
7.3.2.1	Interface Class IpAppServiceAgreementManagement	70

7.3.2.1.1	Method signServiceAgreement().....	71
7.3.2.1.2	Method terminateServiceAgreement()	71
7.3.2.2	Interface Class IpServiceAgreementManagement	72
7.3.2.2.1	Method signServiceAgreement().....	72
7.3.2.2.2	Method terminateServiceAgreement()	73
7.3.2.2.3	Method selectService().....	74
7.3.2.2.4	Method initiateSignServiceAgreement()	74
7.3.3	Integrity Management Interface Classes	75
7.3.3.1	Interface Class IpAppFaultManager	75
7.3.3.1.1	Method activityTestRes()	76
7.3.3.1.2	Method appActivityTestReq()	76
7.3.3.1.3	Method <<deprecated>> fwFaultReportInd()	76
7.3.3.1.4	Method <<deprecated>> fwFaultRecoveryInd()	76
7.3.3.1.5	Method <<deprecated>> svcUnavailableInd()	76
7.3.3.1.6	Method <<deprecated>> genFaultStatsRecordRes().....	77
7.3.3.1.7	Method <<deprecated>> fwUnavailableInd()	77
7.3.3.1.8	Method activityTestErr()	77
7.3.3.1.9	Method <<deprecated>> genFaultStatsRecordErr().....	77
7.3.3.1.10	Method appUnavailableInd().....	78
7.3.3.1.11	Method <<deprecated>> genFaultStatsRecordReq()	78
7.3.3.1.12	Method svcAvailStatusInd()	78
7.3.3.1.13	Method <<new>> generateFaultStatisticsRecordRes()	78
7.3.3.1.14	Method <<new>> generateFaultStatisticsRecordErr().....	79
7.3.3.1.15	Method <<new>> generateFaultStatisticsRecordReq().....	79
7.3.3.1.16	Method <<new>> fwAvailStatusInd()	79
7.3.3.2	Interface Class IpFaultManager	80
7.3.3.2.1	Method activityTestReq().....	81
7.3.3.2.2	Method appActivityTestRes()	81
7.3.3.2.3	Method svcUnavailableInd().....	81
7.3.3.2.4	Method <<deprecated>> genFaultStatsRecordReq()	82
7.3.3.2.5	Method appActivityTestErr()	82
7.3.3.2.6	Method <<deprecated>> appUnavailableInd().....	82
7.3.3.2.7	Method <<deprecated>> genFaultStatsRecordRes().....	83
7.3.3.2.8	Method <<deprecated>> genFaultStatsRecordErr().....	83
7.3.3.2.9	Method appAvailStatusInd()	83
7.3.3.2.10	Method <<new>> generateFaultStatisticsRecordReq().....	84
7.3.3.2.11	Method <<new>> generateFaultStatisticsRecordRes()	84
7.3.3.2.12	Method <<new>> generateFaultStatisticsRecordErr().....	85
7.3.3.3	Interface Class IpAppHeartBeatMgmt.....	85
7.3.3.3.1	Method enableAppHeartBeat().....	85
7.3.3.3.2	Method disableAppHeartBeat().....	86
7.3.3.3.3	Method changeInterval()	86
7.3.3.4	Interface Class IpAppHeartBeat.....	86
7.3.3.4.1	Method pulse()	86
7.3.3.5	Interface Class IpHeartBeatMgmt.....	86
7.3.3.5.1	Method enableHeartBeat()	87
7.3.3.5.2	Method disableHeartBeat().....	87
7.3.3.5.3	Method changeInterval()	87
7.3.3.6	Interface Class IpHeartBeat	88
7.3.3.6.1	Method pulse()	88
7.3.3.7	Interface Class IpAppLoadManager	88
7.3.3.7.1	Method <<deprecated>> queryAppLoadReq()	89
7.3.3.7.2	Method <<deprecated>> queryLoadRes().....	89
7.3.3.7.3	Method <<deprecated>> queryLoadErr()	90
7.3.3.7.4	Method loadLevelNotification().....	90
7.3.3.7.5	Method resumeNotification()	90
7.3.3.7.6	Method suspendNotification()	90
7.3.3.7.7	Method createLoadLevelNotification()	90
7.3.3.7.8	Method destroyLoadLevelNotification()	91
7.3.3.7.9	Method <<new>> queryAppLoadStatsReq()	91
7.3.3.7.10	Method <<new>> queryLoadStatsRes()	91
7.3.3.7.11	Method <<new>> queryLoadStatsErr()	91

7.3.3.8	Interface Class IpLoadManager	92
7.3.3.8.1	Method reportLoad()	93
7.3.3.8.2	Method <<deprecated>> queryLoadReq()	93
7.3.3.8.3	Method <<deprecated>> queryAppLoadRes()	93
7.3.3.8.4	Method <<deprecated>> queryAppLoadErr()	94
7.3.3.8.5	Method createLoadLevelNotification()	94
7.3.3.8.6	Method destroyLoadLevelNotification()	94
7.3.3.8.7	Method resumeNotification()	95
7.3.3.8.8	Method suspendNotification()	95
7.3.3.8.9	Method <<new>> queryLoadStatsReq()	95
7.3.3.8.10	Method <<new>> queryAppLoadStatsRes()	96
7.3.3.8.11	Method <<new>> queryAppLoadStatsErr()	96
7.3.3.9	Interface Class IpOAM	97
7.3.3.9.1	Method systemDateTimeQuery()	97
7.3.3.10	Interface Class IpAppOAM	97
7.3.3.10.1	Method systemDateTimeQuery()	98
7.3.4	Event Notification Interface Classes	98
7.3.4.1	Interface Class IpAppEventNotification	98
7.3.4.1.1	Method reportNotification()	98
7.3.4.1.2	Method notificationTerminated()	98
7.3.4.2	Interface Class IpEventNotification	99
7.3.4.2.1	Method createNotification()	99
7.3.4.2.2	Method destroyNotification()	99
7.4	State Transition Diagrams	100
7.4.1	Service Discovery State Transition Diagrams	100
7.4.1.1	State Transition Diagrams for IpServiceDiscovery	100
7.4.1.1.1	Active State	100
7.4.2	Service Agreement Management State Transition Diagrams	100
7.4.3	Integrity Management State Transition Diagrams	101
7.4.3.1	State Transition Diagrams for IpLoadManager	101
7.4.3.1.1	Idle State	101
7.4.3.1.2	Notification Suspended State	101
7.4.3.1.3	Active State	102
7.4.3.2	State Transition Diagrams for LoadManagerInternal	102
7.4.3.2.1	Normal load State	102
7.4.3.2.2	Application Overload State	102
7.4.3.2.3	Internal overload State	102
7.4.3.2.4	Internal and Application Overload State	103
7.4.3.3	State Transition Diagrams for IpOAM	103
7.4.3.3.1	Active State	103
7.4.3.4	State Transition Diagrams for IpFaultManager	103
7.4.3.4.1	Framework Active State	104
7.4.3.4.2	Framework Faulty State	104
7.4.3.4.3	Framework Activity Test State	104
7.4.3.4.4	Service Activity Test State	104
7.4.4	Event Notification State Transition Diagrams	104
7.4.4.1	State Transition Diagrams for IpEventNotification	104
8	Framework-to-Enterprise Operator API	104
8.1	Sequence Diagrams	108
8.1.1	Event Notification Sequence Diagrams	108
8.1.2	Service Subscription Sequence Diagrams	108
8.1.2.1	Service Discovery and Subscription Scenario	108
8.1.2.2	Enterprise Operator and Client Application Subscription Management Sequence Diagram	110
8.2	Class Diagrams	112
8.3	Interface Classes	113
8.3.1	Event Notification Interface Classes	113
8.3.1.1	Interface Class IpClientEventNotification	113
8.3.1.1.1	Method reportNotification()	114
8.3.1.1.2	Method notificationTerminated()	114
8.3.1.2	Interface Class IpEventNotification	114
8.3.1.2.1	Method createNotification()	115

8.3.1.2.2	Method destroyNotification()	115
8.3.2	Service Subscription Interface Classes	115
8.3.2.1	Interface Class IpClientAppManagement	115
8.3.2.1.1	Method createClientApp()	116
8.3.2.1.2	Method modifyClientApp()	116
8.3.2.1.3	Method deleteClientApp()	117
8.3.2.1.4	Method createSAG()	117
8.3.2.1.5	Method modifySAG()	117
8.3.2.1.6	Method deleteSAG()	118
8.3.2.1.7	Method addSAGMembers()	118
8.3.2.1.8	Method removeSAGMembers()	118
8.3.2.1.9	Method requestConflictInfo()	119
8.3.2.2	Interface Class IpClientAppInfoQuery	119
8.3.2.2.1	Method describeClientApp()	120
8.3.2.2.2	Method listClientApps()	120
8.3.2.2.3	Method describeSAG()	121
8.3.2.2.4	Method listSAGs()	121
8.3.2.2.5	Method listSAGMembers()	121
8.3.2.2.6	Method listClientAppMembership()	122
8.3.2.3	Interface Class IpServiceProfileManagement	122
8.3.2.3.1	Method createServiceProfile()	123
8.3.2.3.2	Method modifyServiceProfile()	123
8.3.2.3.3	Method deleteServiceProfile()	123
8.3.2.3.4	Method assign()	124
8.3.2.3.5	Method deassign()	124
8.3.2.3.6	Method requestConflictInfo()	124
8.3.2.4	Interface Class IpServiceProfileInfoQuery	125
8.3.2.4.1	Method listServiceProfiles()	125
8.3.2.4.2	Method describeServiceProfile()	126
8.3.2.4.3	Method listAssignedMembers()	126
8.3.2.5	Interface Class IpServiceContractManagement	126
8.3.2.5.1	Method createServiceContract()	127
8.3.2.5.2	Method modifyServiceContract()	127
8.3.2.5.3	Method deleteServiceContract()	128
8.3.2.6	Interface Class IpServiceContractInfoQuery	128
8.3.2.6.1	Method describeServiceContract()	128
8.3.2.6.2	Method listServiceContracts()	129
8.3.2.6.3	Method listServiceProfiles()	129
8.3.2.7	Interface Class IpEntOpAccountManagement	130
8.3.2.7.1	Method modifyEntOpAccount()	130
8.3.2.7.2	Method deleteEntOpAccount()	130
8.3.2.8	Interface Class IpEntOpAccountInfoQuery	131
8.3.2.8.1	Method describeEntOpAccount()	131
8.4	State Transition Diagrams	131
8.4.1	Event Notification State Transition Diagrams	131
8.4.2	Service Subscription State Transition Diagrams	131
9	Framework-to-Service API	132
9.1	Sequence Diagrams	132
9.1.1	Service Discovery Sequence Diagrams	132
9.1.2	Service Registration Sequence Diagrams	132
9.1.2.1	New SCF Sub Type Registration	132
9.1.2.2	New SCF Registration	133
9.1.3	Service Instance Lifecycle Manager Sequence Diagrams	134
9.1.3.1	Sign Service Agreement	134
9.1.4	Integrity Management Sequence Diagrams	135
9.1.4.1	Load Management: Service callback registration and load control	135
9.1.4.2	Load Management: Framework callback registration and service load control	136
9.1.4.3	Load Management: Client and Service Load Balancing	137
9.1.4.4	Heartbeat Management: Start/perform/end heartbeat supervision of the service	138
9.1.4.5	Fault Management: Service requests Framework activity test	138
9.1.4.6	Fault Management: Service requests Application activity test	139

9.1.4.7	Fault Management: Application requests Service activity test	140
9.1.4.8	Fault Management: Application detects service is unavailable	141
9.1.5	Event Notification Sequence Diagrams	141
9.2	Class Diagrams	142
9.3	Interface Classes	144
9.3.1	Service Registration Interface Classes	144
9.3.1.1	Interface Class IpFwServiceRegistration	144
9.3.1.1.1	Method registerService()	145
9.3.1.1.2	Method announceServiceAvailability()	146
9.3.1.1.3	Method unregisterService()	147
9.3.1.1.4	Method describeService()	147
9.3.1.1.5	Method unannounceService()	147
9.3.1.1.6	Method <<new>> registerServiceSubType()	148
9.3.2	Service Instance Lifecycle Manager Interface Classes	149
9.3.2.1	Interface Class IpServiceInstanceLifecycleManager	149
9.3.2.1.1	Method createServiceManager()	149
9.3.2.1.2	Method destroyServiceManager()	150
9.3.3	Service Discovery Interface Classes	150
9.3.3.1	Interface Class IpFwServiceDiscovery	150
9.3.3.1.1	Method listServiceTypes()	151
9.3.3.1.2	Method describeServiceType()	151
9.3.3.1.3	Method discoverService()	152
9.3.3.1.4	Method listRegisteredServices()	152
9.3.4	Integrity Management Interface Classes	153
9.3.4.1	Interface Class IpFwFaultManager	153
9.3.4.1.1	Method activityTestReq()	154
9.3.4.1.2	Method svcActivityTestRes()	154
9.3.4.1.3	Method appUnavailableInd()	154
9.3.4.1.4	Method <<deprecated>> genFaultStatsRecordReq()	154
9.3.4.1.5	Method <<deprecated>> svcUnavailableInd()	155
9.3.4.1.6	Method svcActivityTestErr()	155
9.3.4.1.7	Method <<deprecated>> genFaultStatsRecordRes()	156
9.3.4.1.8	Method <<deprecated>> genFaultStatsRecordErr()	156
9.3.4.1.9	Method <<deprecated>> generateFaultStatsRecordRes()	156
9.3.4.1.10	Method <<deprecated>> generateFaultStatsRecordErr()	157
9.3.4.1.11	Method svcAvailStatusInd()	157
9.3.4.1.12	Method <<new>> generateFaultStatisticsRecordReq()	157
9.3.4.1.13	Method <<new>> generateFaultStatisticsRecordRes()	158
9.3.4.1.14	Method <<new>> generateFaultStatisticsRecordErr()	158
9.3.4.2	Interface Class IpSvcFaultManager	158
9.3.4.2.1	Method activityTestRes()	159
9.3.4.2.2	Method svcActivityTestReq()	160
9.3.4.2.3	Method <<deprecated>> fwFaultReportInd()	160
9.3.4.2.4	Method <<deprecated>> fwFaultRecoveryInd()	160
9.3.4.2.5	Method <<deprecated>> fwUnavailableInd()	160
9.3.4.2.6	Method svcUnavailableInd()	161
9.3.4.2.7	Method <<deprecated>> appUnavailableInd()	161
9.3.4.2.8	Method <<deprecated>> genFaultStatsRecordRes()	161
9.3.4.2.9	Method activityTestErr()	162
9.3.4.2.10	Method <<deprecated>> genFaultStatsRecordErr()	162
9.3.4.2.11	Method <<deprecated>> genFaultStatsRecordReq()	162
9.3.4.2.12	Method <<deprecated>> generateFaultStatsRecordReq()	163
9.3.4.2.13	Method appAvailStatusInd()	163
9.3.4.2.14	Method <<new>> generateFaultStatisticsRecordRes()	163
9.3.4.2.15	Method <<new>> generateFaultStatisticsRecordErr()	164
9.3.4.2.16	Method <<new>> generateFaultStatisticsRecordReq()	164
9.3.4.2.17	Method <<new>> fwAvailStatusInd()	165
9.3.4.3	Interface Class IpFwHeartBeatMgmt	165
9.3.4.3.1	Method enableHeartBeat()	165
9.3.4.3.2	Method disableHeartBeat()	166
9.3.4.3.3	Method changeInterval()	166
9.3.4.4	Interface Class IpFwHeartBeat	166

9.3.4.4.1	Method pulse()	166
9.3.4.5	Interface Class IpSvcHeartBeatMgmt	167
9.3.4.5.1	Method enableSvcHeartBeat()	167
9.3.4.5.2	Method disableSvcHeartBeat()	167
9.3.4.5.3	Method changeInterval()	168
9.3.4.6	Interface Class IpSvcHeartBeat	168
9.3.4.6.1	Method pulse()	168
9.3.4.7	Interface Class IpFwLoadManager	169
9.3.4.7.1	Method reportLoad()	169
9.3.4.7.2	Method <<deprecated>> queryLoadReq()	170
9.3.4.7.3	Method <<deprecated>> querySvcLoadRes()	170
9.3.4.7.4	Method <<deprecated>> querySvcLoadErr()	170
9.3.4.7.5	Method createLoadLevelNotification()	171
9.3.4.7.6	Method destroyLoadLevelNotification()	171
9.3.4.7.7	Method suspendNotification()	171
9.3.4.7.8	Method resumeNotification()	172
9.3.4.7.9	Method <<new>> queryLoadStatsReq()	172
9.3.4.7.10	Method <<new>> querySvcLoadStatsRes()	172
9.3.4.7.11	Method <<new>> querySvcLoadStatsErr()	173
9.3.4.8	Interface Class IpSvcLoadManager	173
9.3.4.8.1	Method <<deprecated>> querySvcLoadReq()	174
9.3.4.8.2	Method <<deprecated>> queryLoadRes()	174
9.3.4.8.3	Method <<deprecated>> queryLoadErr()	175
9.3.4.8.4	Method loadLevelNotification()	175
9.3.4.8.5	Method suspendNotification()	175
9.3.4.8.6	Method resumeNotification()	175
9.3.4.8.7	Method createLoadLevelNotification()	176
9.3.4.8.8	Method destroyLoadLevelNotification()	176
9.3.4.8.9	Method <<new>> querySvcLoadStatsReq()	176
9.3.4.8.10	Method <<new>> queryLoadStatsRes()	177
9.3.4.8.11	Method <<new>> queryLoadStatsErr()	177
9.3.4.9	Interface Class IpFwOAM	177
9.3.4.9.1	Method systemDateTimeQuery()	178
9.3.4.10	Interface Class IpSvcOAM	178
9.3.4.10.1	Method systemDateTimeQuery()	178
9.3.5	Event Notification Interface Classes	179
9.3.5.1	Interface Class IpFwEventNotification	179
9.3.5.1.1	Method createNotification()	179
9.3.5.1.2	Method destroyNotification()	179
9.3.5.2	Interface Class IpSvcEventNotification	180
9.3.5.2.1	Method reportNotification()	180
9.3.5.2.2	Method notificationTerminated()	180
9.4	State Transition Diagrams	181
9.4.1	Service Registration State Transition Diagrams	181
9.4.1.1	State Transition Diagrams for IpFwServiceRegistration	181
9.4.1.1.1	SCF Registered State	181
9.4.1.1.2	SCF Announced State	182
9.4.2	Service Instance Lifecycle Manager State Transition Diagrams	182
9.4.3	Service Discovery State Transition Diagrams	182
9.4.4	Integrity Management State Transition Diagrams	182
9.4.4.1	State Transition Diagrams for IpFwLoadManager	182
9.4.4.1.1	Idle State	182
9.4.4.1.2	Notification Suspended State	183
9.4.4.1.3	Active State	183
9.4.4.2	State Transition Diagrams for IpFwFaultManager	183
9.4.4.2.1	Framework Active State	183
9.4.4.2.2	Framework Activity Test State	183
9.4.4.2.3	Application Activity Test State	183
9.4.4.2.4	Framework Faulty State	184
9.4.5	Event Notification State Transition Diagrams	184
10	Service Properties	184

10.1	Service Super and Sub Types	184
10.2	Service Property Types	184
10.3	General Service Properties	186
10.3.1	Service Name	187
10.3.2	Service Version	187
10.3.3	Service ID	187
10.3.4	Service Description	187
10.3.5	Product Name	187
10.3.6	Product Version	187
10.3.7	<<deprecated>> Supported Interfaces	188
10.3.8	Operation Set	188
10.3.9	Compatible Service	188
10.3.10	Backward Compatibility Level	189
10.3.11	Migration Required	189
10.3.12	Data Migrated	190
10.3.13	Migration Date And Time	191
10.3.14	Support for Regular Expressions in Address Range	191
11	Data Definitions	191
11.1	Common Framework Data Definitions	192
11.1.1	TpClientAppID	192
11.1.2	TpClientAppIDList	192
11.1.3	TpDomainID	192
11.1.4	TpDomainIDType	192
11.1.5	TpEntOpID	192
11.1.6	TpPropertyName	192
11.1.7	TpPropertyValue	193
11.1.8	TpProperty	193
11.1.9	TpPropertyList	193
11.1.10	TpEntOpIDList	193
11.1.11	TpFwID	193
11.1.12	TpService	193
11.1.13	TpServiceList	193
11.1.14	TpServiceDescription	193
11.1.15	TpServiceID	193
11.1.16	TpServiceIDList	194
11.1.17	TpServiceInstanceID	194
11.1.18	TpServiceTypeProperty	194
11.1.19	TpServiceTypePropertyList	194
11.1.20	TpServiceTypePropertyMode	194
11.1.21	TpServicePropertyTypeName	194
11.1.22	TpServicePropertyName	194
11.1.23	TpServicePropertyNameList	194
11.1.24	TpServicePropertyValue	194
11.1.25	TpServicePropertyValueList	195
11.1.26	TpServiceProperty	195
11.1.27	TpServicePropertyList	195
11.1.28	TpServiceSupplierID	195
11.1.29	TpServiceTypeDescription	195
11.1.30	TpServiceTypeName	196
11.1.31	TpServiceTypeNameList	196
11.1.32	TpSubjectType	196
11.1.33	TpServiceTypePropertyValue	197
11.1.34	TpServiceTypePropertyValueList	197
11.2	Event Notification Data Definitions	197
11.2.1	TpFwEventName	197
11.2.2	TpFwEventCriteria	198
11.2.3	TpFwEventInfo	198
11.2.4	TpFwMigrationServiceAvailableInfo	199
11.2.5	TpMigrationAdditionalInfo	199
11.2.6	TpMigrationAdditionalInfoType	200
11.2.7	TpMigrationAdditionalInfoSet	200

11.2.8	TpFwAgreementInfo	200
11.3	Trust and Security Management Data Definitions	200
11.3.1	TpAccessType	200
11.3.2	TpAuthType	200
11.3.3	TpEncryptionCapability	201
11.3.4	TpEncryptionCapabilityList	201
11.3.5	TpEndAccessProperties	201
11.3.6	TpAuthDomain	201
11.3.7	TpInterfaceName	202
11.3.8	TpInterfaceNameList	202
11.3.9	TpServiceToken	202
11.3.10	TpSignatureAndServiceMgr	202
11.3.11	TpSigningAlgorithm	203
11.3.12	TpSigningAlgorithmCapabilityList	203
11.3.13	TpAuthMechanism	203
11.3.14	TpAuthMechanismList	203
11.4	Integrity Management Data Definitions	204
11.4.1	TpActivityTestRes	204
11.4.2	TpFaultStatsRecord	204
11.4.3	TpFaultStats	204
11.4.4	TpFaultStatisticsError	204
11.4.5	TpFaultStatsSet	204
11.4.6	TpActivityTestID	204
11.4.7	TpInterfaceFault	205
11.4.8	TpSvcUnavailReason	205
11.4.9	TpFwUnavailReason	205
11.4.10	TpLoadLevel	205
11.4.11	TpLoadThreshold	205
11.4.12	TpLoadInitVal	206
11.4.13	TpLoadPolicy	206
11.4.14	TpLoadStatistic	206
11.4.15	TpLoadStatisticList	206
11.4.16	TpLoadStatisticData	206
11.4.17	TpLoadStatisticEntityID	206
11.4.18	TpLoadStatisticEntityType	207
11.4.19	TpLoadStatisticInfo	207
11.4.20	TpLoadStatisticInfoType	207
11.4.21	TpLoadStatisticError	207
11.4.22	TpSvcAvailStatusReason	208
11.4.23	TpAppAvailStatusReason	208
11.4.24	TpLoadTestID	208
11.4.25	TpFaultStatsErrorList	209
11.4.26	TpFaultReqID	209
11.4.27	TpFwAvailStatusReason	209
11.5	Service Subscription Data Definitions	209
11.5.1	TpPropertyName	209
11.5.2	TpPropertyValue	209
11.5.3	TpProperty	209
11.5.4	TpPropertyList	209
11.5.5	TpEntOpProperties	210
11.5.6	TpEntOp	210
11.5.7	TpServiceContractID	210
11.5.8	TpServiceContractIDList	210
11.5.9	TpPersonName	210
11.5.10	TpPostalAddress	210
11.5.11	TpTelephoneNumber	210
11.5.12	TpEmail	210
11.5.13	TpHomePage	210
11.5.14	TpPersonProperties	210
11.5.15	TpPerson	211
11.5.16	TpServiceStartDate	211
11.5.17	TpServiceEndDate	211

11.5.18	TpServiceRequestor	211
11.5.19	TpBillingContact	211
11.5.20	TpServiceSubscriptionProperties	211
11.5.21	TpServiceContract	211
11.5.22	TpServiceContractDescription	212
11.5.23	TpClientAppProperties	212
11.5.24	TpClientAppDescription	212
11.5.25	TpSagID	212
11.5.26	TpSagIDList	212
11.5.27	TpSagDescription	212
11.5.28	TpSag	213
11.5.29	TpServiceProfileID	213
11.5.30	TpServiceProfileIDList	213
11.5.31	TpServiceProfile	213
11.5.32	TpServiceProfileDescription	213
11.5.33	TpSagProfilePair	214
11.5.34	TpAddSagMembersConflict	214
11.5.35	TpAddSagMembersConflictList	214
11.5.36	TpAssignSagToServiceProfileConflict	215
11.5.37	TpAssignSagToServiceProfileConflictList	215
12	Exception Classes	215
Annex A (normative):	OMG IDL Description of Framework	217
Annex B (informative):	W3C WSDL Description of Framework	218
Annex C (informative):	Java™ API Description of the Framework	219
Annex D (informative):	Contents of 3GPP OSA R6 Framework	220
Annex E (informative):	Description of the Framework for 3GPP2 cdma2000 networks	221
E.1	General Exceptions	221
E.2	Specific Exceptions	221
E.2.1	Clause 1: Scope	221
E.2.2	Clause 2: References	221
E.2.3	Clause 3: Definitions and abbreviations	221
E.2.4	Clause 4: Overview of the Framework	221
E.2.5	Clause 5: The Base Interface Specification	221
E.2.6	Clause 6: Framework Access Session API	221
E.2.7	Clause 7 Framework-to-Application Sequence Diagrams	221
E.2.8	Clause 9: Framework-to-Service API	222
E.2.9	Clause 10: Service Properties	222
E.2.10	Clause 11: Data Definitions	222
E.2.11	Clause 12: Exception Classes	222
E.2.12	Annex A (normative): OMG IDL Description of the Framework	222
E.2.13	Annex B (informative): W3C WSDL Description of the Framework	222
E.2.14	Annex C (informative): Java™ API Description of the Framework	222
Annex F (informative):	Record of changes	223
F.1	Interfaces	223
F.1.1	New	223
F.1.2	Deprecated	223
F.1.3	Removed	223
F.2	Methods	224
F.2.1	New	224
F.2.2	Deprecated	224
F.2.3	Modified	224
F.2.4	Removed	224
F.3	Data Definitions	225

F.3.1	New	225
F.3.2	Modified	225
F.3.3	Removed.....	226
F.4	Service Properties.....	226
F.4.1	New	226
F.4.2	Deprecated.....	226
F.4.3	Modified	226
F.4.4	Removed.....	227
F.5	Exceptions	227
F.5.1	New	227
F.5.2	Modified	227
F.5.3	Removed.....	227
F.6	Others	227
History	228

iTeh STANDARD PREVIEW
 (standards.iteh.ai)

Full standard:
<https://standards.iteh.ai/catalog/standards/sist/229e1f97-1c83-44c5-b1a2-98a2b70fc86a/etsi-es-203-915-3-v1.3.1-2008-04>