

**NPAC SMS**

**Interoperability Test Plan**

**Release 3.1.0**

**Final**

**Version [1.12](#)**

**Supporting NANC IIS Version 3.1.0**

**~~October~~ [November 195](#), 2001**

# Table of Contents

<b>1</b>	<b>INTRODUCTION.....</b>	<b>1-1</b>
1.1	DOCUMENT OVERVIEW.....	1-1
1.1.1	Document Structure	1-2
1.2	DOCUMENT NUMBERING STRATEGY.....	1-3
1.3	TESTING OVERVIEW.....	1-3
1.4	DOCUMENT VERSION HISTORY.....	1-4
1.4.1	ITP Version 1.7	1-4
1.4.2	ITP Version 1.8	1-4
1.4.3	Release 2.0.1	1-4
1.4.4	Release 3.0.0	1-4
1.4.5	Release 3.0.1	1-4
1.4.6	Release 3.1.0	1-4
1.5	RELATED PUBLICATIONS.....	1-4
<b>2</b>	<b>THE TESTING PROCESS.....</b>	<b>2-1</b>
2.1	INTEROPERABILITY AND REGRESSION TESTING GUIDELINES.....	2-1
2.2	TEST PHASES.....	2-1
2.3	KEY LISTS AND TUNABLE PARAMETERS.....	2-2
2.4	TEST CASE DESCRIPTION.....	2-3
2.4.1	Example	2-3
2.5	TEST CASE NUMBERING.....	2-4
2.5.1	Example	2-4
2.6	TEST LOGS.....	2-4
2.7	TEST REPORTS.....	2-5
2.8	TESTING CONSIDERATIONS.....	2-6
2.9	CONFORMANCE TO STANDARDS.....	2-6
2.10	CONNECTIVITY.....	2-6
<b>3</b>	<b>STACK-TO-STACK INTEROPERABILITY TESTING.....</b>	<b>3-1</b>
3.1	OVERVIEW.....	3-1
3.2	REQUIREMENTS FOR TESTING.....	3-1
3.2.1	General Requirements	3-1
3.3	SCOPE OF TESTING.....	3-2
3.3.1	Stack-to-Stack Testing Parameters	3-2
3.3.2	NPAC SMS Simulator SAPs	3-2
3.3.3	Communication Parameters	3-2
3.3.4	NPAC Association Information	3-3
3.3.5	Presentation Context Definition List	3-3
3.4	ASSIGNMENT OF RESPONSIBILITIES.....	3-3
3.5	DEFINITION OF TESTS.....	3-3
3.5.1	TCP/IP Layers Tests	3-3
3.5.2	Valid ACSE Tests	3-4
<b>4</b>	<b>SECURITY INTEROPERABILITY TESTING.....</b>	<b>4-1</b>
4.1	OVERVIEW.....	4-1
4.2	REQUIREMENTS FOR TESTING.....	4-1
4.2.1	LNP Access Control Attribute	4-1
4.3	SCOPE OF TESTING.....	4-2
4.3.1	CMIP User Information	4-2
4.3.2	Access Control	4-3
4.4	ASSIGNMENT OF RESPONSIBILITIES.....	4-3

4.5	DEFINITION OF TESTS.....	4-3
4.5.1	Valid Security Test	4-3
4.5.2	Invalid Security Tests	4-4
<b>5</b>	<b>MANAGED OBJECT CONFORMANCE INTEROPERABILITY TESTING.....</b>	<b>5-1</b>
5.1	OVERVIEW.....	5-1
5.2	REQUIREMENTS FOR TESTING.....	5-1
5.2.1	General Requirements	5-1
5.2.2	Order of Tests	5-1
5.2.3	Association Type	5-1
5.3	SCOPE OF TESTING.....	5-2
5.4	ASSIGNMENT OF RESPONSIBILITIES.....	5-2
5.5	DEFINITION OF TESTS.....	5-2
5.5.1	Capability Tests	5-2
5.5.2	Behavior Tests	5-4
<b>6</b>	<b>ASSOCIATION MANAGEMENT INTEROPERABILITY TESTING.....</b>	<b>6-1</b>
6.1	OVERVIEW.....	6-1
6.2	REQUIREMENTS FOR TESTING.....	6-1
6.2.1	General Requirements	6-1
6.3	SCOPE OF TESTING.....	6-1
6.4	ASSIGNMENT OF RESPONSIBILITIES.....	6-1
6.5	DEFINITION OF TESTS.....	6-1
6.5.1	Retry Same/Other Host	6-1
6.5.2	Security Violation Tests	6-1
6.5.3	Loss of Association Tests	6-2
6.5.4	NPAC SMS Down Tests	6-2
<b>7</b>	<b>APPLICATION TO APPLICATION INTEROPERABILITY TESTING.....</b>	<b>7-1</b>
7.1	OVERVIEW.....	7-1
7.2	REQUIREMENTS FOR TESTING.....	7-2
7.2.1	General Requirements	7-2
7.2.2	Order of Tests	7-2
7.3	SCOPE OF TESTING.....	7-2
7.4	ASSIGNMENT OF RESPONSIBILITIES.....	7-2
7.5	DEFINITION OF TESTS.....	7-3
7.5.1	Valid Behavior Tests	7-3
7.5.2	Inopportune Behavior Tests	7-4
<b>8</b>	<b>INTEROPERABILITY TESTING EXIT CRITERIA.....</b>	<b>8-1</b>
8.1	INTRODUCTION.....	8-1
8.2	SUT CERTIFICATION GUIDELINES.....	8-1
<b>9</b>	<b>STACK TO STACK TEST CASES.....</b>	<b>9-1</b>
9.1	TEST CASES.....	9-1
9.1.1	S2S.SOA.PING and S2S.LSMS.PING	9-1
9.1.2	S2S.SOA.FTP and S2S.LSMS.FTP	9-1
9.1.3	S2S.SOA.VAL.ASSOC and S2S.LSMS.VAL.ASSOC	9-1
9.1.4	S2S.SOA.VAL.RELES and S2S.LSMS.VAL.RELES	9-2
9.1.5	S2S.SOA.VAL.RELES.BYNPAC and S2S.LSMS.VAL.RELES.BYNPAC	9-2
9.1.6	S2S.SOA.VAL.ABORT and S2S.LSMS.VAL.ABORT	9-2
9.1.7	S2S.SOA.VAL.ABORT.BYNPAC and S2S.LSMS.VAL.ABORT.BYNPAC	9-3
<b>10</b>	<b>SECURITY TEST CASES.....</b>	<b>10-1</b>
10.1	GROUP A SECURITY TEST CASES.....	10-1
10.1.1	SEC.SOA.VAL.ASSOC.NOSIG and SEC.LSMS.VAL.ASSOC.NOSIG	10-1

10.1.2	SEC.SOA.INV.ASSOC.INVSYS and SEC.LSMS.INV.ASSOC.INVSYS	10-1
10.1.3	SEC.SOA.INV.ASSOC.INVT and SEC.LSMS.INV.ASSOC.INVT	10-2
10.1.4	SEC.SOA.INV.ASSOC.SEQ and SEC.LSMS.INV.ASSOC.SEQ	10-2
10.2	GROUP B TEST CASES.....	10-2
10.2.1	SEC.SOA.VAL.ASSOC and SEC.LSMS.VAL.ASSOC	10-3
10.2.2	SEC.SOA.INV.ASSOC.INVK and SEC.LSMS.INV.ASSOC.INVK	10-3
10.2.3	SEC.SOA.INV.ASSOC.INVSIG and SEC.LSMS.INV.ASSOC.INVSIG	10-4
10.2.4	SEC.SOA.INV.NOT.INVSIG and SEC.LSMS.INV.NOT.INVSIG	10-4
10.2.5	SEC.SOA.INV.CREATE.INVSEQ and SEC.LSMS.INV.CREATE.INVSEQ	10-4
10.2.6	SEC.SOA.INV.SET.INVSIG and SEC.LSMS.INV.SET.INVSIG	10-5
10.2.7	SEC.SOA.INV.ACTION.INVSYS and SEC.LSMS.INV.ACTION.INVSYS	10-5
10.2.8	SEC.SOA.INV.GET.INVT and SEC.LSMS.INV.GET.INVT	10-6
10.2.9	SEC.SOA.INV.DELETE.INVSIG and SEC.LSMS.INV.DELETE.INVSIG	10-6
10.2.10	SEC.SOA.INV.ASSOC.ASSOCSP.INVSYS	10-6
<b>11</b>	<b>SOA TO NPAC MOC TEST CASES.....</b>	<b>11-1</b>
11.1	LNPNPAC-SMS.....	11-1
11.1.1	MOC.SOA.CAP.OP.GET.InpNPAC-SMS	11-1
11.1.2	MOC.SOA.CAP.NOT.InpNPAC-SMS-Operational-Information	11-1
11.1.3	MOC.SOA.INV.NOT.InpNPAC-SMS-Operational-Information	11-2
11.1.4	MOC.SOA.CAP.NOT.subscriptionVersionNewNPA-NXX	11-2
11.1.5	MOC.SOA.INV.GET.InpNPAC-SMS	11-2
11.1.6	MOC.SOA.INV.NOT.subscriptionVersionNewNPA-NXX	11-2
11.1.7	MOC.SOA.CAP.ACT.InpNotificationRecovery	11-3
11.1.8	MOC.SOA.INV.ACT.InpNotificationRecovery	11-3
11.1.9	MOC.SOA.CAP.OP.ACT.InpRecoveryComplete	11-4
11.1.10	MOC.SOA.INV.ACT.InpRecoveryComplete	11-4
11.2	LNPSERVICEPROVS.....	11-4
11.2.1	MOC.SOA.CAP.OP.GET.InpServiceProvs	11-4
11.2.2	MOC.SOA.INV.GET.InpServiceProvs	11-5
11.3	LNPAUDITS.....	11-5
11.3.1	MOC.SOA.CAP.OP.GET.InpAudits	11-5
11.3.2	MOC.SOA.INV.GET.InpAudits	11-6
11.4	LNPSUBSCRIPTIONS.....	11-6
11.4.1	MOC.SOA.CAP.OP.GET.InpSubscriptions	11-6
11.4.2	MOC.SOA.CAP.ACT.subscriptionVersionNewSP-Create-Initial	11-6
11.4.3	MOC.SOA.CAP.ACT.subscriptionVersionOldSP-Create-Initial	11-7
11.4.4	MOC.SOA.CAP.ACT.subscriptionVersionNewSP-Create-Second	11-7
11.4.5	MOC.SOA.CAP.ACT.subscriptionVersionOldSP-Create-Second	11-8
11.4.6	MOC.SOA.CAP.ACT.subscriptionVersionActivate-VersionId	11-8
11.4.7	MOC.SOA.CAP.ACT.subscriptionVersionActivate-TN	11-8
11.4.8	MOC.SOA.CAP.ACT.subscriptionVersionActivate-TNRange	11-9
11.4.9	MOC.SOA.CAP.ACT.subscriptionVersionModify	11-9
11.4.10	MOC.SOA.CAP.ACT.subscriptionVersionCancel	11-10
11.4.11	MOC.SOA.CAP.ACT.subscriptionVersionOldSP-CancellationAcknowledge	11-10
11.4.12	MOC.SOA.CAP.ACT.subscriptionVersionNewSP-CancellationAcknowledge	11-11
11.4.13	MOC.SOA.CAP.ACT.subscriptionVersionDisconnect	11-11
11.4.14	MOC.SOA.CAP.ACT.subscriptionVersionRemoveFromConflict	11-12
11.4.15	MOC.SOA.INV.GET.InpSubscriptions	11-12
11.4.16	MOC.SOA.INV.ACT.subscriptionVersionNewSP-Create	11-13
11.4.17	MOC.SOA.INV.ACT.subscriptionVersionOldSP-Create	11-13
11.4.18	MOC.SOA.INV.ACT.subscriptionVersionActivate	11-13
11.4.19	MOC.SOA.INV.ACT.subscriptionVersionModify	11-14
11.4.20	MOC.SOA.INV.ACT.subscriptionVersionCancel	11-14
11.4.21	MOC.SOA.INV.ACT.subscriptionVersionOldSP-CancellationAcknowledge	11-14
11.4.22	MOC.SOA.INV.ACT.subscriptionVersionNewSP-CancellationAcknowledge	11-15

11.4.23	MOC.SOA.INV.ACT.subscriptionVersionDisconnect	11-15
11.4.24	MOC.SOA.INV.ACT.subscriptionVersionRemoveFromConflict	11-15
11.4.25	MOC.SOA.CAP.ACT.numberPoolBlockCreateAction	11-15
11.4.26	MOC.SOA.INV.ACT.numberPoolBlockCreateAction	11-16
11.4.27	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeStatusAttributeValueChange	11-16
11.4.28	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeAttributeValueChange	11-16
11.4.29	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeObjectCreation	11-17
11.4.30	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeDonorSP-CustomerDisconnectDate	11-17
11.4.31	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeCancellationAcknowledgeRequest	11-18
11.4.32	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeNewSP-CreateRequest	11-18
11.4.33	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeOldSP-ConcurrenceRequest	11-19
11.4.34	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeOldSPFinalConcurrenceWindowExpiration	11-19
11.4.35	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeNewSPFinalCreateWindowExpiration	11-19
11.4.36	MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeStatusAttributeValueChange	11-20
11.4.37	MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeAttributeValueChange	11-20
11.4.38	MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeObjectCreation	11-21
11.4.39	MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeDonorSP-CustomerDisconnectDate	11-21
11.4.40	MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeCancellationAcknowledgeRequest	11-21
11.4.41	MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeNewSP-CreateRequest	11-22
11.4.42	MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeOldSP-ConcurrenceRequest	11-22
11.4.43	MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeOldSPFinalConcurrenceWindowExpiration	11-23
11.4.44	MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeNewSPFinalCreateWindowExpiration	11-23
11.4.45	MOC.SOA.INV.NOT.subscriptionVersionRangeStatusAttributeValueChange	11-23
11.4.46	MOC.SOA.INV.NOT.subscriptionVersionRangeAttributeValueChange	11-24
11.4.47	MOC.SOA.INV.NOT.subscriptionVersionRangeObjectCreation	11-24
11.4.48	MOC.SOA.INV.NOT.subscriptionVersionRangeDonorSP-CustomerDisconnectDate	11-25
11.4.49	MOC.SOA.INV.NOT.subscriptionVersionRangeCancellationAcknowledgeRequest	11-25
11.4.50	MOC.SOA.INV.NOT.subscriptionVersionRangeNewSP-CreateRequest	11-25
11.4.51	MOC.SOA.INV.NOT.subscriptionVersionRangeOldSP-ConcurrenceRequest	11-26
11.4.52	MOC.SOA.INV.NOT.subscriptionVersionRangeOldSPFinalConcurrenceWindowExpiration	11-26
11.4.53	MOC.SOA.INV.NOT.subscriptionVersionRangeNewSPFinalCreateWindowExpiration	11-26
11.5	LNPNETWORK.....	11-27
11.5.1	MOC.SOA.CAP.OP.GET.lnpNetwork	11-27
11.5.2	MOC.SOA.INV.GET.lnpNetwork	11-27
11.5.3	MOC.SOA.CAP.ACT.lnpNetwork.lnpDownload	11-28
11.5.4	MOC.SOA.INV.ACT.lnpNetwork.lnpDownload	11-28
11.5.5	MOC.SOA.VAL.lnpDownload-NPA-NXX-X	11-28
11.6	SERVICEPROV.....	11-29
11.6.1	MOC.SOA.CAP.OP.SET.serviceProv	11-29
11.6.2	MOC.SOA.CAP.OP.GET.serviceProv	11-29
11.6.3	MOC.SOA.VAL.SET.SING.serviceProv	11-30
11.6.4	MOC.SOA.VAL.SET.SING.COND.serviceProv	11-30
11.6.5	MOC.SOA.VAL.SET.MULT.serviceProv	11-30
11.6.6	MOC.SOA.INV.SET.serviceProv	11-31

11.6.7	MOC.SOA.INV.GET.serviceProv	11-31
11.6.8	MOC.SOA.BND.MIN.SET.serviceProv	11-31
11.6.9	MOC.SOA.BND.MAX.SET.serviceProv	11-32
11.7	SUBSCRIPTIONAUDIT.....	11-32
11.7.1	MOC.SOA.CAP.OP.CRE.subscriptionAudit	11-32
11.7.2	MOC.SOA.CAP.OP.GET.subscriptionAudit	11-33
11.7.3	MOC.SOA.CAP.OP.DEL.subscriptionAudit	11-33
11.7.4	MOC.SOA.CAP.NOT.subscriptionAuditResults	11-33
11.7.5	MOC.SOA.CAP.NOT.subscriptionAudit-DiscrepancyReport	11-34
11.7.6	MOC.SOA.VAL.CRE.AUTO.subscriptionAudit	11-34
11.7.7	MOC.SOA.VAL.GET.SCOP.FILT.subscriptionAudit	11-34
11.7.8	MOC.SOA.VAL.DEL.SCOP.subscriptionAudit	11-35
11.7.9	MOC.SOA.INV.CRE.subscriptionAudit	11-35
11.7.10	MOC.SOA.INV.GET.subscriptionAudit	11-36
11.7.11	MOC.SOA.INV.DEL.subscriptionAudit	11-36
11.7.12	MOC.SOA.INV.NOT.subscriptionAuditResults	11-36
11.7.13	MOC.SOA.INV.NOT.subscriptionAudit-DiscrepancyReport	11-37
11.7.14	MOC.SOA.INV.CAP.OP.CRE.subscriptionAudit	11-37
11.8	SUBSCRIPTIONVERSIONNPAC.....	11-37
11.8.1	MOC.SOA.CAP.OP.SET.OldSP.subscriptionVersionNPAC	11-37
11.8.2	MOC.SOA.CAP.OP.SET.NewSP.subscriptionVersionNPAC	11-38
11.8.3	MOC.SOA.CAP.OP.GET.subscriptionVersionNPAC	11-38
11.8.4	MOC.SOA.CAP.NOT.subscriptionVersionOldSP-ConcurrenceRequest	11-39
11.8.5	MOC.SOA.CAP.NOT.subscriptionVersionOldSP-FinalConcurrenceWindowExpiration	11-39
11.8.6	MOC.SOA.CAP.NOT.subscriptionVersionNewSP-CreateRequest	11-40
11.8.7	MOC.SOA.CAP.NOT.subscriptionVersionCancellationAcknowledgeRequest	11-40
11.8.8	MOC.SOA.CAP.NOT.subscriptionVersionDonorSP-CustomerDisconnectDate	11-40
11.8.9	MOC.SOA.VAL.SET.SING.subscriptionVersionNPAC	11-40
11.8.10	MOC.SOA.VAL.SET.MULT.subscriptionVersionNPAC	11-41
11.8.11	MOC.SOA.VAL.GET.SCOP.subscriptionVersionNPAC	11-41
11.8.12	MOC.SOA.VAL.NOT.subscriptionVersionNewNPA-NXX	11-42
11.8.13	MOC.SOA.VAL.NOT.subscriptionVersionStatusAttributeValueChange	11-42
11.8.14	MOC.SOA.INV.SET.SING.subscriptionVersionNPAC	11-42
11.8.15	MOC.SOA.INV.GET.subscriptionVersionNPAC	11-43
11.8.16	MOC.SOA.INV.NOT.subscriptionVersionOldSp-ConcurrenceRequest	11-43
11.8.17	MOC.SOA.INV.NOT.subscriptionVersionNewSP-CreateRequest	11-43
11.8.18	MOC.SOA.INV.NOT.subscriptionVersionCancellationAcknowledgeRequest	11-44
11.8.19	MOC.SOA.INV.NOT.subscriptionVersionDonorSP-CustomerDisconnectDate	11-44
11.8.20	MOC.SOA.INV.NOT.subscriptionVersionStatusAttributeValueChange	11-44
11.8.21	MOC.SOA.INV.NOT.attributeValueChange.subscriptionVersion	11-45
11.8.22	MOC.SOA.INV.NOT.subscriptionVersionNewNPA-NXX	11-45
11.8.23	MOC.SOA.BND.GET.MAXQ.subscriptionVersionNPAC	11-45
11.8.24	MOC.SOA.INV.QUERY.SCOPE.D.subscriptionVersion	11-46
11.8.25	MOC.SOA.CAP.NOT.subscriptionVersionNewSP-FinalConcurrenceWindowExpiration	11-46
11.8.26	MOC.SOA.INV.NOT.subscriptionVersionNewSP-FinalConcurrenceWindowExpiration	11-46
11.9	SERVICEPROVNETWORK.....	11-47
11.9.1	MOC.SOA.CAP.OP.GET.serviceProvNetwork	11-47
11.9.2	MOC.SOA.INV.GET.serviceProvNetwork	11-47
11.10	SERVICEPROVNPA-NXX.....	11-48
11.10.1	MOC.SOA.CAP.OP.GET.serviceProvNPA-NXX	11-48
11.10.2	MOC.SOA.CAP.OP.DEL.serviceProvNPA-NXX	11-48
11.10.3	MOC.SOA.VAL.CRE.AUTO.serviceProvNPA-NXX	11-49
11.10.4	MOC.SOA.VAL.GET.SCOP.FILT.serviceProvNPA-NXX	11-49
11.10.5	MOC.SOA.VAL.DEL.SCOP.FILT.serviceProvNPA-NXX	11-49
11.10.6	MOC.SOA.INV.CRE.serviceProvNPA-NXX	11-50
11.10.7	MOC.SOA.INV.GET.serviceProvNPA-NXX	11-50

11.10.8	MOC.SOA.INV.DEL.serviceProvNPA-NXX	11-50
11.11	SERVICEPROVLRN.....	11-51
11.11.1	MOC.SOA.CAP.OP.GET.serviceProvLRN	11-51
11.11.2	MOC.SOA.CAP.OP.DEL.serviceProvLRN	11-51
11.11.3	MOC.SOA.VAL.CRE.AUTO.serviceProvLRN	11-51
11.11.4	MOC.SOA.VAL.GET.SCOP.FILT.serviceProvLRN	11-52
11.11.5	MOC.SOA.VAL.DEL.SCOP.FILT.serviceProvLRN	11-52
11.11.6	MOC.SOA.INV.CRE.serviceProvLRN	11-53
11.11.7	MOC.SOA.INV.GET.serviceProvLRN	11-53
11.11.8	MOC.SOA.INV.DEL.serviceProvLRN	11-53
11.12	NUMBERPOOLBLOCKNPAC.....	11-54
11.12.1	MOC.SOA.CAP.OP.GET.numberPoolBlockNPAC	11-54
11.12.2	MOC.SOA.CAP.OP.SET.numberPoolBlockNPAC	11-54
11.12.3	MOC.SOA.VAL.GET.SCOP.numberPoolBlockNPAC	11-54
11.12.4	MOC.SOA.INV.GET.numberPoolBlockNPAC	11-55
11.12.5	MOC.SOA.INV.SET.numberPoolBlockNPAC	11-55
11.12.6	MOC.SOA.INV.GET.SCOP.numberPoolBlockNPAC	11-55
11.13	SERVICEPROVNPA-NXX-X.....	11-56
11.13.1	MOC.SOA.CAP.OP.GET.serviceProvNPA-NXX-X	11-56
11.13.2	MOC.SOA.VAL.GET.SCOP.serviceProvNPA-NXX-X	11-56
11.13.3	MOC.SOA.INV.GET.serviceProvNPA-NXX-X	11-57
11.13.4	MOC.SOA.INV.GET.SCOP.serviceProvNPA-NXX-X	11-57
<b>12</b>	<b>NPAC SMS TO SOA MOC TEST CASES.....</b>	<b>12-1</b>
12.1	LNP SOA.....	12-1
12.1.1	MOC.NPAC.CAP.OP.GET.lnpSOA	12-1
12.1.2	MOC.NPAC.INV.CRE.INH.lnpSOA	12-1
12.1.3	MOC.NPAC.INV.SET.lnpSOA	12-2
12.1.4	MOC.NPAC.INV.DEL.lnpSOA	12-2
12.2	LNP NETWORK.....	12-2
12.2.1	MOC.NPAC.SOA.CAP.OP.GET.lnpNetwork	12-2
12.2.2	MOC.NPAC.SOA.INV.CRE.INH.lnpNetwork	12-3
12.2.3	MOC.NPAC.SOA.INV.SET.lnpNetwork	12-3
12.2.4	MOC.NPAC.SOA.INV.ACT.lnpNetwork	12-3
12.2.5	MOC.NPAC.SOA.INV.DEL.lnpNetwork	12-4
12.3	SERVICEPROVNETWORK.....	12-4
12.3.1	MOC.NPAC.SOA.CAP.OP.CRE.serviceProvNetwork	12-4
12.3.2	MOC.NPAC.SOA.CAP.OP.GET.serviceProvNetwork	12-4
12.3.3	MOC.NPAC.SOA.CAP.OP.SET.serviceProvNetwork	12-5
12.3.4	MOC.NPAC.SOA.CAP.OP.DEL.serviceProvNetwork	12-5
12.3.5	MOC.NPAC.SOA.INV.CRE.DUP.serviceProvNetwork	12-5
12.3.6	MOC.NPAC.SOA.INV.SET.RO.serviceProvNetwork	12-6
12.3.7	MOC.NPAC.SOA.INV.SET.SYN.serviceProvNetwork	12-6
12.3.8	MOC.NPAC.SOA.INV.SET.serviceProvNetwork	12-6
12.3.9	MOC.NPAC.SOA.INV.GET.serviceProvNetwork	12-7
12.3.10	MOC.NPAC.SOA.INV.DEL.serviceProvNetwork	12-7
12.3.11	MOC.NPAC.SOA.INV.DEL.CO.serviceProvNetwork	12-7
12.3.12	MOC.NPAC.SOA.BND.SET.MIN.serviceProvNetwork	12-8
12.3.13	MOC.NPAC.SOA.BND.SET.MAX.serviceProvNetwork	12-8
12.4	SERVICEPROVNPA-NXX.....	12-8
12.4.1	MOC.NPAC.SOA.CAP.OP.CRE.serviceProvNPA-NXX	12-9
12.4.2	MOC.NPAC.SOA.CAP.OP.DEL.serviceProvNPA-NXX	12-9
12.4.3	MOC.NPAC.SOA.INV.CRE.DUP.serviceProvNPA-NXX	12-9
12.4.4	MOC.NPAC.SOA.INV.SET.serviceProvNPA-NXX	12-10
12.4.5	MOC.NPAC.SOA.INV.DEL.serviceProvNPA-NXX	12-10
12.5	SERVICEPROVLRN.....	12-10



12.5.1	MOC.NPAC.SOA.CAP.OP.CRE.serviceProvLRN	12-11
12.5.2	MOC.NPAC.SOA.CAP.OP.DEL.serviceProvLRN	12-11
12.5.3	MOC.NPAC.SOA.INV.CRE.DUP.serviceProvLRN	12-11
12.5.4	MOC.NPAC.SOA.INV.SET.serviceProvLRN	12-12
12.5.5	MOC.NPAC.SOA.INV.DEL.serviceProvLRN	12-12
12.6	NUMBERPOOLBLOCKNPAC.....	12-12
12.6.1	MOC.SOA.CAP.NOT.numberPoolBlockAttributeValueChange	12-12
12.6.2	MOC.SOA.CAP.NOT.numberPoolBlockStatusAttributeValueChange	12-13
12.7	SERVICEPROVNPA-NXX-X.....	12-13
12.7.1	MOC.NPAC.SOA.CAP.OP.CRE.serviceProvNPA-NXX-X	12-13
12.7.2	MOC.NPAC.SOA.CAP.OP.SET.serviceProvNPA-NXX-X	12-13
12.7.3	MOC.NPAC.SOA.CAP.OP.DEL.serviceProvNPA-NXX-X	12-14
12.7.4	MOC.NPAC.SOA.INV.CRE.DUP.serviceProvNPA-NXX-X	12-14
12.7.5	MOC.NPAC.SOA.INV.SET.serviceProvNPA-NXX-X	12-14
12.7.6	MOC.NPAC.SOA.INV.DEL.serviceProvNPA-NXX-X	12-15
<b>13</b>	<b>LSMS TO NPAC MOC TEST CASES.....</b>	<b>13-1</b>
13.1	LNPNPAC-SMS.....	13-1
13.1.1	MOC.LSMS.CAP.OP.GET.InpNPAC-SMS	13-1
13.1.2	MOC.LSMS.CAP.OP.ACT.InpRecoveryComplete	13-1
13.1.3	MOC.LSMS.CAP.NOT.InpNPAC-SMS-Operational-Information	13-1
13.1.4	MOC.LSMS.INV.GET.InpNPAC-SMS	13-2
13.1.5	MOC.LSMS.INV.ACT.InpRecoveryComplete	13-2
13.1.6	MOC.LSMS.INV.NOT.InpNPAC-SMS-Operational-Information	13-2
13.1.7	MOC.LSMS.CAP.NOT.subscriptionVersionNewNPA-NXX	13-3
13.1.8	MOC.LSMS.INV.NOT.subscriptionVersionNewNPA-NXX	13-3
13.1.9	MOC.LSMS.CAP.ACT.InpNotificationRecovery	13-3
13.1.10	MOC.LSMS.INV.ACT.InpNotificationRecovery	13-4
13.2	LNPSERVICEPROVS.....	13-4
13.2.1	MOC.LSMS.CAP.OP.GET.InpServiceProvs	13-4
13.2.2	MOC.LSMS.INV.GET.InpServiceProvs	13-5
13.3	LNP SUBSCRIPTIONS.....	13-5
13.3.1	MOC.LSMS.CAP.OP.GET.InpSubscriptions	13-5
13.3.2	MOC.LSMS.CAP.ACT.InpSubscriptions.InpDownload	13-5
13.3.3	MOC.LSMS.INV.GET.InpSubscriptions	13-6
13.3.4	MOC.LSMS.INV.ACT.InpSubscriptions	13-6
13.3.5	MOC.LSMS.VAL.InpDownload-NumberPoolBlock	13-6
13.4	LNPNETWORK.....	13-7
13.4.1	MOC.LSMS.CAP.OP.GET.InpNetwork	13-7
13.4.2	MOC.LSMS.CAP.ACT.InpNetwork.InpDownload	13-7
13.4.3	MOC.LSMS.INV.GET.InpNetwork	13-8
13.4.4	MOC.LSMS.INV.ACT.InpNetwork	13-8
13.4.5	MOC.LSMS.VAL.InpDownload-NPA-NXX-X	13-8
13.5	SERVICEPROV.....	13-8
13.5.1	MOC.LSMS.CAP.OP.SET.serviceProv	13-9
13.5.2	MOC.LSMS.CAP.OP.GET.serviceProv	13-9
13.5.3	MOC.LSMS.VAL.SET.SING.serviceProv	13-9
13.5.4	MOC.LSMS.VAL.SET.SING.COND.serviceProv	13-10
13.5.5	MOC.LSMS.VAL.SET.MULT.serviceProv	13-10
13.5.6	MOC.LSMS.INV.SET.serviceProv	13-10
13.5.7	MOC.LSMS.INV.GET.serviceProv	13-11
13.5.8	MOC.LSMS.BND.MIN.SET.serviceProv	13-11
13.5.9	MOC.LSMS.BND.MAX.SET.serviceProv	13-11
13.6	LSMSFILTERNPA-NXX.....	13-12
13.6.1	MOC.LSMS.CAP.OP.CRE.lsmsFilterNPA-NXX	13-12
13.6.2	MOC.LSMS.CAP.OP.GET.lsmsFilterNPA-NXX	13-12



13.6.3	MOC.LSMS.CAP.OP.DEL.IsmsFilterNPA-NXX	13-13
13.6.4	MOC.LSMS.VAL.CRE.AUTO.IsmsFilterNPA-NXX	13-13
13.6.5	MOC.LSMS.VAL.GET.SCOP.FILT.IsmsFilterNPA-NXX	13-13
13.6.6	MOC.LSMS.VAL.DEL.SCOP.FILT.IsmsFilterNPA-NXX	13-14
13.6.7	MOC.LSMS.INV.CRE.IsmsFilterNPA-NXX	13-14
13.6.8	MOC.LSMS.INV.GET.IsmsFilterNPA-NXX	13-15
13.6.9	MOC.LSMS.INV.DEL.IsmsFilterNPA-NXX	13-15
13.7	SUBSCRIPTIONVERSIONNPAC.....	13-15
13.7.1	MOC.LSMS.CAP.OP.GET.subscriptionVersionNPAC	13-15
13.7.2	MOC.LSMS.CAP.NOT.subscriptionVersionNewNPA-NXX	13-16
13.7.3	MOC.LSMS.VAL.GET.SCOP.subscriptionVersionNPAC	13-16
13.7.4	MOC.LSMS.INV.GET.subscriptionVersionNPAC	13-17
13.7.5	MOC.LSMS.INV.NOT.subscriptionVersionNPAC	13-17
13.7.6	MOC.LSMS.BND.GET.MAXQ.subscriptionVersionNPAC	13-17
13.7.7	MOC.LSMS.INV.QUERY.SCOPE.subscriptionVersion	13-18
13.8	SERVICEPROVNETWORK.....	13-18
13.8.1	MOC.LSMS.CAP.OP.GET.serviceProvNetwork	13-18
13.8.2	MOC.LSMS.INV.GET.serviceProvNetwork	13-18
13.9	SERVICEPROVNPA-NXX.....	13-19
13.9.1	MOC.LSMS.CAP.OP.GET.serviceProvNPA-NXX	13-19
13.9.2	MOC.LSMS.CAP.OP.DEL.serviceProvNPA-NXX	13-19
13.9.3	MOC.LSMS.VAL.CRE.AUTO.serviceProvNPA-NXX	13-20
13.9.4	MOC.LSMS.VAL.GET.SCOP.FILT.serviceProvNPA-NXX	13-20
13.9.5	MOC.LSMS.VAL.DEL.SCOP.FILT.serviceProvNPA-NXX	13-20
13.9.6	MOC.LSMS.INV.CRE.serviceProvNPA-NXX	13-21
13.9.7	MOC.LSMS.INV.GET.serviceProvNPA-NXX	13-21
13.9.8	MOC.LSMS.INV.DEL.serviceProvNPA-NXX	13-22
13.10	SERVICEPROVLRN.....	13-22
13.10.1	MOC.LSMS.CAP.OP.GET.serviceProvLRN	13-22
13.10.2	MOC.LSMS.CAP.OP.DEL.serviceProvLRN	13-22
13.10.3	MOC.LSMS.VAL.CRE.AUTO.serviceProvLRN	13-23
13.10.4	MOC.LSMS.VAL.GET.SCOP.FILT.serviceProvLRN	13-23
13.10.5	MOC.LSMS.VAL.DEL.SCOP.FILT.serviceProvLRN	13-24
13.10.6	MOC.LSMS.INV.CRE.serviceProvLRN	13-24
13.10.7	MOC.LSMS.INV.GET.serviceProvLRN	13-24
13.10.8	MOC.LSMS.INV.DEL.serviceProvLRN	13-25
13.11	NUMBERPOOLBLOCKNPAC.....	13-25
13.11.1	MOC.LSMS.CAP.OP.GET.numberPoolBlockNPAC	13-25
13.11.2	MOC.LSMS.VAL.GET.SCOP.numberPoolBlockNPAC	13-26
13.11.3	MOC.LSMS.INV.GET.numberPoolBlockNPAC	13-26
13.11.4	MOC.LSMS.INV.GET.SCOP.numberPoolBlockNPAC	13-26
13.12	SERVICEPROVNPA-NXX-X.....	13-26
13.12.1	MOC.LSMS.CAP.OP.GET.serviceProvNPA-NXX-X	13-27
13.12.2	MOC.LSMS.VAL.GET.SCOP.serviceProvNPA-NXX-X	13-27
13.12.3	MOC.LSMS.INV.GET.serviceProvNPA-NXX-X	13-27
13.12.4	MOC.LSMS.INV.GET.SCOP.serviceProvNPA-NXX-X	13-28
<b>14</b>	<b>NPAC TO LSMS MOC TEST CASES.....</b>	<b>14-1</b>
14.1	LNPLOCALSMS.....	14-1
14.1.1	MOC.NPAC.CAP.OP.GET.lnpLocalSMS	14-1
14.1.2	MOC.NPAC.INV.CRE.INH.lnpLocalSMS	14-1
14.1.3	MOC.NPAC.INV.SET.lnpLocalSMS	14-2
14.1.4	MOC.NPAC.INV.DEL.lnpLocalSMS	14-2
14.1.5	MOC.LSMS.CAP.NOT.lnpNPAC-SMS-Operational-Information	14-2
14.2	LNP SUBSCRIPTIONS.....	14-3
14.2.1	MOC.NPAC.CAP.OP.GET.lnpSubscriptions	14-3

14.2.2	MOC.NPAC.CAP.OP.ACT.InpSubscriptions	14-3
14.2.3	MOC.NPAC.CAP.OP.NOT.InpSubscriptions	14-4
14.2.4	MOC.NPAC.INV.CRE.INH.InpSubscriptions	14-4
14.2.5	MOC.NPAC.INV.SET.InpSubscriptions	14-4
14.2.6	MOC.NPAC.INV.ACT.SYN.ID.InpSubscriptions	14-5
14.2.7	MOC.NPAC.INV.ACT.SYN.CLS.InpSubscriptions	14-5
14.2.8	MOC.NPAC.INV.ACT.InpSubscriptions	14-5
14.2.9	MOC.NPAC.INV.NOT.InpSubscriptions	14-6
14.2.10	MOC.NPAC.INV.DEL.InpSubscriptions	14-6
14.3	LNPNETWORK.....	14-6
14.3.1	MOC.NPAC.CAP.OP.GET.InpNetwork	14-6
14.3.2	MOC.NPAC.INV.CRE.INH.InpNetwork	14-7
14.3.3	MOC.NPAC.INV.SET.InpNetwork	14-7
14.3.4	MOC.NPAC.INV.ACT.InpNetwork	14-7
14.3.5	MOC.NPAC.INV.DEL.InpNetwork	14-8
14.4	SUBSCRIPTIONVERSION.....	14-8
14.4.1	MOC.NPAC.CAP.OP.CRE.subscriptionVersion	14-8
14.4.2	MOC.NPAC.CAP.OP.SET.subscriptionVersion	14-9
14.4.3	MOC.NPAC.CAP.OP.GET.subscriptionVersion	14-9
14.4.4	MOC.NPAC.CAP.OP.DEL.subscriptionVersion	14-9
14.4.5	MOC.NPAC.VAL.SET.SING.subscriptionVersion	14-10
14.4.6	MOC.NPAC.VAL.SET.MULT.subscriptionVersion	14-10
14.4.7	MOC.NPAC.VAL.SET.SCOP.FILT.subscriptionVersion	14-10
14.4.8	MOC.NPAC.VAL.GET.SCOP.FILT.subscriptionVersion	14-11
14.4.9	MOC.NPAC.VAL.DEL.SCOP.FILT.subscriptionVersion	14-11
14.4.10	MOC.NPAC.INV.CRE.subscriptionVersion	14-11
14.4.11	MOC.NPAC.INV.SET.RO.subscriptionVersion	14-12
14.4.12	MOC.NPAC.INV.SET.MULT.subscriptionVersion	14-12
14.4.13	MOC.NPAC.INV.SET.SYN.subscriptionVersion	14-12
14.4.14	MOC.NPAC.INV.SET.SCOP.subscriptionVersion	14-13
14.4.15	MOC.NPAC.INV.DEL.SCOP.subscriptionVersion	14-13
14.4.16	MOC.NPAC.BND.SET.MIN.subscriptionVersion	14-13
14.4.17	MOC.NPAC.BND.SET.MAX.subscriptionVersion	14-14
14.5	SERVICEPROVNETWORK.....	14-14
14.5.1	MOC.NPAC.CAP.OP.CRE.serviceProvNetwork	14-14
14.5.2	MOC.NPAC.CAP.OP.GET.serviceProvNetwork	14-15
14.5.3	MOC.NPAC.CAP.OP.SET.serviceProvNetwork	14-15
14.5.4	MOC.NPAC.CAP.OP.DEL.serviceProvNetwork	14-15
14.5.5	MOC.NPAC.INV.CRE.DUP.serviceProvNetwork	14-16
14.5.6	MOC.NPAC.INV.SET.RO.serviceProvNetwork	14-16
14.5.7	MOC.NPAC.INV.SET.SYN.serviceProvNetwork	14-16
14.5.8	MOC.NPAC.INV.SET.serviceProvNetwork	14-17
14.5.9	MOC.NPAC.INV.GET.serviceProvNetwork	14-17
14.5.10	MOC.NPAC.INV.DEL.serviceProvNetwork	14-17
14.5.11	MOC.NPAC.INV.DEL.CO.serviceProvNetwork	14-18
14.5.12	MOC.NPAC.BND.SET.MIN.serviceProvNetwork	14-18
14.5.13	MOC.NPAC.BND.SET.MAX.serviceProvNetwork	14-18
14.6	SERVICEPROVNPA-NXX.....	14-18
14.6.1	MOC.NPAC.CAP.OP.CRE.serviceProvNPA-NXX	14-19
14.6.2	MOC.NPAC.CAP.OP.DEL.serviceProvNPA-NXX	14-19
14.6.3	MOC.NPAC.INV.CRE.DUP.serviceProvNPA-NXX	14-19
14.6.4	MOC.NPAC.INV.SET.serviceProvNPA-NXX	14-20
14.6.5	MOC.NPAC.INV.DELserviceProvNPA-NXX	14-20
14.7	SERVICEPROVLRN.....	14-20
14.7.1	MOC.NPAC.CAP.OP.CRE.serviceProvLRN	14-21
14.7.2	MOC.NPAC.CAP.OP.DEL.serviceProvLRN	14-21

14.7.3	MOC.NPAC.INV.CRE.DUP.serviceProvLRN	14-21
14.7.4	MOC.NPAC.INV.SET.serviceProvLRN	14-22
14.7.5	MOC.NPAC.INV.DEL.serviceProvLRN	14-22
14.8	NUMBERPOOLBLOCK.....	14-22
14.8.1	MOC.NPAC.CAP.OP.CRE.numberPoolBlock	14-22
14.8.2	MOC.NPAC.CAP.OP.SET.numberPoolBlock	14-23
14.8.3	MOC.NPAC.CAP.OP.GET.numberPoolBlock	14-23
14.8.4	MOC.NPAC.CAP.OP.GET.MULTIPLE.numberPoolBlock	14-23
14.8.5	MOC.NPAC.CAP.OP.DEL.numberPoolBlock	14-24
14.8.6	MOC.NPAC.CAP.OP.SET.SING.numberPoolBlock	14-24
14.8.7	MOC.NPAC.CAP.OP.SET.MULT.numberPoolBlock	14-24
14.8.8	MOC.NPAC.INV.CRE.numberPoolBlock	14-24
14.8.9	MOC.NPAC.INV.SET.numberPoolBlock	14-25
14.8.10	MOC.NPAC.INV.DEL.numberPoolBlock	14-25
14.9	SERVICEPROVNPA-NXX-X.....	14-25
14.9.1	MOC.NPAC.CAP.OP.CRE.serviceProvNPA-NXX-X	14-26
14.9.2	MOC.NPAC.CAP.OP.SET.serviceProvNPA-NXX-X	14-26
14.9.3	MOC.NPAC.CAP.OP.DEL.serviceProvNPA-NXX-X	14-26
14.9.4	MOC.NPAC.INV.CRE.DUP.serviceProvNPA-NXX-X	14-26
14.9.5	MOC.NPAC.INV.SET.serviceProvNPA-NXX-X	14-27
14.9.6	MOC.NPAC.INV.DEL.serviceProvNPA-NXX-X	14-27
<b>15</b>	<b>ASSOCIATION MANAGEMENT TEST CASES.....</b>	<b>15-1</b>
15.1	TEST CASES.....	15-1
15.1.1	AMG.SOA.ASSOC.SAME and AMG.LSMS.ASSOC.SAME	15-1
15.1.2	AMG.SOA.ASSOC.OTHER and AMG.LSMS.ASSOC.OTHER	15-1
15.1.3	AMG.SOA.REQTMOT and AMG.LSMS.REQTMOT	15-2
15.1.4	AMG.SOA.RETRY.CMIP and AMG.LSMS.RETRY.CMIP	15-2
15.1.5	AMG.SOA.RETRY.ASSOC and AMG.LSMS.RETRY.ASSOC	15-3
15.1.6	AMG.SOA.SECVIOL and AMG.LSMS.SECVIOL	15-3
15.1.7	AMG.SOA.LOSS and AMG.LSMS.LOSS	15-4
15.1.8	AMG.SOA.DOWN and AMG.LSMS.DOWN	15-4
<b>16</b>	<b>APP-TO-APP TEST CASES.....</b>	<b>16-1</b>
16.1	AUDIT TEST CASES.....	16-1
16.1.1	A2A.LSMS.VAL.MISSVER.subscriptionAudit	16-1
16.1.2	A2A.LSMS.VAL.OBSVER.subscriptionAudit	16-1
16.1.3	A2A.LSMS.VAL.ERRVER.subscriptionAudit	16-2
16.1.4	A2A.SOA.VAL.NODIS.TN.subscriptionAudit	16-2
16.1.5	A2A.SOA.VAL.NODIS.TNRNG.subscriptionAudit	16-3
16.1.6	A2A.SOA.VAL.NODIS.ACTRNG.subscriptionAudit	16-4
16.1.7	A2A.SOA.VAL.WITHDIS.TN.subscriptionAudit	16-5
16.1.8	A2A.SOA.VAL.WITHDIS.TNRNG.subscriptionAudit	16-6
16.1.9	A2A.SOA.VAL.WITHDIS.ACTRNG.subscriptionAudit	16-7
16.1.10	A2A.SOA.VAL.NPACCNCLD.subscriptionAudit	16-8
16.1.11	A2A.SOA.INV.CRENOT.TIMOUT.subscriptionAudit	16-9
16.1.12	A2A.SOA.VAL.WITHDIS.WSMSC.RANGE.subscriptionAudit	16-9
16.1.13	A2A.SOA.VAL.WITHDIS.WSMSC.SINGLE.subscriptionAudit	16-10
16.1.14	A2A.SOA.VAL.WITHDIS.ASSOCSP.RANGE.subscriptionAudit	16-11
16.1.15	A2A.SOA.VAL.WITHDIS.ASSOCSP.SINGLE.subscriptionAudit	16-12
16.1.16	LSMS.VAL.MISSVER.subscriptionAudit.POOL	16-13
16.2	SERVICE PROVIDER AND NETWORK DATA TEST CASES.....	16-14
16.2.1	A2A.LSMS.VAL.CREND.serviceProviderNPA-NXX	16-14
16.2.2	A2A.LSMS.VAL.DELND.serviceProviderNPA-NXX	16-14
16.2.3	A2A.LSMS.VAL.CREND.serviceProviderLRN	16-15
16.2.4	A2A.LSMS.VAL.DELND.serviceProviderLRN	16-15

16.2.5	A2A.SOA.CAP.OP.SET.ASSOCSP.serviceProv	16-15
16.2.6	A2A.SOA.CAP.OP.GET.ASSOCSP.serviceProv	16-16
16.2.7	A2A.SOA.VAL.CREND.ASSOCSP.serviceProviderNPA-NXX	16-16
16.2.8	A2A.SOA.VAL.DELND.ASSOCSP.serviceProviderNPA-NXX	16-16
16.2.9	A2A.SOA.VAL.CREND.ASSOCSP.serviceProviderLRN	16-17
16.2.10	A2A.SOA.VAL.DELND.ASSOCSP.serviceProviderLRN	16-17
16.3	SUBSCRIPTION VERSION CREATE TEST CASES.....	16-17
16.3.1	A2A.NSOA.VAL.CREATE.TN-RANGE.SubscriptionVersion	16-18
16.3.2	A2A.NSOA.VAL.CREATE.CONFLICT.SubscriptionVersion	16-18
16.3.3	A2A.OSOA.VAL.CREATE.TN-RANGE.SubscriptionVersion	16-19
16.3.4	A2A.OSOA.VAL.NOCONC.ACTIVATE.SubscriptionVersion	16-20
16.3.5	A2A.OSOA.VAL.NOCONC.NOACTIVATE.SubscriptionVersion	16-21
16.3.6	A2A.OSOA.VAL.CREATE.CONFLICT.SubscriptionVersion	16-23
16.3.7	A2A.NSOA.VAL.CREATE.INTRA-SP-PORT.SubscriptionVersion	16-23
16.3.8	A2A.DSOA.VAL.PORT-TO-ORIG.SubscriptionVersion	16-24
16.3.9	A2A.NSOA.INV.MISS.INITIAL.CONC.SubscriptionVersion	16-25
16.3.10	A2A.NSOA.INV.STATE-TRANS.PEND-ACTIVE.SubscriptionVersion	16-25
16.3.11	A2A.NSOA.INV.STATE-TRANS.PEND-OLD.SubscriptionVersion	16-26
16.3.12	A2A.OSOA.INV.STATE-TRANS.PEND-OLD.SubscriptionVersion	16-27
16.3.13	A2A.OSOA.INV.STATE-TRANS.PEND-FAILED.SubscriptionVersion	16-27
16.3.14	A2A.NSOA.INV.CREATE.ACTIVE.SubscriptionVersion	16-28
16.3.15	A2A.OSOA.INV.CREATE.SENDING.SubscriptionVersion	16-29
16.3.16	A2A.NSOA.INV.OBJCRE.NOTMISS.SubscriptionVersion	16-29
16.3.17	A2A.OSOA.INV.OBJCRE.NOTMISS.SubscriptionVersion	16-30
16.4	SUBSCRIPTION VERSION ACTIVATE TEST CASES.....	16-30
16.4.1	A2A.NSOA.VAL.ACTIVATE.BYNPAC.SubscriptionVersion	16-30
16.4.2	A2A.NSOA.VAL.ACTIVATE.SubscriptionVersion	16-31
16.4.3	A2A.NSOA.VAL.ACTIVATE.FAIL.SubscriptionVersion	16-31
16.4.4	A2A.NSOA.VAL.ACTIVATE.PARTFAIL.SubscriptionVersion	16-32
16.4.5	A2A.OSOA.VAL.ACTIVATE.SubscriptionVersion	16-33
16.4.6	A2A.OSOA.VAL.ACTIVATE.FAIL.SubscriptionVersion	16-33
16.4.7	A2A.OSOA.VAL.ACTIVATE.PARTFAIL.SubscriptionVersion	16-34
16.4.8	A2A.NSOA.ACTIVATE.ACTNOTMISS.SubscriptionVersion	16-34
16.4.9	A2A.NSOA.INV.ACTIVATE.PARTFAIL.SubscriptionVersion	16-35
16.4.10	A2A.OSOA.INV.ACTIVATE.PARTFAIL.SubscriptionVersion	16-36
16.4.11	A2A.NSOA.VAL.ACTIVATE.TN-RANGE.SubscriptionVersion	16-36
16.5	SUBSCRIPTION VERSION MODIFY TEST CASES.....	16-37
16.5.1	A2A.NSOA.VAL.MODIFY.PEND.SubscriptionVersion	16-37
16.5.2	A2A.OSOA.VAL.MODIFY.PEND.SubscriptionVersion	16-38
16.5.3	A2A.SOA.VAL.MODIFY.ACTIVE.SubscriptionVersion	16-38
16.5.4	A2A.SOA.VAL.MODIFY.ACTIVE.TN-RANGE.SubscriptionVersion	16-39
16.5.5	A2A.SOA.VAL.MODIFY.BYNPAC.ACTIVE.SubscriptionVersion	16-39
16.5.6	A2A.SOA.VAL.MODIFY.PARTFAIL.SubscriptionVersion	16-40
16.5.7	A2A.SOA.VAL.MODIFY.FAIL.SubscriptionVersion	16-40
16.5.8	A2A.SOA.INV.MODIFY.PARTFAIL.NOSPLIST.SubscriptionVersion	16-41
16.5.9	A2A.SOA.INV.MODIFY.ACTIVE.NOTMISS.SubscriptionVersion	16-42
16.5.10	A2A.SOA.INV.MODIFY.ATTRCHNG.NOTMISS.SubscriptionVersion	16-42
16.5.11	A2A.SOA.INV.MODIFY.ATTRSAME.NOTMISS.SubscriptionVersion	16-43
16.5.12	A2A.SOA.VAL.MODIFY.PEND.TN-RANGE.SubscriptionVersion	16-43
16.6	SUBSCRIPTION VERSION CANCEL TEST CASES.....	16-44
16.6.1	A2A.SOA.VAL.CANCEL.SubscriptionVersion	16-44
16.6.2	A2A.NSOA.VAL.CANCEL.BYOSOA.SubscriptionVersion	16-44
16.6.3	A2A.NSOA.VAL.CANCEL.TN-RANGE.SubscriptionVersion	16-45
16.6.4	A2A.OSOA.VAL.CANCEL.SubscriptionVersion	16-46
16.6.5	A2A.OSOA.VAL.CANCEL.BYNSOA.SubscriptionVersion	16-47
16.6.6	A2A.OSOA.VAL.CANCEL.TN-RANGE.SubscriptionVersion	16-48

16.6.7	A2A.OSOA.VAL.CANCEL.NOCONC.SubscriptionVersion	16-49
16.6.8	A2A.NSOA.VAL.CANCEL.BYNPAC.SubscriptionVersion	16-50
16.6.9	A2A.OSOA.VAL.CANCEL.BYNPAC.SubscriptionVersion	16-51
16.6.10	A2A.NSOA.VAL.CANCEL.ACKREQ.SubscriptionVersion	16-51
16.6.11	A2A.OSOA.VAL.CANCEL.ACKREQ.SubscriptionVersion	16-52
16.6.12	A2A.NSOA.INV.CANCEL.CONFLICT.SubscriptionVersion	16-53
16.6.13	A2A.NSOA.VAL.CANCEL.CANCELED.SubscriptionVersion	16-53
16.6.14	A2A.OSOA.VAL.CANCEL.CONFLICT.SubscriptionVersion	16-54
16.6.15	A2A.NSOA.INV.CANCEL.PEND.SubscriptionVersion	16-55
16.6.16	A2A.OSOA.INV.CANCEL.CONFLICT.SubscriptionVersion	16-56
16.6.17	A2A.NSOA.INV.CANCEL.ACTIVE.SubscriptionVersion	16-57
16.7	SUBSCRIPTION VERSION DISCONNECT TEST CASES.....	16-57
16.7.1	A2A.SOA.VAL.IMMDISC.SubscriptionVersion	16-58
16.7.2	A2A.SOA.VAL.DEFDISC.SubscriptionVersion	16-58
16.7.3	A2A.SOA.VAL.IMMDISC.BYNPAC.SubscriptionVersion	16-58
16.7.4	A2A.SOA.VAL.IMMDISC.FAIL.SubscriptionVersion	16-59
16.7.5	A2A.SOA.VAL.IMMDISC.PARTFAIL.SubscriptionVersion	16-59
16.7.6	A2A.SOA.VAL.IMMDISC.TN-RANGE.SubscriptionVersion	16-60
16.7.7	A2A.SOA.INV.IMMDISC.ACT.OLD.SubscriptionVersion	16-60
16.7.8	A2A.SOA.INV.IMMDISC.OLD.SubscriptionVersion	16-61
16.7.9	A2A.SOA.INV.IMMDISC.FAILED.SubscriptionVersion	16-61
16.7.10	A2A.SOA.INV.IMMDISC.OLD.FAILService Provider.SubscriptionVersion	16-62
16.7.11	A2A.SOA.VAL.CANCEL.DISCPEND.SubscriptionVersion	16-62
16.8	SUBSCRIPTION VERSION CONFLICT TEST CASES.....	16-63
16.8.1	A2A.NSOA.VAL.CONFLICT.RESOLV.SubscriptionVersion	16-63
16.8.2	A2A.NSOA.VAL.CONFLICT.RESOLV.BYNNSOA.SubscriptionVersion	16-63
16.8.3	A2A.OSOA.VAL.CONFLICT.RESOLV.SubscriptionVersion	16-64
16.8.4	A2A.OSOA.VAL.CONFLICT.RESOLV.BYOSOA.SubscriptionVersion	16-64
16.8.5	A2A.NSOA.VAL.CONFLICT.RESOLV.TN-RANGE.BYNNSOA.SubscriptionVersion	16-65
16.9	LSMS TEST CASES.....	16-66
16.9.1	A2A.LSMS.VAL.ACTIVATE.BYNPAC.SubscriptionVersion	16-66
16.9.2	A2A.LSMS.VAL.MODIFY.BYNPAC.ACTIVE.SubscriptionVersion	16-66
16.9.3	A2A.LSMS.VAL.IMMDISC.BYNPAC.SubscriptionVersion	16-66
16.9.4	A2A.LSMS.VAL.CREATE.MULT.SubscriptionVersion	16-67
16.9.5	A2A.LSMS.INV.CREATE.MULT.SubscriptionVersion	16-67
16.9.6	A2A.LSMS.INV.CREATE.UNKNOWN.NPA-NXX.SubscriptionVersion	16-68
16.10	SOA WSMSC DATA TEST CASES (NANC 203).....	16-68
16.10.1	A2A.NSOA.VAL.CREATE.WSMSC.SubscriptionVersion	16-68
16.10.2	A2A.NSOA.VAL.MODIFY.WSMSC.SubscriptionVersion	16-69
16.10.3	A2A.SOA.VAL.QUERY.WSMSC.SubscriptionVersion	16-69
16.11	LSMS WSMSC DATA TEST CASES (NANC 203).....	16-70
16.11.1	A2A.LSMS.VAL.CREATE.WSMSC.SubscriptionVersion	16-70
16.11.2	A2A.LSMS.VAL.CREATE.MULT.WSMSC.SubscriptionVersion	16-70
16.11.3	A2A.LSMS.VAL.QUERY.SCOPE.WSMSC.SubscriptionVersion	16-70
16.11.4	A2A.LSMS.VAL.MODIFY.WSMSC.SubscriptionVersion	16-71
16.12	SUBSCRIPTION TIMER AND BUSINESS TYPES (NANC 201 AND 202).....	16-71
16.12.1	A2A.SOA.VAL.QUERY.SUBTIMER.SubscriptionVersion	16-71
16.12.2	A2A.SOA.VAL.QUERY.BUSTYPE.SubscriptionVersion	16-71
16.12.3	A2A.OSOA.VAL.NOT.subscriptionVersionOldSP-ConcurrenceRequest	16-72
16.12.4	A2A.OSOA.VAL.NOT.subscriptionVersionOldSPFinalConcurrenceWindowExpiration	16-72
16.12.5	A2A.NSOA.VAL.NOT.subscriptionVersionNewSP-CreateRequest	16-72
16.13	MISSING SENDING NOTIFICATION TEST CASES (NANC 207).....	16-73
16.13.1	A2A.NSOA.VAL.ACTIVATE.NOTMISS.SubscriptionVersion	16-73
16.13.2	A2A.OSOA.VAL.ACTIVATE.NOTMISS.SubscriptionVersion	16-74
16.13.3	A2A.SOA.VAL.MODIFY.ACTIVE.NOTMISS.SubscriptionVersion	16-74
16.13.4	A2A.SOA.VAL.IMMDISC.NOTMISS.SubscriptionVersion	16-75



16.14	ASSOCIATED SERVICE PROVIDER TEST CASES (NANC 48).....	16-76
16.14.1	A2A.NSOA.VAL.CREATE.FIRST.ASSOCSP.SubscriptionVersion	16-76
16.14.2	A2A.NSOA.VAL.CREATE.SECOND.ASSOCSP.SubscriptionVersion	16-77
16.14.3	A2A.OSOA.VAL.CREATE.FIRST.ASSOCSP.SubscriptionVersion	16-77
16.14.4	A2A.OSOA.VAL.CREATE.SECOND.ASSOCSP.SubscriptionVersion	16-78
16.14.5	A2A.OSOA.VAL.NOCONC.ACTIVATE.ASSOCSP.SubscriptionVersion	16-79
16.14.6	A2A.NSOA.VAL.ACTIVATE.ASSOCSP.SubscriptionVersion	16-80
16.14.7	A2A.NSOA.VAL.MODIFY.PEND.ASSOCSP.SubscriptionVersion	16-80
16.14.8	A2A.OSOA.VAL.MODIFY.PEND.ASSOCSP.SubscriptionVersion	16-81
16.14.9	A2A.SOA.VAL.MODIFY.ACTIVE.ASSOCSP.SubscriptionVersion	16-82
16.14.10	A2A.NSOA.VAL.CANCEL.ASSOCSP.SubscriptionVersion	16-82
16.14.11	A2A.OSOA.VAL.CANCEL.ASSOCSP.SubscriptionVersion	16-83
16.14.12	A2A.NSOA.VAL.CANCEL.ACKREQ.ASSOCSP.SubscriptionVersion	16-84
16.14.13	A2A.OSOA.VAL.CANCEL.ACKREQ.ASSOCSP.SubscriptionVersion	16-85
16.14.14	A2A.SOA.VAL.IMMDISC.ASSOCSP.SubscriptionVersion	16-86
16.14.15	A2A.SOA.VAL.DEFDISC.ASSOCSP.SubscriptionVersion	16-87
16.14.16	A2A.NSOA.VAL.CONFLICT.RESOLV.ASSOCSP.SubscriptionVersion	16-88
16.14.17	A2A.OSOA.VAL.CONFLICT.RESOLV.ASSOCSP.SubscriptionVersion	16-88
16.14.18	A2A.SOA.VAL.PORT-TO-ORIG.ASSOCSP.SubscriptionVersion	16-88
16.14.19	A2A.SOA.CAP.ACT.ASSOCSP.numberPoolBlockCreateAction	16-89
16.14.20	A2A.SOA.CAP.OP.SET.ASSOCSP.numberPoolBlock	16-90
16.15	MISCELLANEOUS SCENARIOS TEST CASES.....	16-90
16.15.1	A2A.SOA.VAL.MISC.ACTION.resync	16-90
16.15.2	A2A.SOA.INV.MISC.ACTION.resync	16-91
16.15.3	A2A.SOA.VAL.MISC.ACTION.ASSOCSP.resync	16-92
16.15.4	A2A.LSMS.VAL.MISC.ACTION.resync	16-92
16.15.5	A2A.LSMS.INV.MISC.ACTION.resync	16-94
16.15.6	A2A.SOA.VAL.MISC.ACTION.resync_3_1	16-94
16.16	A2A NUMBER POOLING – SOA TO NPAC SMS.....	16-95
16.16.1	A2A.SOA.VAL.GET.SCOPE.subsriptionVersion.TN-LNPTYPE	16-95
16.17	A2A NUMBER POOLING – LSMS TO NPAC SMS.....	16-95
16.17.1	A2A.LSMS.VAL.GET.SCOPE.subsriptionVersion.TN-LNPTYPE	16-95
16.18	A2A NUMBER POOLING NPAC SMS TO LSMS.....	16-96
16.18.1	A2A.LSMS.VAL.CREATE.BYNPAC.subsriptionVersion.POOL	16-96
16.18.2	A2A.LSMS.VAL.CREATE.RANGE.BYNPAC.subsriptionVersion.POOL	16-96
16.18.3	A2A.LSMS.VAL.GET.SCOPE.BYNPAC.subsriptionVersion.TN-LNPTYPE	16-96
<b>APPENDIX A TESTING REGISTRATION FORM.....</b>		<b>1</b>
<b>APPENDIX B TEST CASE NOMENCLATURE.....</b>		<b>1</b>
<b>APPENDIX C COMPLETE ITP TEST CASE CHECKLIST.....</b>		<b>1</b>
<b>APPENDIX D STANDARD REGRESSION TEST CASE CHECKLIST.....</b>		<b>1</b>
Appendix E Release 3.1 Test Case Checklist.....		2



# 1 Introduction

## 1.1 Document Overview

This document identifies an Interoperability Test Plan (ITP) to be performed by users of the NeuStar Interoperability Testing Service offered at the TMN Test Center. The ITP will be jointly executed by the Test Center and the individual Service Providers or SOA/LSMS Vendors wishing to test their SOA and/or LSMS systems. The test cases defined in this test plan must be executed and passed, before any service provider is allowed to connect their SOA or LSMS to the actual NPAC SMS. This is to ensure that the SOA and LSMS do not corrupt the NPAC SMS and vice versa. The SOA-NPAC and LSMS-NPAC Interoperability Test is broken down into five groups of test cases:

1. Stack-to-Stack (S2S)  
For S2S, due to the nature of association establishment, if the SOA and LSMS share a common computing environment, the S2S test cases for only one of the first two interfaces are required. The S2S test cases applied to the SOA and/or LSMS to NPAC SMS interfaces are identical. The S2S test cases in this plan will basically test the ACSE connectivity between the SOA/LSMS and the NPAC SMS. The end-to-end delivery of PDUs will be implicitly covered by the MOC test cases.
2. Security  
The Security test cases will test the functionality of each item in the `lnpAccessControl` field. For example, these test cases will address the `systemId`, `cmipDepartureTime`, `sequenceNumber`, `listId`, `keyId`, and `signature` sub-fields. The Local Carriers will have the option to postpone execution of the signature check test case until the end of the test process. However the other Security test cases (i.e. `cmipDepartureTime` and sequence numbers checks) must be completed successfully prior to MOC testing.
3. Managed Object Conformance (MOC)  
MOC test cases address the basic operations of each and every Managed Object (MO) in the system. They test everything described in the MOCS and the GDMO file except for the application-to-application level of behavior of the MOs. They mainly consist of a single CMIP request and the corresponding response.
4. Association Management  
The Association Management test group is performed to ensure that a SOA and a LSMS can recover an association. It also tests whether the SOA or LSMS can switch to the backup NPAC SMS when it is instructed to do so by the NPAC SMS.
3. Application-to-Application (A2A)  
The A2A test cases are concerned with the behavior of the application as a whole on the SOA and the LSMS. They focus on the transactions that are allowed by the two interfaces (SOA to NPAC and NPAC to LSMS) and span multiple MO classes and CMIP requests/responses.

This document assumes familiarity with the terms, structure, and content of the following documents:

NPAC SMS Functional Requirements Specification

NPAC SMS Interoperable Interface Specification

This document describes the test process, its inputs and outputs, and the specific responsibilities of those participating in the test process. The nomenclature used to describe the individual tests, and the formats of the test results are also described. The specific test cases documented are:

SOA or LSMS to NPAC S2S Test Cases

SOA or LSMS to NPAC Security Test Cases

SOA to NPAC SMS MOC Test Cases

LSMS to NPAC SMS MOC Test Cases

NPAC SMS to SOA MOC Test Cases  
NPAC SMS to LSMS MOC Test Cases  
SOA or LSMS to NPAC SMS Association Management Test Cases  
SOA to NPAC SMS A2A Test Cases  
NPAC SMS to LSMS A2A Test Cases  
NPAC SMS to LSMS A2A Test Cases

### 1.1.1 Document Structure

The ITP contains the following sections:

***Chapter 1 Introduction***

Introduction to the document.

***Chapter 2 The Testing Process***

Defines the interoperability testing process.

***Chapter 3 Stack to Stack Test Cases***

Explanation of the stack-to-stack interoperability testing.

***Chapter 4 Security Test Cases***

Explanation of the security interoperability testing.

***Chapter 5 Managed Object Conformance Interoperability Testing***

Explanation of the MOC interoperability testing.

***Chapter 6 Association Management Interoperability Testing***

Explanation of the recovery interoperability testing.

***Chapter 7 Application to Application Interoperability Testing***

Explanation of the App-to-App interoperability testing.

***Chapter 8 Interoperability Testing Exit Criteria***

Explanation of the exit criteria.

***Chapter 9 Stack to Stack Test Cases***

Test cases that verify the OSI Protocol stack.

***Chapter 10 Security Test Cases***

Security related test cases.

***Chapter 11 SOA to NPAC MOC Test Cases***

Managed Object Conformance (MOC) test cases that test the basic CMIP functionality for the SOA to NPAC interface.

***Chapter 12 NPAC to SOA MOC Test Cases***

MOC test cases that test the basic CMIP functionality for the NPAC to SOA interface.

***Chapter 13 LSMS to NPAC MOC Test Cases***

MOC test cases that test the basic CMIP functionality for the LSMS to NPAC interface.

***Chapter 14 NPAC to LSMS MOC Test Cases***

MOC test cases that test the basic CMIP functionality for the NPAC to LSMS interface.

***Chapter 15 Association Management Test Cases***

Test cases related to association recovery.

***Chapter 16 APP-to-APP Test Cases***

Test cases related how the application operates. As opposed to the single events tested in the MOC test cases, these incorporate entire scenarios.

***Appendix A Testing Registration Form***

An example of the ITP registration form.

***Appendix B Test Case Nomenclature***

A definition of terms used in the test case names.

***Appendix C Complete ITP Test Case Checklist***

A complete listing of all test cases.

***Appendix D Standard Regression Test Case Checklist***

A checklist of all Release 3.0 regression test cases.

***Appendix E 3.0 Test Case Checklist***

A checklist of all Release 3.0 test cases.

## **1.2 Document Numbering Strategy**

Starting with Release 2.0, the documentation number of the ITP document will be Version X.Y.Z as follows:

X – will only be incremented when a new major release of the NPAC SMS system is authorized. It will contain only the Change Orders that have been authorized for inclusion in this new major release.

Y – will only be incremented when a new sub-release of an existing release X is authorized. It will contain only the Change Orders that have been authorized for inclusion in this new sub-release.

Z – will be incremented when documentation only clarifications and/or backward compatibility issues or other deficiencies, are made in the FRS and/or IIS. This number will be reset to 0 when Y is incremented.

For example, the first release of the Release 2 ITP will be numbered 2.0.0. If documentation only clarifications are introduced in the next release of the ITP document it will be numbered 2.0.1. If requirements are added to Release 2.0 that require NPAC SMS software changes then the next release of the ITP document will be numbered 2.1.0.

This number scheme is intended to make the mapping between NPAC SMS and the FRS and IIS documentation consistent.

## **1.3 Testing Overview**

The interoperability testing for Release 3.0 includes all of the new test cases in this ITP document and the standard regression test cases.

**Note:**

- It is mandatory for all LSMS and SOA products to execute the interoperability testing for Release 3.0 including the standard regression test cases to ensure backward compatibility with their existing SOA or LSMS products.
- During regression , 3.0 and 3.1.0 interoperability testing, the test data should reflect the features currently supported by the LSMS or SOA product. (e.g., WSMSC data, Timer Type or Business Day Type attributes)
- Conditional test cases that were not run in ITP 2.0 need not be re-run for Release 3.0 or 3.1.0 interoperability testing if the SOA or LSMS product does not support the related functionality.
- New 3.0 and 3.1.0 interoperability test cases must be run if a service provider is supporting the new functionality.
- If a SOA or LSMS product implements new features that existed in the NPAC SMS prior to Release 3.1.0, the product must execute the prior ITP test cases corresponding to the new functionality.
- The access control must be present for all test cases. However, the digital signature does not have to be specified.
- The action and notification names reflect those used in the NPAC SMS GDMO file not the ASN.1 file.

- The invalid test case scenarios reflect the preferred behavior or response. Other behavior or responses will be reviewed on a case-by-case basis.

## **1.4 Document Version History**

### **1.4.1 ITP Version 1.7**

Released on 07/31/97, supports NANC IIS Version 1.2.

### **1.4.2 ITP Version 1.8**

Released on 06/30/98, supports NANC IIS Version 1.9 and the Illinois Number Pooling Flows and contains new test cases supporting the subscriptionVersionNewNPA-NXX notification and the LNP Type of 'POOL'.

### **1.4.3 Release 2.0.1**

- Incorporated test cases from NPAC SMS Interoperability Test Plan 1.8 and reformatted to new style.
- Merged test cases from NPAC SMS Interoperability Test Plan 2.0.0.
- Removed obsolete test cases.
- Removed all test cases of format "MOC.XXX.VAL.GET.SING.\*" and "MOC.XXX.VAL.GET.MULT.\*".
- Converted all "MOC.XXX.INV.GET.\*" test cases to retrieve ALL attributes.
- Changed any hard-coded values to "an agreed upon value".

### **1.4.4 Release 3.0.0**

- Updated test case wording to current guidelines.
- Removed obsolete test cases.
- Created 3.x test cases and Test Case Checklist.
- Updated 16.15.4 (LSMS Resync) to include Number Pool Blocks.
- Updated Regression Test Case List to include 16.15.1 (SOA Resync).
- Removed 2.x Test Case Checklists.
- Removed Release 1.4 Illinois Number Pooling test cases.

### **1.4.5 Release 3.0.1**

- Updated Regression Test Case List to include Release 3.0 test cases.

### **1.4.6 Release 3.1.0**

- Updated for NPAC SMS Release 3.1.0.

## **1.5 Related Publications**

*North American Numbering Council (NANC) Functional Requirements Specification, Number Portability Administration Center (NPAC, Service Management System (SMS), Version 3.1.0, August 31, 2001.*

*North American Numbering Council (NANC) NPAC SMS Interoperability Interface Specification, Version 3.1.0, August 22, 2001.*

*ITU-T Recommendation X.290, OSI Conformance Testing Methodology and Framework for Protocol Recommendations For ITU-T Applications - General Concepts*, April, 1995.

*ITU-T Recommendation X.291, OSI Conformance Testing Methodology and Framework for Protocol Recommendations For ITU-T Applications - Abstract Test Suite Specification*, April, 1995.

*ISO/IEC 9646-5, Information Technology - Open Systems Interconnection - Conformance Testing Methodology and Framework - Part 5: Requirements on Test Laboratories and Clients for the Conformance Assessment Process*, Second edition, December 15, 1994. Also published as ITU-T Recommendations X.294.

## 2 The Testing Process

### 2.1 Interoperability and Regression Testing Guidelines

The LNPA WG recommends that ITP testing be performed on a SOA/LSMS Developer's software anytime that a change is made to the interface of either the NPAC SMS or the Developer's SOA/LSMS. In the event that the interface change is initiated by the NPAC SMS, the SOA/LSMS Developers shall perform ITP on each version of SOA/LSMS software that may potentially be used by Service Providers with the new NPAC SMS interface.

The following provides guidelines for specific scenarios and what level of testing is recommended for them:

1. When a local product (SOA/LSMS) is compiled with the current interface model, and a new local feature (SOA/LSMS feature) is implemented that DOES NOT involve a change in the use of the interface model, and the NPAC SMS is compiled with the current model, then no ITP testing is required.
2. When a local product is compiled with the current interface model, and no new local features implemented, and the NPAC SMS is compiled with the new interface model, then ITP testing is required [standard regression test cases - MOC only].

*This testing verifies the backwards compatibility of the interface.*

3. When a local product is compiled with the new interface model, and no new local features implemented, and the NPAC SMS is compiled with the new interface model, then ITP testing is required [standard regression test cases - MOC only].

*This testing verifies the backwards compatibility of the interface.*

4. When a local product is compiled with the new interface model, and new local features are implemented that involve the interface, and the NPAC SMS is compiled with the new interface model, then ITP testing is required [all appropriate standard regression test cases and new functionality test cases].

*This testing verifies the new software against the model and the NPAC SMS.*

5. When a local product is compiled with the current interface model, and new local features are implemented that involve the interface, and the NPAC SMS is compiled with the current model, then ITP testing is required [verify new feature - new functionality test cases]. Note: the regression test cases would have been addressed when the vendor upgraded the local product to the current version of the interface model.

For regression testing, the LNPA WG recommends that all service providers perform a minimum set of regression tests for all new releases.

### 2.2 Test Phases

The NPAC SMS Interoperability Test plan defines five phases of testing: Stack-to-Stack testing, Security testing, Managed Object Conformance testing, Association Management testing and Application-to-Application testing. Each phase can be completed separately but with the following constraints:

- Completion of Stack-to-Stack testing is a prerequisite to Security testing.



- Completion of Stack-to-Stack testing is a prerequisite to Managed Object Conformance testing.
- Completion of the Security check of cmipDepartureTime and sequence (Security Group A) is a prerequisite to MOC testing.
- Completion of Managed Object Conformance testing is a prerequisite to Association Management testing.
- Some parts of Security (namely signature checking i.e. Security Group B) testing may be postponed until after MOC or Association Management testing.
- Application-to-Application testing may not be started until the completion of the first four testing phases.

Before commencing the testing process with the Test Center, the Service Provider or SOA/LSMS Vendor must have completed CTS-3 testing. The S2S test consists of Transport Sanity Test, Association Establishment, Release and Abort Test. The Security test consists of two parts. The first part (Group A) tests basic authentication functions using the cmipDepartureTime, sequenceNumber, and systemId sub-fields of the access control field. The second part (Group B) of the security test consists of verification of the signature sub-field using hashing and MD5 encryption techniques using the listId and keyId from the accessControl structure. The MOC test verifies that all possible CMIP operations, i.e. m-get, m-set, m-create, m-delete, m-action, notification and confirmation may be performed for the respective MOs, as well as the various information model MO implementations i.e. support of attributes, correct name bindings, etc. The Association Management test covers Retry-Same-Host, Retry-Other-Host of the ACSE association establishment test, and the ability of the SUT to recover from time-outs, security violation and association loss. The A2A test examines the capability of the application on the SUT to carry out the transactions listed in the requirements.

### **2.3 Key Lists and Tunable Parameters**

A basic requirement for connecting to the NPAC SMS is the shared knowledge of a set of key lists that are referenced in the InpAccessControl data. A key from one of these lists is used to encrypt/decrypt the signature of the InpAccessControl structure. A single Key List will be generated and provided to all the service providers for the purposes of performing interoperability testing. The Service Provider must supply DSET with a Key List.

The values to be used for NPAC SMS tunable parameters are listed in the following table. If NPAC SMS tunable parameters and/or guidelines change, the updated values should be used.

<b>Table 2.1 - NPAC SMS Tunable Parameters</b>	
<b>(from Release 3.0.0 FRS, Appendix C)</b>	
<b>PARAMETER</b>	<b>VALUE</b>
<b>SOA Retry Interval</b>	15 minutes
<b>LSMS Retry Interval</b>	15 minutes
<b>SOA Retry Attempts</b>	1
<b>LSMS Retry Attempts</b>	1
<b>Maximum Subscription Query</b>	150 Objects Maximum

## 2.4 Test Case Description

The format for the ITP is as follows:

<b>Purpose</b>	The purpose of the test case for the system under test.
<b>Severity</b>	C – conditional must be run if the functionality is implemented by the product. O – optional depending on product implementation and vendor choice. R – required for basic LSMS and/or SOA functionality.  These values replace the existing severity 1, 2, 3, and 4 severity values.
<b>Severity Explanation</b>	Explanation of the test case severity assigned. In cases of conditional or optional severity, an explanation of what optional functionality that may implemented by the SOA or LSMS that would necessitate the execution of the test case.
<b>Prerequisites</b>	SOA, LSMS and/or the NPAC SMS Simulator prerequisites for execution of the test case. If the prerequisite is a test case, that test case should have been successfully completed.
<b>Procedure</b>	SOA, LSMS, and/or the NPAC SMS Simulator steps that must be followed for the execution of the test case.
<b>Expected Results</b>	Results expected for the SOA or LSMS under test.

### 2.4.1 Example

The following example shows a Stack-to-Stack test, initiated by the SOA to test for invalid security key detection. The test number (S2S.SOA.INV.ASSOC.INVK) is further described in the next sub-section.

<b>Table 2.2 - Test Case Example</b>	
<i>Test Id:</i>	S2S.SOA.PING
<i>Purpose:</i>	To verify that the IP layer is functioning properly.
<i>Severity:</i>	O
<i>Severity Explanation:</i>	Does not impact ability to provide service. No requirements for functionality. May be waived if System Software used does not support/provide a ping utility.
<i>Prerequisite:</i>	No association established between the SOA/LSMS and NPAC SMS Simulator.
<i>Procedure:</i>	1. SOA/LSMS issues a ping. 1. NPAC SMS Simulator responds to ping.
<i>Expected Results:</i>	Ping is successful.

## 2.5 Test Case Numbering

Test case numbers are the concatenation of a set of test descriptors that together uniquely identify the test being performed. Each descriptor making up a test number is separated from the next using a period. The general form of a test number is:

<Test Type>.<System Under Test (SUT) or Initiator for MOC only>.<Category>[.<Sub-category>].Operation[.<Sub-operation>]

The *Test Type* and *SUT/Initiator* descriptors are defined in the following tables. The *Category*, *Sub-category* (if present), *Operation*, and *Sub-operation* (if present), are described under the test type specific sections of this document.

Table 2.3 - ITP Test Types	
TYPE	DESCRIPTION
S2S	Stack-to-Stack Testing
SEC	Security Testing
MOC	Managed Object Conformance Testing
AMG	Association Management Testing
A2A	Application-to-Application Testing

Table 2.4 - SUT Symbols	
SUT/MOC-INITIATOR	DESCRIPTION
NPAC	NPAC SMS
SOA	Service Provider's SOA
LSMS	Service Provider's Local SMS

### 2.5.1 Example

A Stack-to-Stack test, initiated by the SOA to test for invalid security key detection will have the following test Identifier:

S2S.SOA.INV.ASSOC.INVK

In this case, the last three components (INV.ASSOC.INVK) identify the *Category*, the *Operation*, and *Sub-operation*. The category (INV) represents tests of invalid situations, the operation (ASSOC) represents an association test, and the sub-operation (INVK) identifies that this explicitly tests for invalid security key handling during association establishment.

## 2.6 Test Logs

Log files will be used as a mechanism for identifying specific problems with failed or inconclusive tests. The log files will include the Test Number followed by a sequence of PDUs exchanged during the test. The PDUs will be listed in ASN.1 Value Notation format. There will be one log file for each sequence of tests. As an example, all the NPAC SMS to LSMS Managed Object Conformance tests will share a single log file. The log file will contain information on all the test cases, not just those that failed. To avoid any

misunderstanding it is important to clarify that this log file is completely different from the log record objects defined in the GDMO file. Log record MO testing is outside the scope of this document. However, Log Record MOs may be created by a Service Provider on their respective SOA and/or LSMS in order to log the various events during testing.

## 2.7 **Test Reports**

Upon completion of the Interoperability Testing, the Test Lab Manager writes an Interoperability Test Report (ITR) documenting the successes and failures of the process. This test report will be provided to the Service Provider or the supplier building the system which was tested. The ITR will:

Identify the test cases used during the Interoperability Testing

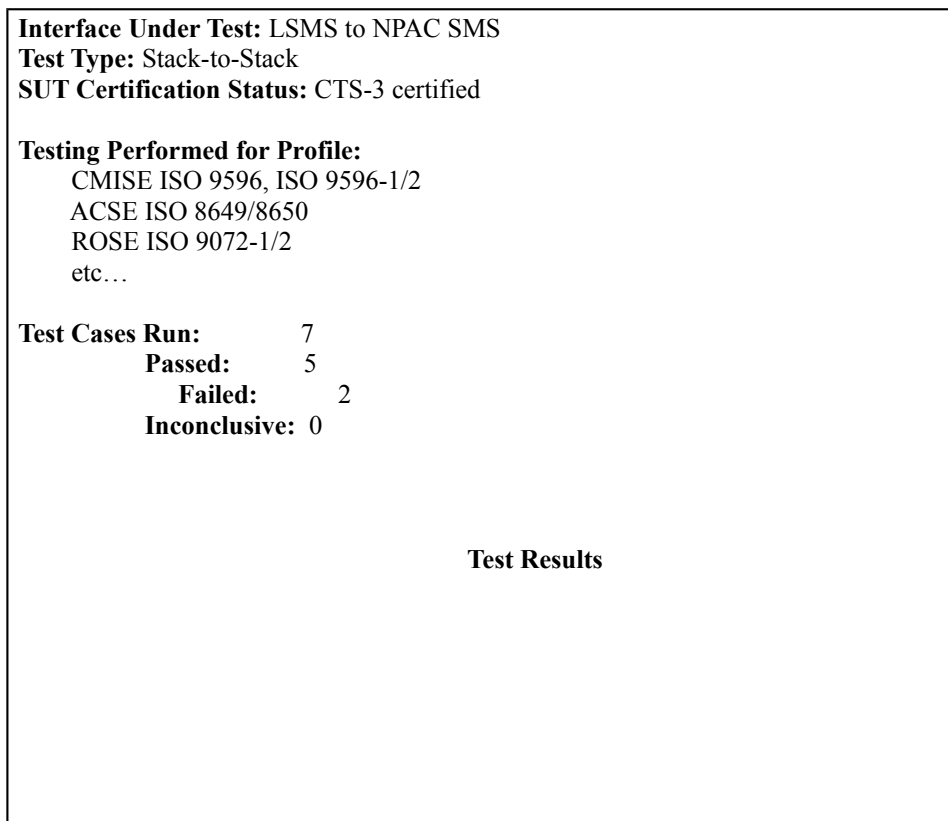
Identify the configuration of each test stage and the tools used

Identify the test cases passed

Identify test cases that failed and, if possible, the reason for failure

Identify test cases that were inconclusive

The test report will have a summary section that will sum up the results of a testing session. A sample of the ITR summary is shown in Figure-4:



**Figure 1 Test Report Layout**

The columns of the Test Result section of the report represent the following:

<i>Index</i>	An integer value, starting from 1 (one) and incrementing by 1 for each test performed. This is presented for convenience only, it has no significance beyond this
--------------	---

	report.
<i>Test Number</i>	The test case number.
<i>Result</i>	Either <i>PASS</i> , <i>FAILED</i> , or <i>INCONCLUSIVE</i>
<i>Reason</i>	If the test result is either <i>FAILED</i> or <i>INCONCLUSIVE</i> , this will reference the log file or an unsatisfied prerequisite.

## **2.8 Testing Considerations**

The NeuStar Interoperability Testing Service will be provided by a series of simulators collectively known as the NPAC SMS Simulator. The NPAC SMS Simulator will be used in different configurations in each phase of the interoperability testing. The NPAC SMS Simulator configurations are documented in the section on each phase of interoperability testing.

## **2.9 Conformance to Standards**

The ITP follows the general guidelines and principles depicted in the ISO/ITU standards for OSI Conformance Testing Methodology and Framework (X.290, X.294/ISO9646-5). For instance, the test cases listed in the ITP form an Abstract Test Method (ATM) since they describe how an Implementation Under Test will be tested independent of any specific realization of a Means of Testing. However, the test case descriptions provide enough details (down to the attribute value level) to enable abstract test cases to be specified for this test method.

The test cases presented in this document were designed to address a representative sample of the Managed Object Conformance Statements available in the IIS. Tests are provided for every mandatory requirement as well as a representative set of conditional and optional requirements listed in the MOCS. In addition, the testing campaigns described by the ITP consist of tests which are best represented by the three types of conformance testing defined by the standards, i.e. Basic Interconnection Tests (S2S and some Security), Capability Tests (MOC), and Behaviour Tests (Security, MOC, Association Management). The OSI Conformance Testing standards also call for a test report to be generated at the conclusion of testing. The test report depicted in the ITP is aligned with the template provided in X.294, and is intended to satisfy that requirement. Finally, the testing process as a whole has been designed in accordance with the standards recommendations. For example the TMN Test Center has a Test Lab Manager who will be responsible for all the issues relating to the lab itself and to the testing process.

## **2.10 Connectivity**

Connectivity information to the NPAC SMS Simulator is available from NeuStar.

### 3 Stack-to-Stack Interoperability Testing

#### 3.1 Overview

The SOA to NPAC SMS and LSMS to NPAC SMS interfaces are based on the services provided by Remote Operations Service Element (ROSE), Association Control Service Element (ACSE), and Common Management Information Service Element (CMISE) of the OSI application Layer. The Common Management Information Protocol (CMIP) carries the actual information to be exchanged.

The following table describes the OSI and RFC 1006 protocol stack profile used by this application:

Table 3.1 - The OSI and RFC 1006 Stacks Used for NPAC		
OSI LAYER	SERVICE/PROTOCOL	STANDARD
Application Layer	CMISE ACSE ROSE	ISO 9595/ITU-T X.710 ISO 9596-1/2, ITU-T X.711/X.712 ISO 8649/8650, ITU-T X.217/X.227 ISO 9072-1/2, ITU-T X.219/229
Presentation Layer	COPS, COPP ASN.1, BER	ISO 8822/8823, ITU-T X.216/226 ISO 8824/8825, ITU-T X.208/X.209
Session Layer	Kernel and Full Duplex	ISO 8326/8327, ITU-T X.215/X.225
Transport Layer	RFC1006, TP0, TCP	
Network Layer	IP	
Data Link Layer	PPP, MAC, Frame Relay, ATM, IEEE 802.3	
Physical Layer	DS-1, DS-0 x n, V.34	

The objectives of Stack-to-Stack interoperability testing are to ensure that the OSI stacks used by the NPAC SMS, SOA, and LSMS are compatible and to confirm that elementary connectivity exists between the NPAC and the SOA/LSMS. In addition, S2S testing provides the groundwork for the subsequent phases of testing.

S2S consists of simple TCP/IP connectivity and basic ACSE association setup and release tests.

#### 3.2 Requirements for Testing

##### 3.2.1 General Requirements

To establish a connection between the SOA and NPAC SMS or between the LSMS and NPAC SMS, the originating system should be capable of initiating and responding to the ACSE primitives. As a result, the Service Provider (or agent thereof) must have implemented all the SOA/LSMS initiated test cases and drivers identified in this section before any S2S test cases may be executed. Detailed ACSE and OSI layer parameters are listed in the next section of this document.

Prior to testing, the Service Provider or their Supplier must provide certain pertinent information to the TMN Test Center in order to prepare the Lab for testing their SOA or LSMS. The required information is described in the Testing Registration Form, which must be requested from NeuStar, Inc. A copy of that form is provided in *Appendix A* for reference purposes only.

Stack-to-Stack testing is independent of the application (SOA or LSMS). Thus, if the operating environment (Operating System, Hardware Platform, Stack) is the same for both SOA and LSMS, the Service Provider or SOA/LSMS Vendor need only perform this test series once. However, if the operating environments for the two systems differ in any way, the Service Provider or SOA/LSMS Vendor must perform this test series for both the SOA and LSMS operating environments.



### 3.3 ***Scope of Testing***

The scope of the stack-to-stack interoperability testing is to verify that the LSMS and SOA systems can establish an association with the NPAC SMS system based on an Association Control Service Element (ACSE) across a TCP/IP network. In doing so, the layers below ACSE (Presentation, Session, TCP/IP) are also being implicitly tested for interoperability.

The stack-to-stack testing is primarily a test of ACSE. It verifies that the OSI/TCP/IP stack can provide platform-to-platform connectivity, and that implementation-specific configuration parameters have been set correctly. However, ACSE testing may also involve verifying Functional Unit negotiation, and Access Control. That type of ACSE testing will be referenced as Security testing and will verify the correct implementation of the Access Control security requirements.

The stack-to-stack testing does not test the system types and association function components of the *InpAccessControl* structure (see section 4.3.2). Thus, only a single association is required for stack-to-stack testing, regardless of the actual number of associations the SOA/LSMS intends to use to connect to the NPAC SMS.

#### 3.3.1 Stack-to-Stack Testing Parameters

The following tables identify the communications parameters necessary to establish a connection between an SOA or LSMS and the NPAC SMS. The tables can be used to determine the values to be used when establishing associations with the A-ASSOCIATE ACSE Service Primitive. The tables do not reflect the NPAC SMS response to an A-ASSOCIATE request.

#### 3.3.2 NPAC SMS Simulator SAPs

Table 3.2 lists the Session selectors, Transport selectors, and NSAPs of the primary and backup NPAC SMS simulators. Note that the values for the S and T Selectors are examples only; actual values will be assigned prior to the start of testing. The S and T Selectors will be unique to differentiate the primary and the backup NPAC SMS Simulator since the NSAP will be the same for both primary and backup. The Presentation selectors will be different for every Service Provider/Supplier and will be supplied to the SP testing contact person prior to the start of testing.

<b>Table 3.2 - Presentation Service Access Points</b>			
<b>NPAC SMS</b>	<b>SSEL</b>	<b>TSEL</b>	<b>NSAP</b>
<b>Primary</b>	“pssel”	“ptsel”	540072872203208143039002
<b>Backup</b>	“bssel”	“bttsel”	540072872203208143039002

#### 3.3.3 Communication Parameters

<b>Table 3.3 - Communications Parameters</b>		
<b>PARAMETER</b>	<b>VALUE</b>	<b>COMMENT</b>
Protocol-Version	Version 1	Default
Application Context Name	{2 9 0 0 2}	Systems Management
Association-information	CMIPUserInfo	See section 4.3.1
Presentation-context-def-list	PCDL	See section 3.3.5
Version Number	Version 2	Functional Unit shall support Kernel and Duplex

### 3.3.4 NPAC Association Information

Table 3.4 - NPACAssociationInfo		
PARAMETER	VALUE	COMMENT
errorCode	success (0), access-denied(1), retry-same-host(2), or try-other-host(3)	
errorText	GraphicString(SIZE(1..80))	

### 3.3.5 Presentation Context Definition List

Table 3.5 - Presentation Context Definition List				
ABSTRACT SYNTAX		TRANSFER SYNTAX		PRESENTATION CONTEXT IDENTIFIER
NAME	VALUE	NAME	VALUE	
ACSE	{2 2 1 0 1}	BER	{2 1 1}	Any unique, odd integer (1)
SMASE	{2 9 0 1 1}	BER	{2 1 1}	Any unique, odd integer (5)
CMIP	{2 9 1 1 4}	BER	{2 1 1}	Any unique, odd integer (3)
Access Control	{Inp-attribute 1}	BER	{2 1 1}	

## 3.4 ***Assignment of Responsibilities***

All associations between the SOA/LSMS and NPAC SMS are initiated by either the SOA or LSMS. The NPAC SMS never initiates an association request. Thus, the tests identified in this document for Stack-to-Stack testing shall be initiated from the SOA and LSMS. A test report will be produced for each test performed. However, correct responses to the ACSE service requests must be verified by the Service Provider or agent thereof.

## 3.5 ***Definition of Tests***

This section describes the types of tests to be performed. A complete list of the test cases is listed in *Appendix C*. The details of the test cases start in *Chapter 9* of this document. There are two subgroups of Stack-to-Stack testing: TCP/IP and valid ACSE Tests. These are described in the sections below.

A Service Provider or SOA/LSMS Vendor may elect to execute the Security Tests at the same time while performing the S2S test cases. This is accomplished by requesting that the Test Center enable the security features of the S2S simulator. As a result the NPAC SMS simulator will examine the access control field of any ACSE or ROSE PDU. The Service Provider or SOA/LSMS Vendor may choose to perform all or some (Group A) of the security test cases at this time. Note that Group A of Security is a prerequisite to MOC testing and that any given SOA/LSMS will have to eventually pass all the Security test cases before it can be connected to the real NPAC.

### 3.5.1 TCP/IP Layers Tests

In order to ensure that the underlying TCP/IP layers are functioning properly, two basic connectivity tests will be performed prior to any ACSE tests. First the IP layer will be tested using “ping”. This test case will not be a required prerequisite to ACSE tests since some systems might not have a ping function available. The ping test case will have the test-Id <S2S.SOA.PING> for SOA systems and <S2S.LSMS.PING> for

LSMS systems under test. The other test case will validate the operation of the TCP layer and involves using FTP to log into the NPAC SMS. This test case will have the following test-Ids <S2S.SOA.FTP> and <S2S.LSMS.FTP> for the SOA and LSMS interfaces respectively. Passing the FTP test case will be a prerequisite to any ACSE tests since the requirements specify that FTP must be supported.

### 3.5.2 Valid ACSE Tests

The Valid ACSE tests check an application's ability to establish, release, and abort associations with well-formed PDUs. These tests rely on the SOA or LSMS initiating the association establish request. The valid stack-to-stack tests will consist of association request, release, and abort by the SUT and association abort by the NPAC simulator. These tests will use well-formed ACSE PDUs.

## 4 Security Interoperability Testing

### 4.1 Overview

The Security Test is based on verifying the `InpAccessControl` attribute which is included in every ACSE and CMIP message exchanged over the interface. The Security test is subdivided into two parts. The first part referenced by Security Test Group A verifies proper uses of all the items in `InpAccessControl` except for the signature field. The second part of Security Tests examines the signature field in the Access Control Attribute for both ACSE and CMIP messages. This set of test cases will be known as Group B. Security Tests may performed in whole or in part (Group A) at the same time as S2S tests. Note that only Group A of the Security test cases is a required prerequisite of the MOC test phase. Also note that the NPAC simulator does not require the encryption software on the SUT to be either deactivated or not implemented in order to run Group A of the security tests. Whether an ACSE association or a CMIP request will be granted or not will depend on the values used in the Access Control attribute in the respective PDUs. For the details of the security requirements of the SOA/LSMS to NPAC SMS interface, please refer to section 5.2 Security of the NPAC SMS Interoperable Interface Specification document.

### 4.2 Requirements for Testing

Satisfactory completion of S2S testing is a pre-requisite for the Security Test. If a Service Provider or SOA/LSMS Vendor elects to defer the signature field Group B test, he/she may do so by asking the Test Center to disable the signature check during the Security Test. However, the Service Provider or SOA/LSMS Vendor must pass all the Security test cases in order to satisfy the security requirements of the interface.

Prior to testing, the Service Provider or SOA/LSMS Vendor must inform the Test Center of the value to be used for the `systemId` component of the `InpAccessControl` structure. This value will be a Service Provider ID value. This value will be supplied as one of the entries of the Testing Registration Form presented in *Appendix A*.

#### 4.2.1 LNP Access Control Attribute

The ASN.1 type of `InpAccessControl` field is shown below. Please refer to chapter 5 of the NPAC SMS Interoperable Interface Specification document for further details.

```
InpAccessControl ::= SEQUENCE {
    systemId          [0] SystemID,
    systemType       [1] SystemType,
    userId           [2] GraphicString60 OPTIONAL,
    listId           [3] INTEGER,
    keyId            [4] INTEGER,
    cmipDepartureTime [5] GeneralizedTime,
    sequenceNumber   [6] INTEGER (0...4294967295),
    function         [7] AssociationFunction,
    recoveryMode     [8] BOOLEAN,
    signature        [9] BITSTRING
}
```

```
ServiceProvID ::= GraphicString4
```

```
SystemID ::= CHOICE {
    serviceProvID [0] ServiceProvId,
    npac-sms     [1] GraphicString60
}
```

```
SystemType ::= ENUM {
    soa(0),
    local-sms(1),
}
```

```

    soa-and-local-sms(2),
    npac-sms(3) --value is only valid for AccessControl definition
}

AssociationFunction ::= SEQUENCE {
    soaUnits [0] SoaUnits,
    lsmsUnits [1] LSMSUnits
}

SoaUnits ::= SEQUENCE {
    soaMgmt [0] NULL OPTIONAL,
    networkDataMgmt [1] NULL OPTIONAL
}

LSMSUnits ::= SEQUENCE {
    dataDownload [0] NULL OPTIONAL,
    networkDataMgmt [1] NULL OPTIONAL,
    query [2] NULL OPTIONAL
}

```

### 4.3 Scope of Testing

The Security Test will verify that the exchanged PDUs contain the correct values for systemID, listId, keyId, cmipDepartureTime, sequenceNumber and signature in the Access Control field. It will not verify the userId, function, and recovery mode. Outside the scope of this testing is to examine the ability of the SUT to recover from a detected security breach which will be handled by the Association Management Testing phase. Furthermore, as mentioned previously, it is outside the scope of Interoperability Testing to address M&Ps related to interoperating with the NPAC SMS. However, it may be necessary for the NPAC Simulator to perform functions related to M&Ps in order to communicate with the System Under Test (SUT). For example, an SUT may require the NPAC Simulator to exchange key list acknowledgement files before Security testing can begin. If necessary, the NPAC Simulator will generate the file and exchange with the SUT as appropriate. However, this does not constitute testing of the M&P and thus does not ensure that the SUT will be able to interoperate with the NPAC SMS regarding that particular M&P. The following tables list the allowed values for CMIP User Information and Access Control.

#### 4.3.1 CMIP User Information

Table 4.1 - CMIPUserInfo		
PARAMETER	VALUE	COMMENT
Protocol-Version	Version 2	Default
Functional Units	Kernel, Multiple Object Selection, Multiple Reply	
Access Control	LnpAccessControl	OID={lnpAttribute 1}; See section 4.3.2
UserInfo	NpacAssociationInfo	OID={LNP-ASN1}; See section 3.3.4

#### 4.3.2 Access Control

Table 4.2 - LnpAccessControl		
PARAMETER	VALUE	COMMENT
SystemId	SPID	
SystemType	soa(0), local-sms(1),	

Table 4.2 - LnpAccessControl		
PARAMETER	VALUE	COMMENT
	soa-and-local-sms(2), npac-sms(3)	
userId	SOA User UserId	Optional - SOA
listId		
keyId		
cmipDepartureTime	System Time	GMT
sequenceNumber	0	Always zero on association establishment
signature	encrypted	
function	dataDownload, networkDataMgmt, query, soaMgmt	LSMS and SOA Functional Units
recoveryMode	TRUE, or FALSE	

#### 4.4 Assignment of Responsibilities

All associations between the SOA/LSMS and NPAC SMS are initiated by either the SOA or LSMS. The NPAC SMS never initiates an association request. Thus, the tests identified in this document for Security testing shall be initiated from the SOA and LSMS. A test report will be produced by the Test Center for each test performed. However, correct responses to the ACSE service requests must be verified by the Service Provider or agent thereof.

#### 4.5 Definition of Tests

There will be two main categories of test cases in both GroupA and GroupB of the Security Test. They are Valid and Invalid tests. Before starting the security test, the system clocks of the NPAC SMS simulator and the SOA and/or LSMS systems under test will be synchronized manually to meet the 5 minutes time variance requirement. Testing the implementation of NTP (Network Time Protocol) is outside the scope of this Test Plan, since it is left up to the Service Providers or SOA/LSMS Vendors to select their own time source. However, the cmipDepartureTime field in the accessControl attribute will be checked in every PDU to verify that it is within the 5 minutes tolerance. The detailed description of all the security test cases is presented in *Chapter 10*.

##### 4.5.1 Valid Security Test

The Valid Security Tests verify the capability of the SUT to issue well formed ACSE PDU's containing valid values for the systemId, listId, keyId, cmipDepartureTime, sequenceNumber and signature fields of the LnpAccessControl attribute. The ability of the SUT to generate valid LnpAccessControl CMIP PDU's will be exercised during MOC testing and is not duplicated here. There are two Valid Security test cases (one per Group), one for the A-associate request without RSA, and the other with RSA signature encryption. Only one valid security test case will need to run depending on whether encryption is implemented or not, i.e. the SUT will not be required to de-activate encryption if it is already implemented in order to run the NOSIG test case. That test case is only available for the convenience of the Service Providers or their SOA/LSMS suppliers. The Test Ids for this test category will have the test number prefix of **SEC.<Initiating System>.VAL**, where *Initiating System* has a value of either SOA or LSMS.

##### 4.5.2 Invalid Security Tests

Invalid Security Tests are performed to ensure that associations are not established by the SOA/LSMS under compromised security conditions. The security of the system is considered compromised if any of the fields in the LNP Access Control Structure are invalid. The objective of these tests is to verify the SOA's and LSMS's interoperability with the NPAC SMS, and not the other way around. Thus, Invalid ACSE

Security Tests require the SUT to issue correctly formed association requests for which the NPAC simulator will respond with well formed ACSE association response PDUs containing an invalid field in the access control structure. The SUT is expected to react according to the requirements specified in the IIS, that is aborting the association without a reason given. This category also include one CMIP PDU test case which examines the ability of the SUT to reject a Notification from the NPAC simulator which contain an invalid signature field. Also included in Group B is a set of CMIP operations test cases targeted assessing the SOA and LSMS's ability to handle security threats. Note that these test cases do not require the SUT to recover from the security threats. The SUT will pass this testing phase if it is able to detect the security violation and abort the association for all the invalid test cases. Testing the capability of the SUT to recover from those conditions, i.e. re-establishing an association with a new Key and List Ids, is handled by the Association Management test cases in section 6 of the ITP. The Test Ids for this test category will have the test number prefix of **SEC.<Initiating System>.INV**, where *Initiating System* has a value of either SOA or LSMS.

## 5 Managed Object Conformance Interoperability Testing

### 5.1 Overview

Managed Object Conformance tests check the ability of the SOA/LSMS to communicate information and instructions concerning specific Managed Object Classes to the NPAC SMS. Determining the results of these tests requires prior knowledge of the objects under test, and a detailed analysis of the messages exchanged between the SOA/LSMS and the NPAC SMS. This knowledge is expressed in the GDMO Information Model defined in the NPAC SMS Interoperable Interface Specification.

Furthermore, these tests check for the ability to interrogate and manipulate objects residing in a remote environment. The purpose is to ensure that the Manager can initiate, and the Agent can assimilate, syntactically and semantically, valid requests, and responds to such requests according to the NPAC SMS specification.

Throughout this section, the term *Manager* is used to represent either the SOA or the manager role of the NPAC SMS or LSMS. The term *Agent* is used to represent either the agent role of the LSMS or that of the NPAC SMS. The NPAC SMS will use the naming attribute value defined in IIS for the real NPAC SMS, i.e. "Midwest Regional NPAC SMS".

### 5.2 Requirements for Testing

#### 5.2.1 General Requirements

Successful completion of the Stack-to-Stack and Security Group A interoperability testing and its prerequisites are required before embarking on the Managed Object Conformance testing phase. Also, the Service Provider or SOA/LSMS Vendor wishing to sign up for testing must submit completed MOCS, SCS, ICS, or IXIT reference documents (See *Appendix A*) at least one week prior to the start date of MOC testing. This allows the Test Lab to determine the testing configuration and needs of that NPAC Client.

#### 5.2.2 Order of Tests

Due to the nature of the containment hierarchy, tests must be performed in the order in which they are specified in the respective test case sections of this document, with the test cases pertaining M-DELETE operations to be executed at the end of the MOC test phase. However, the following test case sets are independent of each other:

SOA Manager to NPAC SMS Agent Managed Object Conformance Test Cases.

LSMS Manager to NPAC SMS Agent Managed Object Conformance Test Cases.

NPAC SMS Manager to SOA Agent Managed Object Conformance Test Cases.

NPAC SMS Manager to LSMS Agent Managed Object Conformance Test Cases.

#### 5.2.3 Association Type

Prior to commencing Managed Object Conformance testing, the Service Provider or SOA/LSMS Vendor must supply the TMN Test Center with the configuration of the association to be used in connecting to the NPAC SMS. The NPAC SMS simulators currently support a single association per simulator instance. The type and functions (see 4.3.2 - *systemType*, and *function*) of the association used by the SUT must be communicated to the NeuStar Interoperability Testing Service Administrator via the Testing Registration Form depicted in *Appendix A*.



### **5.3 Scope of Testing**

The main focus of Managed Object Conformance testing is to check conformance to the Managed Object Conformance Statements (MOCS) contained in the NPAC SMS Interoperable Interface Specification, Chapter 9. The test cases will exercise those capabilities that can be derived directly from the GDMO Information Model. It is not the purpose of these tests to check sequences of operations as expressed in the Operational Flows listed in Appendix B of the IIS.

One of the objectives of the MOC Test will be to ensure that the NPAC is unable to adversely affect a local carrier's system, and vice versa.

### **5.4 Assignment of Responsibilities**

The SOA and LSMS have a peer-to-peer relationship with the NPAC SMS. Thus, the SOA and LSMS perform both a Manager and Agent role from a CMIP perspective. The implementation of the Managed Object Conformance tests is the joint responsibility of the SOA and LSMS vendor and the TMN Test Center. SOA and LSMS initiated tests are the responsibility of the Service Provider or agent thereof, and NPAC SMS initiated tests are the responsibility of the Test Center.

Completion of the tests outlined in this section will be documented by the Test Center. However, determination of success for SOA or LSMS initiated tests will be left to the performer of the tests. Thus, interoperability testing is limited to the correct processing of request and responses by the NPAC SMS Simulator.

### **5.5 Definition of Tests**

#### **5.5.1 Capability Tests**

Capability tests provide limited testing of the observable capabilities of the Manager/Agent regarding static conformance requirements. These tests check the existence and basic validity of the specified capabilities. The results of these tests and the behavior tests form the basis for claims of conformity. This test group consists of Operation and Notification tests.

Capability Tests will assume the test number descriptor value of **CAP**. All tests in this category will start with the prefix **MOC.<Initiating System>.CAP**, where <Initiating System> is either SOA, LSMS, or NPAC.

##### **5.5.1.1 Object Operation Tests**

Operation tests check the Manager's ability to initiate CMIP requests and handle their results and the Agent's ability to respond to a manager's requests and report the events described in the NPAC SMS specifications. These tests check only the syntax and semantics of the messages exchanged between the Manager and Agent; neither the Manager nor Agent behavior is checked. The test cases are developed by reviewing the MOCS and the GDMO Behavior Definitions for each Managed Object Class to determine the valid set of CMIS requests that can be issued to that Managed Object Class. For each operation identified, a test is formulated to initiate the operation with a complete, well-formed CMIP PDU.

As an example, the *InpLocalSMS* object class, which has a single attribute other than those it inherits from *top*, supports only the M-GET and M-ACTION operations across the NPAC SMS Manager to LSMS Agent interface. Thus, for this Managed Object Class, two Object Operation Capability tests will be implemented:

A single M-GET test to check that the *nameBinding*, *managedObjectClass*, and *InpLocal-SMS-Name* can be retrieved.

A *InpRecoveryComplete* M-ACTION (which has no user data in the request) to test that a reply is returned (to the LSMS or SOA) with all mandatory data items.

Test numbers in this category will have the prefix **MOC.<Initiating System>.CAP.OP**.

### 5.5.1.2 Object Notification Tests

Notification tests check the Manager’s ability to handle correctly notifications produced by the Agent entity, and the Manager’s ability to reply to the notifications appropriate to the NPAC SMS Interoperable Interface Specifications. They also verify the Agent’s ability to issue valid M-EVENT-REPORT requests to the Manager entity and handle the Manager responses to those requests according to the NPAC SMS Interoperable Interface Specifications.

Test numbers in this category will have the prefix **MOC.<Initiating System>.CAP.NOT**.

With Release 3.1.0, the Service Provider SOAs have the option of supporting either “individual” notifications for subscription version processing or “range/list” notifications. If a SOA will be supporting both notifications, test cases using the subscription version notifications must be run for both versions.

The “list” portion of a “range/list” notification will only be sent if the subscription version ids are non-consecutive. If the range/list notifications are supported and a single subscription version is involved, a “range” of one subscription version will be sent as opposed to a “list” of one.

Below is a list of each type of SOA to NPAC SMS subscription version/InpSubscription notification.

Individual Notifications	Range/List Notifications
Attribute Value Change	subscriptionVersionRangeAttributeValueChange
subscriptionVersionCancellationAcknowledgeRequest	subscriptionVersionRangeCancellationAcknowledgeRequest
subscriptionVersionDonorSP-CustomerDisconnectDate	subscriptionVersionRangeDonorSP-CustomerDisconnectDate
subscriptionVersionNewSP-CreateRequest	subscriptionVersionRangeNewSP-CreateRequest
subscriptionVersionNewSPFinalCreateWindowExpiration	subscriptionVersionRangeNewSPFinalCreateWindowExpiration
Object Creation	subscriptionVersionRangeObjectCreation
subscriptionVersionOldSP-ConcurrenceRequest	subscriptionVersionRangeOldSP-ConcurrenceRequest
subscriptionVersionOldSPFinalConcurrenceWindowExpiration	subscriptionVersionRangeOldSPFinalConcurrenceWindowExpiration
subscriptionVersionStatusAttributeValueChange	subscriptionVersionRangeStatusAttributeValueChange

Below is a list of the remaining notifications.

Notification Name	Interface
InpNPAC-SMS-Operational-Information	SOA to NPAC SMS LSMS to NPAC SMS
subscriptionAudit-DiscrepancyRpt	SOA to NPAC SMS
subscriptionAuditResults	SOA to NPAC SMS
subscriptionNewNPA-NXX	SOA to NPAC SMS LSMS to NPAC SMS
subscriptionVersionLocalSMS-ActionResults	NPAC SMS to LSMS

## 5.5.2 Behavior Tests

Behavior tests check an implementation as thoroughly as practicable over the full range of dynamic conformance requirements. Since the number of possible combinations of events is astronomical, such testing can't be exhaustive. There are several types of Behavior tests as listed in the following subsections.

### 5.5.2.1 Valid Tests

Valid behavior tests check the Manager's ability to initiate correctly with, and respond to, syntactically and semantically valid requests and responses. They also check the Agent's ability to handle a Manager's valid requests and respond to them appropriately. This type of testing uses several mechanisms to provide adequate testing coverage. The testing mechanisms are represented by test groups in the test suite structure that divide the valid behavior test group accordingly. Each of these mechanism test groups is then subsequently structured.

Valid Tests go beyond those defined within the Capability Tests to provide a more complete set of tests that check optional components as well as mandatory components. This test group will have the test number prefix of **MOC.<Initiating System>.VAL**.

Example test categories are:

Create of Object Instance (use a Get to check that the object now exists in the Agent).

Create of Object Instance by reference

Create of Object Instance using Auto Naming

Get a group of objects within a scope

Set a single attribute (Use a Get to determine whether the Set has taken effect in the Agent).

Set multiple attributes

Delete an Object

Delete objects specified within a scope request

### 5.5.2.2 Invalid Tests

Invalid behavior tests check the Manager and Agent's ability to respond correctly to syntactically or semantically invalid object messages. These tests are divided into Invalid Semantics, Invalid Syntax, and Inopportune Behavior tests. Inopportune Behavior tests are considered outside the scope of this document. All tests in this subgroup will have the test number prefix of **MOC.<Initiating System>.INV**. A SOA or LSMS Manager entity that is being tested will not be required to initiate invalid requests. However the error handling capabilities of that Manager will be tested by the NPAC SMS Agent emulator responding with invalid and error response PDUs to valid Manager requests. In the case where an Agent entity (LSMS) is under test, the NPAC SMS Manager emulator will verify this Agent error handling abilities by sending it semantically and syntactically invalid requests. The expected result for this type of test cases will be for the SUT to detect the error and report it back to the NPAC simulator using one of the valid CMIP error codes. CMIP Error Response PDUs as well as CMIP Reject PDUs will be accepted in return for any invalid requests. Note that in the cases where more than one CMIP error code applies, the return of any single and reasonable (e.g. processingFailureEr instead of setListErrorEr) code will be considered a successful test result.

#### 5.5.2.2.1 Invalid Semantics

These tests check the Manager's and Agent's ability to respond to Protocol Data Units (PDUs) containing invalid parameter values, invalid combinations of parameter values, or inappropriate requests. For example, for the LSMS implemented Managed Object Class of *ImpLocalSMS*, tests would address the following:

the LSMS responds with an error to an M-CREATE request,  
the LSMS responds with an error to an M-SET request,  
the LSMS responds with an “Invalid Attribute” error ,  
etc.

There are two categories of these tests – generic, which would apply to all the Managed Object Classes in the Information Model, and MO specific invalid tests. Examples of the generic type of invalid tests are:

Create object that already exists

Delete non-existent object

Complexity Limitation tests (getting a group of more than 50 objects)

Invalid attributes

#### 5.5.2.2.2 Invalid Syntax

These tests check the Manager and Agent’s ability to respond to syntactically invalid PDUs. A PDU is syntactically invalid if it does not conform to the ASN.1 specification for the application messages and object classes, as well as the Basic Encoding Rules (BER).

Example test types are as follows:

Set an attribute to an invalid enumerated value of an enumerated type

Set an attribute to an invalid string value

Set an integer attribute beyond its range

Set an attribute to an invalid type (i.e. set an Integer typed PDU to a String)

#### 5.5.2.3 Name Binding Tests

Name Binding tests check for the system behavior under various name binding variations - both correct and incorrect. The current definition of the NPAC SMS Interoperable Interface Specification does not require the definition and implementation of Name Binding tests.

#### 5.5.2.4 Boundary Tests

Boundary condition tests check to verify the behavior of systems over maximum and minimum attribute values for each object class. As an example, if an integer attribute has a range specified, a test will check that the lowest and highest values within that range can successfully be established in a set request. Test numbers for tests in this category have test numbers prefixed with **MOC.<Initiating System>.BND**.

## **6 Association Management Interoperability Testing**

### **6.1 Overview**

Association Management tests check the association recovery capabilities of the SOA and LSMS under an abnormal or erroneous operating environment. The tests are divided into the following sub-categories:

Retry same/other host upon association establishment

Response time tests

Security Violation tests

Loss of Association

NPAC SMS Down

### **6.2 Requirements for Testing**

#### **6.2.1 General Requirements**

Successful completion of the Managed Object Conformance testing and its prerequisites are required before starting the Association Management test phase. Also, a completed Testing Registration Form must be submitted to NeuStar, Inc prior to testing.

### **6.3 Scope of Testing**

Association Management Testing is limited to those tests that can be performed without knowledge of the application. This does not address issues such as MIT recovery.

### **6.4 Assignment of Responsibilities**

All associations between the SOA/LSMS and NPAC SMS are initiated by either the SOA or LSMS. The NPAC SMS never initiates an association request. Thus, the tests identified in this document for Association Management testing shall be initiated from the SOA and LSMS. A test report will be produced by the Test Center for each test performed. However, correct responses to the ACSE service requests must be verified by the Service Provider or agent thereof.

### **6.5 Definition of Tests**

#### **6.5.1 Retry Same/Other Host**

These tests verify that the LSMS and/or SOA are able to re-send the ACSE A-Associate request to the appropriate host address when the NPAC SMS responds with an A-Associate reject containing one of the error codes “retry same host” or “retry other host “. For “retry other host”, the SUT is expected to re-issue the A-Associate request with the backup selectors and NSAP listed in Table 3.2.

#### **6.5.2 Security Violation Tests**

These tests check to see if the SOA and LSMS correctly detect and recover from security violations. The term “recover from” means abort the association, log the security violation, and (under certain conditions) re-establish the association using alternative security keys.

### 6.5.3 Loss of Association Tests

These tests check to see if the SOA and LSMS correctly react to the loss of an association between the SOA/LSMS and the NPAC SMS. The behavior of the SOA/LSMS with respect to association loss is documented in section 5.3 of the NPAC SMS Interoperable Interface Specification.

### 6.5.4 NPAC SMS Down Tests

This group of tests checks to see if the SOA and LSMS perform the correct sequence after detecting that the NPAC SMS is down. The behavior of the SOA/LSMS with respect to NPAC SMS downtime is documented in section 5.3 of the NPAC SMS Interoperable Interface Specification. This test group requires the provision of a second stack and NPAC SMS simulation environment. These tests will be accomplished by enabling a second instance of the NPAC SMS simulators on the same testing platform available in the TMN Test Center.

## 7 Application to Application Interoperability Testing

### 7.1 Overview

The objective of Application-to-Application testing is to verify that the applications implemented by the Service Providers, or the agents of service providers, on their respective SOA and LSMS systems satisfy the requirements for such applications as listed in the IIS and FRS. Determining the results of these tests requires prior knowledge of the scenarios under test, and a detailed analysis of the transactions exchanged between the SOA/LSMS and the NPAC SMS. This knowledge is expressed in the NPAC SMS Interoperable Interface Specification, and the Functional Requirement Specifications.

In contrast to a MOC testing campaign which addresses the behavior of a given MO in the context of a specific CMIP request, an A2A testing campaign will address complete transactions where each test case will require multiple CMIP operations occurring in a specific order, and may span multiple MO classes and scenarios as per Appendix B of the IIS. The utility of such testing will be underlined by the fact that the error handling capabilities of the SUT will be paid special attention during that campaign because of the flexibility afforded by the NPAC simulators. Every transaction (e.g. Action to create a subscription version) will have one test case representing a sunny day scenario where all operations proceed successfully according to the scenarios listed in the IIS. This will be true as long as that transaction's A2A test case does not duplicate a pre-existing MOC test case. Also, a given transaction will have at least one and probably more rainy day scenarios where one or more of the steps listed in the scenarios of the IIS, are dropped or executed with an intentional error in order to test the SOA/LSMS applications ability to detect and possibly cause the correction of such errors. The error handling capabilities which will be tested are those listed in the IIS and/or FRS or agreed to by the SP committee and subsequently incorporated in the IIS.

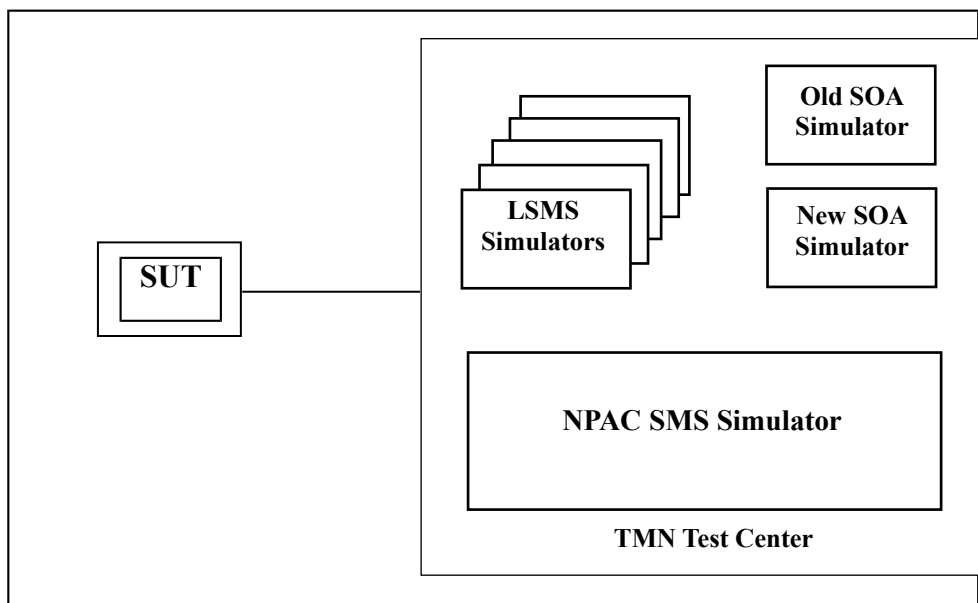


Figure 2 NPAC SMS Simulators for A2A testing

Other factors which guided the development of the App-to-App test plan are as follows:

- Additional behavior beyond that specified in the GDMO MOCS tables is required.
- The NPAC simulators must coordinate the behavior required by more than one MO at a time responding to, or initiating a transaction-dependent set of CMIP operations.
- The NPAC simulators must demonstrate the real NPAC's behavior as visible by more than one local carrier interface. E.g., some App-to-App test cases involve interactions between the NPAC and one or more SOAs (new, old, donor) and between the NPAC and one or more LSMSs. In this

case the NPAC simulators will emulate the system(s) which are not under test as depicted in Figure 5.

- The App-to-App test cases must allow an SP to test their SOA and LSMS independently where the NPAC simulators will emulate the behavior of the other system which is not being tested.

Throughout this section, the term *Manager* is used to represent either the SOA or the manager role of the NPAC SMS or LSMS. The term *Agent* is used to represent either the agent role of the LSMS or that of the NPAC SMS.

## **7.2 Requirements for Testing**

### **7.2.1 General Requirements**

The A2A testing phase is a required prerequisite to Turn-up testing. A2A will concentrate the testing effort on the application as a whole (MOC tests the parts) and specifically on its ability to handle the error conditions and inopportune behaviour that it may encounter in real operating conditions. Successful completion of the all the prior ITP test phases (S2S, SEC, MOC, AMG) is required before embarking on the Application to Application testing phase.

### **7.2.2 Order of Tests**

Due to the nature of the transactions implemented by this interface, tests must be performed in the order in which they are specified in the respective test case groups of this section. The App-to-App test cases will address the following scenario groups (transactions):

1. Audit scenarios.
2. Service Provider and Network Data scenarios
3. Subscription Version scenarios.
4. Miscellaneous scenarios.

## **7.3 Scope of Testing**

The main focus of App-to-App testing is to check the implementation of the application behaviour specified in the requirements (IIS and FRS). The test cases will exercise those capabilities that can be derived directly from the GDMO Information Model and the Scenarios of the Message Flow Diagram section of the IIS. The ability of an application under test to handle the error conditions and inopportune behavior of the NPAC SMS and other simulated systems (i.e. a simulated old SOA, etc...) will be verified. This test plan does not require a service provider to cause its own SOA and/or LSMS to behave incorrectly or to be inconsistent with the requirements. However, if an SP elects to test such behavior, special arrangements may be made by the TMN Test Center to accommodate that request. It is outside the scope of App-to-App testing to verify a SUT's behaviour under real life operating conditions such as those offered by Turn-Up testing, i.e. the presence of multiple SOAs and LSMSs testing with a single NPAC SMS, or the use of Timers and Tunable Parameters as specified by the requirements. Finally, the Mass Update and NPA-NXX Split scenarios are outside the scope of the ITP since they involve processing internal to the NPAC which will not introduce any special CMIP behavior across the interface. For instance, the Mass Update will result in M-SET requests to the LSMS and attribute value change notifications to the SOA. This type of CMIP exchanges is covered by many MOC and A2A test cases. The NPA-NXX Split does not involve the exchange of any CMIP PDUs across the interface.

## **7.4 Assignment of Responsibilities**

The implementation of the Application to Application test plan is the joint responsibility of the LSMS and/or SOA vendor and the TMN Test Center. LSMS/SOA initiated tests are the responsibility of the Service Provider or agent thereof, and NPAC SMS initiated tests are the responsibility of the Test Center.



Completion of the tests outlined in this section will be documented by the Test Center. However, determination of success for SOA or LSMS initiated tests will be left to the performer of the tests. Thus, App-to-App testing is limited to the correct processing (as per the IIS) of a transaction by the LSMS/SOA SUTs and the NPAC SMS Simulators.

**7.5 Definition of Tests**

The App-to-App test cases are defined in such a way as to address a given SOA or LSMS application singly. Test cases which target both the SOA and LSMS of a given service provider have not been defined here. The NPAC simulators will emulate all the other systems (Old/New SOA, LSMSs) required by a test case for a given SUT.

**7.5.1 Valid Behavior Tests**

Valid behavior test cases are designed to verify the capability of an application (SOA/LSMS) to correctly process a LNP transaction from start to finish. A typical valid behavior test case will execute the CMIP operations that constitute a complete transaction and will address the expected “positive” results of that CMIP sequence of requests and responses. A test case may span one or more of the scenarios listed in Appendix B of the IIS. For example, in order to create a new subscription version on the NPAC SMS by the new service provider, the set of CMIP requests and responses listed in the following scenarios of Appendix B of the IIS will be performed:

SubscriptionVersion Create by the Initial SOA (New Service Provider)

SubscriptionVersion Create by Second SOA (Old Service Provider)

Note that the Old Service Provider’s SOA as well as any internal NPAC SMS processing (i.e. local M-CREATE) will be simulated by the NPAC simulators. If a Service Provider elects to test their LSMS at the same time, testing proceeds with the Activation by new SP and Active Subscription Version Create on LSMS scenarios. The expected result of that test case is the successful completion of the transaction to create a new subscription version. An example of a test case that addresses only one scenario is a SOA Initiated Audit carried out from the first CMIP request (i.e. the M-Create of the subscriptionAudit) to the last notification generated by that audit (i.e. M-Event-Report for object deletion). Again the focus here would be the successful assimilation of the Audit Results by the SOA.

Application to Application test cases for valid behavior will have the type A2A and the test number descriptor value of VAL. All the test identifiers for this category will start with the prefix **A2A.<System Under Test>.VAL**, where the possible values of <System Under Test> are shown in Table 7.1.

<b>Table 7.1 - System Under Test</b>	
<b>SUT</b>	<b>DESCRIPTION</b>
SOA	Service Provider’s SOA
OSOA	Old Service Provider’s SOA
NSOA	New Service Provider’s SOA
DSOA	Donor Service Provider’s SOA
NPAC	NPAC SMS
LSMS	Service Provider’s LSMS

## 7.5.2 Inopportune Behavior Tests

Inopportune or invalid behavior tests are designed to verify the capability of the system under test (SOA/LSMS) to correctly detect and handle the error conditions described in the requirements. Some of the test cases included in this category go beyond the requirements in order to examine the SUT's responses to situations where the NPAC SMS exhibits unexpected behavior. For example, some test cases require the NPAC SMS simulator to drop one of the CMIP operations listed in a transaction's scenario. The objective of the invalid test cases is to address the adverse effects that error conditions may have on the SOA and LSMS by subjecting those systems to abnormal processing flows. The focus of these tests will be the semantics of a given transaction and not the syntax or semantics of the CMIP operations which constitute that transaction. Thus, the CMIP requests/responses required for those tests will always be made up of valid, well-formed PDUs. However, key steps in the scenario of performing a valid transaction (i.e. creating subscription versions or audits) will be either missing or executed erroneously and the SUT will be expected to perform properly (as per the requirements) in such situations.

Application to Application test cases for inopportune behavior will have the type A2A and the test number descriptor value of INV. All the test identifiers for this category will start with the prefix **A2A.<System Under Test>.INV**, where the possible values of <System Under Test> are shown in Table 7.1 above.

## 8 Interoperability Testing Exit Criteria

### 8.1 Introduction

The purpose of this section is to describe the criteria for exiting the Interoperability Testing Stage and achieve “certification” for SOAs and LSMSs. The exit criteria defines the minimum set of requirements and test cases that must be adhered to and passed in order for a SOA or an LSMS, the System Under Test (SUT), to enter the Turn-Up testing stage and interoperate with the NPAC SMS.

The basis for developing the exit criteria is the Interoperability Test Plan, and every test case in this document has been analyzed for inclusion in the exit criteria. The methodology used in the analysis is to categorize the failure (i.e., fail a test case) into different severity levels. The following is the list of severity levels:

1. **Required:** The test case is critical and must be addressed in order for the SUT to be certified. This corresponds to functionality whose absence would severely limit, if not prevent all together, the ability of a SUT to provide LNP service and interoperate with the other systems which compose the LNP network
2. **Conditional:** If the SUT is supporting the functionality, this equates to a **Required** test case.
3. **Optional:** The test case failure is minor, however any error should be corrected.

As a general approach, each test case will be identified with a severity level as guidance for exiting the Interoperability Test stage. However, the actual severity level of the failure will not be determined until the time of test execution. This is due to the need to perform the root-cause analysis of the failure; depending on the actual nature of the failure, the severity level may vary within any particular test case.

### 8.2 SUT Certification Guidelines

There are two phases to certification of an SUT: Meeting criteria for Entrance to Turn-Up Testing with the NPAC, and Meeting Criteria for Exit from Interoperability Testing and obtaining certification. The first phase, meeting criteria for Entrance to Turn-Up Testing is provided to allow parallel Turn-Up and Interoperability testing of less critical test cases. To compress testing schedules, an SUT is allowed to enter Turn-Up Testing at partial completion of Interoperability Testing.

An SUT is considered to meet criteria for **Entrance to Turn-Up Testing** if it meets the following criteria:

- Passed all **Required** test cases for the S2S phase.
- Passed all **Required** test cases for the SEC phase.
- Passed all **Required** test cases for the AMG phase.
- Passed all **Required** test cases for the MOC phase for required functionalities and for implemented optional functionalities.
- Passed all normal **Required** test cases for the A2A phase for required functionalities and for implemented optional functionalities.

An SUT is considered to meet criteria for **Exit from Interoperability Testing** and obtain certification to interoperate with the NPAC SMS if it meets the following criteria:

- Passed all **Required** test cases for the S2S phase.
- Passed all **Required** test cases for the SEC phase.

- Passed all **Required** test cases for the AMG phase.
- Passed all **Required** test cases for the MOC phase for required functionalities and for implemented optional functionalities. For test cases related to implemented optional functionalities that fail or cannot be executed, the functionalities cannot be used with the NPAC SMS until the required test cases are passed.

For the A2A phase:

- Executed all test cases related to implemented functionalities. All severity level test cases are executed to provide a base line of known behaviors of supported functionalities.
- Passed all **Required** test cases and all **CONDITIONAL** test cases that the SUT supports the functionality.
- In A2A test cases where one or more of the operations that make up the transaction are optional, the SUT need not run the test case if it does not intend to support the optional functionality. Examples of this type are the Service Provider and Network Data A2A test cases where the operations performed by the LSMS or SOA are optional.

## 9 Stack to Stack Test Cases

### 9.1 Test Cases

#### 9.1.1 S2S.SOA.PING and S2S.LSMS.PING

<b>Purpose</b>	To verify that the IP layer is functioning properly.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide service. No requirements for functionality. May be waived if System Software used does not support/provide a ping utility.
<b>Prerequisites</b>	No association established between the SOA/LSMS and NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA/LSMS issues a ping.</li> <li>2. NPAC SMS Simulator responds to ping.</li> </ol>
<b>Expected Results</b>	Ping is successful.

#### 9.1.2 S2S.SOA.FTP and S2S.LSMS.FTP

<b>Purpose</b>	To verify that the TCP layer is functioning properly.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to accomplish downloads. May also be used instead of “ping” as a diagnostic tool to test TCP/IP part of stack.
<b>Prerequisites</b>	No association established between the SOA/LSMS and NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA/LSMS issues an FTP open with user and password</li> <li>2. NPAC SMS Simulator accepts FTP login request.</li> </ol>
<b>Expected Results</b>	FTP login is successful.

#### 9.1.3 S2S.SOA.VAL.ASSOC and S2S.LSMS.VAL.ASSOC

<b>Purpose</b>	To verify that the SOA/LSMS can issue an ACSE association request and establish an association with the NPAC SMS Simulator.
<b>Severity</b>	R
<b>Severity Explanation</b>	Severe impact on ability to provide service.
<b>Prerequisites</b>	<p>No association established between the SOA/LSMS and NPAC SMS Simulator.</p> <p>System clocks synchronized to within 5 minutes.</p>
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA/LSMS issues association request (AARQ).</li> <li>2. NPAC SMS Simulator accepts association indication.</li> <li>3. NPAC SMS Simulator issues an association response (AARE).</li> <li>4. SOA/LSMS accepts association confirmation.</li> </ol>

<b>Expected Results</b>	Association is correctly established between SOA/LSMS and NPAC SMS Simulator.
-------------------------	---

9.1.4 S2S.SOA.VAL.RELES and S2S.LSMS.VAL.RELES

<b>Purpose</b>	To verify that the SOA/LSMS can issue an ACSE association release request.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service. A-ABORT may be used to provide this functionality (See S2S.SOA.Val.Abort).
<b>Prerequisites</b>	An association has been established between the SOA/LSMS and the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA/LSMS issues an association release request (RLRQ).</li> <li>2. NPAC SMS Simulator accepts the association release indication.</li> <li>3. NPAC SMS Simulator issues an association release response (RLRE).</li> <li>4. SOA/LSMS accepts the association release confirmation.</li> </ol>
<b>Expected Results</b>	Association between SOA/LSMS and NPAC SMS Simulator is correctly released.

9.1.5 S2S.SOA.VAL.RELES.BYNPAC and S2S.LSMS.VAL.RELES.BYNPAC

<b>Purpose</b>	To verify that the NPAC SMS Simulator can terminate an ACSE association established by the SOA/LSMS
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide service. No requirements.
<b>Prerequisites</b>	An association has been established between the SOA/LSMS and the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues an association release request (RLRQ).</li> <li>2. SOA/LSMS accepts the association release indication.</li> <li>3. SOA/LSMS issues an association release response (RLRE).</li> <li>4. NPAC SMS Simulator accepts the association release confirmation.</li> </ol>
<b>Expected Results</b>	Association between SOA/LSMS and NPAC SMS Simulator is correctly released.

9.1.6 S2S.SOA.VAL.ABORT and S2S.LSMS.VAL.ABORT

<b>Purpose</b>	To verify that the SOA/LSMS can issue an ACSE abort request.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service. May be used instead of Unbind (S2S.SOA.VAL.RELES).
<b>Prerequisites</b>	An association has been established between the SOA/LSMS and

	the NPAC SMS Simulator.
<b>Procedure</b>	1. SOA/LSMS issues abort request (ABRT). 2. NPAC SMS accepts abort indication.
<b>Expected Results</b>	Association between SOA/LSMS and NPAC SMS Simulator is no longer established.

9.1.7 S2S.SOA.VAL.ABORT.BYNPAC and S2S.LSMS.VAL.ABORT.BYNPAC

<b>Purpose</b>	To verify that the NPAC SMS Simulator can terminate an ACSE association established by the SOA/LSMS with an ACSE abort request.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service.
<b>Prerequisites</b>	An association has been established between the SOA/LSMS and the NPAC SMS Simulator.
<b>Procedure</b>	1. NPAC SMS Simulator issues abort request (ABRT). 2. SOA/LSMS accepts abort indication.
<b>Expected Results</b>	Association between SOA/LSMS and NPAC SMS Simulator is no longer established.

## 10 Security Test Cases

### 10.1 Group A Security Test Cases

#### 10.1.1 SEC.SOA.VAL.ASSOC.NOSIG and SEC.LSMS.VAL.ASSOC.NOSIG

<b>Purpose</b>	To verify that the SOA/LSMS can issue an ACSE association request with the access control field populated with the proper values for all fields except for signature and establish an association with the NPAC SMS Simulator.
<b>Severity</b>	O
<b>Severity Explanation</b>	No impact on ability to provide service, available for convenience only to allow a phased approach to the implementation of the security protocol. Function will never be used outside this test.
<b>Prerequisites</b>	No association established between the SOA/LSMS and NPAC SMS Simulator.  System clocks synchronized to within 5 minutes.  Access Control attribute set according to Chapter 5 of NPAC SMS Interoperable Interface Specification. The Signature field is not populated.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA/LSMS issues the association request (AARQ).</li> <li>2. NPAC SMS Simulator accepts association indication.</li> <li>3. NPAC SMS Simulator issues an association response (AARE).</li> <li>4. SOA/LSMS accepts association confirmation.</li> </ol>
<b>Expected Results</b>	Association is correctly established between SOA/LSMS and NPAC SMS Simulator.

#### 10.1.2 SEC.SOA.INV.ASSOC.INVSYS and SEC.LSMS.INV.ASSOC.INVSYS

<b>Purpose</b>	Verify SOA/LSMS aborts the association when the NPAC SMS Simulator replies with an invalid System ID.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide a secure interface.
<b>Prerequisites</b>	No association established between the SOA/LSMS and NPAC SMS Simulator.  System clocks synchronized to within 5 minutes.  Access Control attribute set according to Chapter 5 of NPAC SMS Interoperable Interface Specification. The Signature field is not populated.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA/LSMS issues association request.</li> <li>2. NPAC SMS Simulator accepts association indication.</li> <li>3. NPAC SMS Simulator issues an association response with invalid systemId.</li> <li>4. SOA/LSMS aborts association.</li> </ol>
<b>Expected Results</b>	SOA/LSMS aborts association with no reason given.



10.1.3 SEC.SOA.INV.ASSOC.INVT and SEC.LSMS.INV.ASSOC.INVT

<b>Purpose</b>	Verify SOA/LSMS aborts the association when the NPAC SMS Simulator replies with delayed CMIP Departure time.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide a secure interface.
<b>Prerequisites</b>	No association established between the SOA/LSMS and NPAC SMS Simulator.  System clocks synchronized to within 5 minutes.  Access Control attribute set according to Chapter 5 of NPAC SMS Interoperable Interface Specification. The Signature field is not populated.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA/LSMS issues association request</li> <li>2. NPAC SMS Simulator issues an association response with delayed CMIP Departure time.</li> <li>3. SOA/LSMS aborts association</li> </ol>
<b>Expected Results</b>	SOA/LSMS aborts association with no reason given.

10.1.4 SEC.SOA.INV.ASSOC.SEQ and SEC.LSMS.INV.ASSOC.SEQ

<b>Purpose</b>	To verify that the SOA/LSMS aborts the association when the NPAC SMS Simulator replies with an out-of-order sequence number.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide a secure interface.
<b>Prerequisites</b>	No association established between the SOA/LSMS and NPAC SMS Simulator.  System clocks synchronized to within 5 minutes.  Access Control attribute set according to Chapter 5 of NPAC SMS Interoperable Interface Specification. The Signature field is not populated.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA/LSMS issues association request.</li> <li>2. NPAC SMS Simulator accepts association indication.</li> <li>3. NPAC SMS Simulator issues an association response with non-zero sequence number.</li> <li>4. SOA/LSMS aborts association.</li> </ol>
<b>Expected Results</b>	SOA/LSMS aborts association with no reason given.

**10.2 Group B Test Cases**

10.2.1 SEC.SOA.VAL.ASSOC and SEC.LSMS.VAL.ASSOC

<b>Purpose</b>	To verify that the SOA/LSMS can issue an ACSE association
----------------	---

	request with the access control field populated with the proper values for all fields and establish an association with the NPAC SMS Simulator.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service. Requirements exists.
<b>Prerequisites</b>	No association established between the SOA/LSMS and NPAC SMS Simulator.  System clocks synchronized to within 5 minutes.  Access Control attribute set according to Chapter 5 of NPAC SMS Interoperable Interface Specification.  Encrypted Signature is included.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA/LSMS issues association request (AARQ).</li> <li>2. NPAC SMS Simulator accepts association indication.</li> <li>3. NPAC SMS Simulator issues an association response (AARE).</li> <li>4. SOA/LSMS accepts association confirmation.</li> </ol>
<b>Expected Results</b>	Association is correctly established between SOA/LSMS and NPAC SMS Simulator.

10.2.2 SEC.SOA.INV.ASSOC.INVK and SEC.LSMS.INV.ASSOC.INVK

<b>Purpose</b>	Verify SOA/LSMS aborts the association when the NPAC SMS Simulator replies with an invalid Security Key.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide a secure interface.
<b>Prerequisites</b>	No association established between the SOA/LSMS and NPAC SMS Simulator.  System clocks synchronized to within 5 minutes.  Access Control attribute set according to Chapter 5 of NPAC SMS Interoperable Interface Specification.  Encrypted Signature is included.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA/LSMS issues association request</li> <li>2. NPAC SMS Simulator accepts association indication.</li> <li>3. NPAC SMS Simulator issues an association response with invalid keyId or listId.</li> <li>4. SOA/LSMS aborts association.</li> </ol>
<b>Expected Results</b>	SOA/LSMS aborts association with no reason given.

10.2.3 SEC.SOA.INV.ASSOC.INVSIG and SEC.LSMS.INV.ASSOC.INVSIG

<b>Purpose</b>	To verify that the SOA/LSMS rejects an ACSE association when the response of the NPAC SMS Simulator contains an access control field with an invalid signature.
<b>Severity</b>	R

<b>Severity Explanation</b>	Direct impact on ability to provide a secure interface.
<b>Prerequisites</b>	No association established between the SOA/LSMS and NPAC SMS Simulator.  System clocks synchronized to within 5 minutes.  Access Control attribute set according to Chapter 5 of NPAC SMS Interoperable Interface Specification  Encrypted Signature is included.
<b>Procedure</b>	1. SOA/LSMS issues association request (AARQ). 2. NPAC SMS Simulator accepts association indication. 3. NPAC SMS Simulator issues an association response with invalid signature. 4. SOA/LSMS aborts association.
<b>Expected Results</b>	SOA/LSMS aborts association with no reason given.

10.2.4 SEC.SOA.INV.NOT.INVSIG and SEC.LSMS.INV.NOT.INVSIG

<b>Purpose</b>	To verify that the SOA/LSMS aborts an association when it receives a notification from the NPAC SMS Simulator which contains an access control field with an invalid signature.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide a secure interface. Verifies security violation handling at CMIP M-EVENT-REPORT level.
<b>Prerequisites</b>	An association is established between the SOA/LSMS and NPAC SMS Simulator.  System clocks synchronized to within 5 minutes.
<b>Procedure</b>	1. NPAC SMS Simulator sends the InpNPAC-SMS-Operational-Information notification with an invalid signature field. 2. SOA/LSMS detects the invalid signature and aborts association.
<b>Expected Results</b>	SOA/LSMS aborts association with no reason given.

10.2.5 SEC.SOA.INV.CREATE.INVSEQ and SEC.LSMS.INV.CREATE.INVSEQ

<b>Purpose</b>	To verify that the SOA/LSMS aborts an association when it receives a create request from the NPAC SMS Simulator which contains an access control field with an invalid sequence number.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide a secure interface. Verifies security violation handling at CMIP M-CREATE level.
<b>Prerequisites</b>	An association is established between the SOA/LSMS and NPAC SMS Simulator.  System clocks synchronized to within 5 minutes
<b>Procedure</b>	1. NPAC SMS Simulator sends a create request with an invalid sequence number.

	2. SOA/LSMS detects the invalid sequence number and aborts the association.
<b>Expected Results</b>	SOA/LSMS aborts association with no reason given.

10.2.6 SEC.SOA.INV.SET.INVSIG and SEC.LSMS.INV.SET.INVSIG

<b>Purpose</b>	To verify that the SOA/LSMS aborts an association when it receives a set request from the NPAC SMS, which contains an access control field with an invalid signature.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide a secure interface. Verifies security violation handling at CMIP M-SET level.
<b>Prerequisites</b>	An association is established between the SOA/LSMS and NPAC SMS Simulator  System clocks synchronized to within 5 minutes.
<b>Procedure</b>	1. NPAC SMS Simulator sends a set request with an invalid signature. 2. SOA/LSMS detects the invalid signature and aborts the association.
<b>Expected Results</b>	LSMS aborts association with no reason given.

10.2.7 SEC.SOA.INV.ACTION.INVSYS and SEC.LSMS.INV.ACTION.INVSYS

<b>Purpose</b>	To verify that the SOA/LSMS aborts an association when it receives an action request from the NPAC SMS Simulator, which contains an access control field with an invalid system ID.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide a secure interface. Verifies security violation handling at CMIP M-ACTION level.
<b>Prerequisites</b>	An association is established between the SOA/LSMS and NPAC SMS Simulator.  System clocks synchronized to within 5 minutes.
<b>Procedure</b>	1. NPAC SMS Simulator sends an action request with an invalid system ID. 2. SOA/LSMS detects the invalid system ID and aborts the association.
<b>Expected Results</b>	SOA/LSMS aborts association with no reason given.

10.2.8 SEC.SOA.INV.GET.INVT and SEC.LSMS.INV.GET.INVT

<b>Purpose</b>	To verify that the SOA/LSMS aborts an association when it receives a get request from the NPAC SMS, which contains an access control field with an invalid CMIP Departure Time.
<b>Severity</b>	R

<b>Severity Explanation</b>	Direct impact on ability to provide a secure interface. Verifies security violation handling at CMIP M-GET level.
<b>Prerequisites</b>	An association is established between the SOA/LSMS and NPAC SMS Simulator.  System clocks synchronized to within 5 minutes.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a get request with an invalid CMIP Departure Time.</li> <li>2. SOA/LSMS detects the invalid CMIP Departure Time and aborts the association.</li> </ol>
<b>Expected Results</b>	SOA/LSMS aborts association with no reason given.

10.2.9 SEC.SOA.INV.DELETE.INVSIG and SEC.LSMS.INV.DELETE.INVSIG

<b>Purpose</b>	To verify that the SOA/LSMS aborts an association when it receives a delete request from the NPAC SMS, which contains an access control field with an invalid signature.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide a secure interface. Verifies security violation handling at CMIP M-DELETE level.
<b>Prerequisites</b>	An association is established between the SOA/LSMS and NPAC SMS Simulator.  System clocks synchronized to within 5 minutes.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a delete request with an invalid signature.</li> <li>2. SOA/LSMS detects the invalid signature and aborts the association.</li> </ol>
<b>Expected Results</b>	SOA/LSMS aborts association with no reason given.

10.2.10 SEC.SOA.INV.ASSOC.ASSOCSP.INVSYS

<b>Purpose</b>	Verify SOA aborts the association when the NPAC SMS Simulator replies with an invalid System ID, the system id of the associated service provider.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the SOA is supporting associated service provider functionality.
<b>Prerequisites</b>	<p>No association established between the SOA and the NPAC SMS Simulator.</p> <p>System clocks synchronized to within 5 minutes.</p> <p>Access Control attribute set according to Chapter 5 of NPAC SMS Interoperable Interface Specification. The Signature field is not populated.</p> <p>SystemId Specified is an invalid SystemId or if supported by the SOA is a SystemId of an associated service provider defined in the NPAC SMS Simulator.</p>

<b><i>Procedure</i></b>	SOA issues association request  The NPAC SMS Simulator accepts association indication.  The NPAC SMS Simulator issues an association response with invalid systemId.  1. SOA aborts association.
<b><i>Expected Results</i></b>	SOA aborts association with no reason given.

## 11 SOA to NPAC MOC Test Cases

### 11.1 InpNPAC-SMS

<b>MO</b>	InpNPAC-SMS
<b>Purpose</b>	This section contains the test cases for the InpNPAC-SMS Managed Object Class pertaining to the SOA to NPAC SMS Interface, as part of the Managed Object Conformance testing of the NPAC SMS Interoperability Test.
<b>Prerequisite</b>	A SOA Management association function is established. A InpNPAC-SMS Managed Object Instance has been inherently created.

#### 11.1.1 MOC.SOA.CAP.OP.GET.InpNPAC-SMS

<b>Purpose</b>	To test the SOA's ability to GET all the attributes of the InpNPAC-SMS managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	SOA does not need to issue this request to provide LNP service. NPAC SMS name is the only attribute in this Managed Object and its value is available in the specifications.
<b>Prerequisites</b>	A InpNPAC-SMS managed object instance has been inherently created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-GET request to retrieve all the attributes for the InpNPAC-SMS object.</li> <li>NPAC SMS Simulator responds with the M-GET result and all the attributes.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-GET request and retrieves the attributes successfully from the NPAC SMS Simulator.

#### 11.1.2 MOC.SOA.CAP.NOT.InpNPAC-SMS-Operational-Information

<b>Purpose</b>	To test the SOA's ability to receive the InpNPAC-SMS-Operational-Information notification.
<b>Severity</b>	R
<b>Severity Explanation</b>	SOA is required to handle this notification which informs it of NPAC down time.
<b>Prerequisites</b>	A InpNPAC-SMS managed object instance has been inherently created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>NPAC SMS Simulator issues the InpNPAC-SMS-Operational-Information notification.</li> <li>SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA responds with a valid M-EVENT-REPORT confirmation.

#### 11.1.3 MOC.SOA.INV.NOT.InpNPAC-SMS-Operational-Information

<b>Purpose</b>	To test the SOA's ability to handle an invalid M-EVENT-REPORT for the InpNPAC-SMS-Operational-Information notification. This
----------------	--

	will be accomplished by setting the stop time attribute of that notification to a value that is before the start time.
<b>Severity</b>	O
<b>Severity Explanation</b>	No impact to providing LNP service. Can be used to verify error handling by the SOA.
<b>Prerequisites</b>	A InpNPAC-SMS managed object instance has been inherently created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the InpNPAC-SMS-Operational-Information notification with stop time less than the start time.</li> <li>2. SOA returns invalidArgumentValueEr error.</li> </ol>
<b>Expected Results</b>	The SOA will correctly handle the invalid M-EVENT-REPORT received from the NPAC SMS Simulator and return the invalidArgumentValueEr error.

#### 11.1.4 MOC.SOA.CAP.NOT.subscriptionVersionNewNPA-NXX

<b>Purpose</b>	To test the SOA's ability to receive the M-EVENT-REPORT for subscriptionVersionNewNPA-NXX notification.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if the SOA is supporting numberPoolBlocks.
<b>Prerequisites</b>	A InpNPAC-SMS managed object instance has been inherently created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionNewNPA-NXX notification.</li> <li>2. SOA responds with M-EVENT-REPORT confirmation.</li> </ol>
<b>Expected Results</b>	The SOA responds with a valid M-EVENT-REPORT confirmation.

#### 11.1.5 MOC.SOA.INV.GET.InpNPAC-SMS

<b>Purpose</b>	To test the SOA's ability to handle the M-GET error response getListError error to a previously initiated and valid M-GET request for all the attributes of the InpNPAC-SMS object.
<b>Severity</b>	C
<b>Severity Explanation</b>	Allows SOA to verify error handling.
<b>Prerequisites</b>	MOC.SOA.CAP.OP.GET.InpNPAC-SMS
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues a valid M-GET request to retrieve all the attributes from the InpNPAC-SMS managed object instance.</li> <li>2. NPAC SMS Simulator responds with the getListError error.</li> </ol>
<b>Expected Results</b>	The SOA correctly handles the error response getListError from the NPAC SMS Simulator.

#### 11.1.6 MOC.SOA.INV.NOT.subscriptionVersionNewNPA-NXX

<b>Purpose</b>	To test the SOA's ability to handle an invalid M-EVENT-REPORT for the subscriptionVersionNewNPA-NXX notification. This will be accomplished by sending the subscriptionVersionNewNPA-NXX notification with an invalid NPA-NXX value.
<b>Severity</b>	O
<b>Severity Explanation</b>	Required if the SOA is supporting numberPoolBlocks.
<b>Prerequisites</b>	A InpNPAC-SMS managed object instance has been inherently



	created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionNewNPA-NXX notification.</li> <li>2. SOA responds with an invalidArgumentValue error.</li> </ol>
<b>Expected Results</b>	The SOA will correctly handle the invalid M-EVENT-REPORT received from the NPAC SMS Simulator and return the appropriate error.

11.1.7 MOC.SOA.CAP.ACT.InpNotificationRecovery

<b>Purpose</b>	Verify SOA can successfully process the InpNotificationRecovery action. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the service provider SOA supports notification recovery. The SOA will recover either the “individual” notifications or the “range/list” version of the notifications.
<b>Prerequisites</b>	Notifications exist of each type of notification that can be recovered for the requesting service provider. If the “range/list” version of the notifications is being recovered, there must be notifications for each type that use both the “list-data” and “range-data” of the RangeNotify-TN-ID-Info or RangeNotify-ID-Info ASN.1 CHOICE field. There are a total of 13 notifications to be recovered if “individual” subscription version notifications are being recovered and 22 total notifications if “range/list” subscription version notifications are being recovered. See section 5.5.1-2 for an entire list.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA sends the InpNotificationRecovery action to the NPAC SMS Simulator to start notification data download for a specified period of time.</li> <li>2. NPAC SMS Simulator responds with an M-ACTION InpNotificationRecovery response.</li> </ol>
<b>Expected Results</b>	SOA sends the M-ACTION and receives action response with the notification data.

11.1.8 MOC.SOA.INV.ACT.InpNotificationRecovery

<b>Purpose</b>	Verify SOA can successfully process an error response to the InpNotificationRecovery action.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the service provider SOA supports notification recovery.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA sends the InpNotificationRecovery action to the NPAC SMS Simulator to start notification data download for a specified period of time.</li> <li>2. NPAC SMS Simulator responds with error status ‘failed’.</li> </ol>
<b>Expected Results</b>	SOA sends the M-ACTION request and receives the action response with the error successfully.

11.1.9 MOC.SOA.CAP.OP.ACT.InpRecoveryComplete

<b>Purpose</b>	Verify the SOA can indicate that the recovery is complete.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the service provider SOA supports subscription, network data or notification recovery.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA sends the InpRecoveryComplete action to the NPAC SMS Simulator.</li> <li>2. NPAC SMS Simulator responds with M-ACTION InpRecoveryComplete response.</li> </ol>
<b>Expected Results</b>	SOA sends the M-ACTION and receives the action response successfully.

11.1.10 MOC.SOA.INV.ACT.InpRecoveryComplete

<b>Purpose</b>	To test the SOA's ability to handle an error response for an M-ACTION request. This will be accomplished by returning the noSuchAction error in response to the InpRecoveryComplete action.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if the SOA supports network or notification data recovery.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.InpRecoveryComplete
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA sends the valid InpRecoveryComplete M-ACTION request to the InpNPAC-SMS object.</li> <li>2. NPAC SMS Simulator responds with a noSuchAction error.</li> </ol>
<b>Expected Results</b>	The SOA will correctly handle the error response received from the NPAC SMS Simulator.

11.2 InpServiceProvs

<b>MO</b>	InpServiceProvs
<b>Purpose</b>	This section contains the test cases for the InpServiceProvs Managed Object Class pertaining to the SOA to NPAC SMS Interface, as part of the Managed Object Conformance testing of the NPAC SMS Interoperability Test.
<b>Prerequisite</b>	A Service Provider and Network Data Management is established. A InpNPAC-SMS and a InpServiceProvs Managed Object Instances have been inherently created.

11.2.1 MOC.SOA.CAP.OP.GET.InpServiceProvs

<b>Purpose</b>	To test the SOA's ability to GET all the attributes of the InpServiceProvs managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	May be performed to verify the InpServiceProvs managed object instance.
<b>Prerequisites</b>	A InpServiceProvs managed object instance has been inherently created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues a valid M-GET request to retrieve all the attributes from the InpServiceProvs managed object instance.</li> <li>2. NPAC SMS Simulator responds with M-GET result and all</li> </ol>

	attributes.
<b>Expected Results</b>	The SOA issues a valid M-GET request and retrieves the attributes successfully from the NPAC SMS Simulator.

### 11.2.2 MOC.SOA.INV.GET.InpServiceProvs

<b>Purpose</b>	To test the SOA's ability to handle the M-GET error response getListError error to a previously initiated and valid M-GET request for all the attributes of the InpServiceProvs object.
<b>Severity</b>	O
<b>Severity Explanation</b>	Allows SOA to verify error handling.
<b>Prerequisites</b>	MOC.SOA.CAP.OP.GET.InpServiceProvs
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-GET request to retrieve the all attributes from the InpServiceProvs managed object instance.</li> <li>NPAC SMS Simulator responds with the getListError error.</li> </ol>
<b>Expected Results</b>	The SOA correctly handles the error response getListError from the NPAC SMS Simulator.

### 11.3 InpAudits

<b>MO</b>	InpAudits
<b>Purpose</b>	This section contains the test cases for the InpAudits Managed Object Class pertaining to the SOA to NPAC SMS Interface, as part of the Managed Object Conformance testing of the NPAC SMS Interoperability Test.
<b>Prerequisite</b>	A SOA Management association function is established. A InpNPAC-SMS and a InpAudits Managed Object Instances have been created inherently.

#### 11.3.1 MOC.SOA.CAP.OP.GET.InpAudits

<b>Purpose</b>	To test the SOA's ability to GET all the attributes of the InpAudits managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	If audits are supported, SOA may perform to verify the InpAudits object.
<b>Prerequisites</b>	A InpAudits managed object instance has been inherently created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-GET request to retrieve all the attributes of the InpAudits object.</li> <li>NPAC SMS Simulator responds with the M-GET result containing all attributes.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-GET request and retrieves the attributes successfully from the NPAC SMS Simulator.

#### 11.3.2 MOC.SOA.INV.GET.InpAudits

<b>Purpose</b>	To test the SOA's ability to handle the M-GET error response getListError error to a previously initiated and valid M-GET request for all the attributes of the InpAudits object.
<b>Severity</b>	O

<b>Severity Explanation</b>	Allows SOA to verify error handling.
<b>Prerequisites</b>	MOC.SOA.CAP.OP.GET.InpAudits
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-GET request to retrieve the all attributes from the InpAudits object.</li> <li>NPAC SMS Simulator responds with the getListError error.</li> </ol>
<b>Expected Results</b>	The SOA correctly handles the error response getListError from the NPAC SMS Simulator.

#### 11.4 InpSubscriptions

<b>MO</b>	InpSubscriptions
<b>Purpose</b>	This section contains the test cases for the InpSubscriptions Managed Object Class pertaining to the SOA to NPAC SMS Interface, as part of the Managed Object Conformance testing of the NPAC SMS Interoperability Test.
<b>Prerequisite</b>	A SOA Management association function is established. A InpNPAC-SMS and Managed Object Instances have been created inherently.

##### 11.4.1 MOC.SOA.CAP.OP.GET.InpSubscriptions

<b>Purpose</b>	To test the SOA's ability to GET all the attributes of the InpSubscriptions managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Can be used for the SOA to verify the InpSubscriptions object.
<b>Prerequisites</b>	A InpSubscriptions managed object instance has been inherently created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-GET request to retrieve all attributes.</li> <li>NPAC SMS Simulator responds with the M-GET result containing all attributes.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-GET request and retrieves the attributes successfully from the NPAC SMS Simulator.

##### 11.4.2 MOC.SOA.CAP.ACT.subscriptionVersionNewSP-Create-Initial

<b>Purpose</b>	To test the SOA's ability to do the initial create of a subscriptionVersionNPAC object as the new service provider. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service.
<b>Prerequisites</b>	A InpSubscriptions managed object instance has been inherently created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues the M-ACTION subscriptionVersionNewSP-Create action.</li> <li>NPAC SMS Simulator responds with a successful M-ACTION reply.</li> <li>NPAC SMS Simulator issues the objectCreation or subscriptionVersionRangeObjectCreation M-EVENT-REPORT.</li> </ol>

	4. SOA confirms the M-EVENT-REPORT.
<b>Expected Results</b>	The SOA sends a valid M-ACTION request, and receives the NPAC SMS Simulator's M-ACTION response and M-EVENT-REPORT properly.

11.4.3 MOC.SOA.CAP.ACT.subscriptionVersionOldSP-Create-Initial

<b>Purpose</b>	To test the SOA's ability to do the initial create of a subscriptionVersionNPAC object as the old service provider. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service.
<b>Prerequisites</b>	A InpSubscriptions managed object instance has been inherently created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues the M-ACTION subscriptionVersionOldSP-Create action.</li> <li>2. NPAC SMS Simulator responds with a successful M-ACTION reply.</li> <li>3. NPAC SMS Simulator issues the objectCreation or subscriptionVersionRangeObjectCreation M-EVENT-REPORT.</li> <li>4. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA sends a valid M-ACTION request and receives the NPAC SMS Simulator's M-ACTION response and M-EVENT-REPORT properly.

11.4.4 MOC.SOA.CAP.ACT.subscriptionVersionNewSP-Create-Second

<b>Purpose</b>	To test the SOA's ability to do the second create of a subscriptionVersionNPAC object as the new service provider. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.subscriptionVersionNewSP-Create-Initial
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues the M-ACTION subscriptionVersionNewSP-Create action.</li> <li>2. NPAC SMS Simulator responds with a successful M-ACTION reply.</li> <li>3. NPAC SMS Simulator issues the attributeValueChange or subscriptionVersionAttributeValueChange M-EVENT-REPORT.</li> <li>4. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA sends a valid M-ACTION request and receives the NPAC SMS Simulator's M-ACTION response and M-EVENT-REPORT properly.

## 11.4.5 MOC.SOA.CAP.ACT.subscriptionVersionOldSP-Create-Second

<b>Purpose</b>	To test the SOA's ability to do the second create of a subscriptionVersionNPAC object when as the old service provider. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.subscriptionVersionOldSP-Create-Initial
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues the M-ACTION subscriptionVersionOldSP-Create action.</li> <li>2. NPAC SMS Simulator responds with a successful M-ACTION reply.</li> <li>3. NPAC SMS Simulator issues the attributeValueChange or subscriptionVersionAttributeValueChange M-EVENT-REPORT.</li> <li>4. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA sends a valid M-ACTION request and receives the NPAC SMS Simulator's M-ACTION response and M-EVENT-REPORT properly.

## 11.4.6 MOC.SOA.CAP.ACT.subscriptionVersionActivate-VersionId

<b>Purpose</b>	To test the SOA's ability to activate a subscription version using the subscriptionVersionId. This will be accomplished by the SOA issuing the confirmed M-ACTION request for subscriptionVersionActivate and subsequently handling the NPAC SMS Simulator's responses to that action, i.e. the M-ACTION response and the M-EVENT-REPORT for subscriptionVersionStatusAttributeValueChange.
<b>Severity</b>	C
<b>Severity Explanation</b>	Direct impact on providing LNP service. SOA must activate by TN or subscriptionVersionId.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues a valid subscriptionVersionActivate and specifies the subscriptionVersionId.</li> <li>2. NPAC SMS Simulator responds with a successful M-ACTION reply.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-ACTION request and receives the NPAC SMS Simulator's M-ACTION response.

## 11.4.7 MOC.SOA.CAP.ACT.subscriptionVersionActivate-TN

<b>Purpose</b>	To test the SOA's ability to activate a subscription version using a single subscriptionVersionTN.
<b>Severity</b>	C
<b>Severity Explanation</b>	Direct impact on providing LNP service. SOA must activate by TN or subscriptionVersionId.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues a valid subscriptionVersionActivate and specifies the subscriptionVersionTN.</li> </ol>

	2. NPAC SMS Simulator responds with a successful M-ACTION reply.
<b>Expected Results</b>	The SOA issues a valid M-ACTION request and receives the NPAC SMS Simulator's M-ACTION response.

#### 11.4.8 MOC.SOA.CAP.ACT.subscriptionVersionActivate-TNRange

<b>Purpose</b>	To test the SOA's ability to activate a subscription version using a range of subscriptionVersionTNs.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if the SOA is supporting the activation of a range of subscription versions using the subscriptionVersionActivate action.. Direct impact on providing LNP service. Requirement exists however functionality can be achieved by issuing single TN activate requests.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues a valid subscriptionVersionActivate and specifies a range of subscriptionVersionTNs.</li> <li>2. NPAC SMS Simulator responds with a successful M-ACTION reply.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-ACTION request and receives the NPAC SMS Simulator's M-ACTION response.

#### 11.4.9 MOC.SOA.CAP.ACT.subscriptionVersionModify

<b>Purpose</b>	To test the SOA's ability to modify an active subscription version.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service. Requirement exists which can be satisfied using M-SET only (test case in subscriptionVersionNPAC).
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues a valid subscriptionVersionModify M-ACTION and specifies either the subscriptionVersionId or subscriptionVersionTN.</li> <li>2. NPAC SMS Simulator responds with a successful M-ACTION response.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-ACTION request and receives the NPAC SMS Simulator's M-ACTION response and M-EVENT-REPORT properly.

#### 11.4.10 MOC.SOA.CAP.ACT.subscriptionVersionCancel

<b>Purpose</b>	To test the SOA's ability to cancel a subscription version. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.subscriptionVersionOldSP-Create-Initial or

	MOC.SOA.CAP.ACT.subscriptionVersionOldSP-Create-Second
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues a valid subscriptionVersionCancel M-ACTION request.</li> <li>2. NPAC SMS Simulator responds with a successful M-ACTION reply.</li> <li>3. NPAC SMS issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange specifying the subscriptionVersionStatus as 'cancel-pending'.</li> <li>4. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-ACTION request and receives the NPAC SMS Simulator's M-ACTION response and M-EVENT-REPORT properly.

11.4.11 MOC.SOA.CAP.ACT.subscriptionVersionOldSP-CancellationAcknowledge

<b>Purpose</b>	To test the SOA's ability to acknowledge the cancellation of a cancel-pending subscription version after the new service provider has requested the action to cancel. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.subscriptionVersionCancel
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange with the subscriptionStatus set to "cancel-pending".</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> <li>3. SOA issues a valid subscriptionVersionOldSP-CancellationAcknowledge M-ACTION request.</li> <li>4. NPAC SMS Simulator responds successfully to the request.</li> <li>5. NPAC SMS Simulator issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange with the subscriptionStatus set to "canceled".</li> <li>6. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-ACTION request and receives the NPAC SMS Simulator's M-ACTION response and M-EVENT-REPORT properly.

11.4.12 MOC.SOA.CAP.ACT.subscriptionVersionNewSP-CancellationAcknowledge

<b>Purpose</b>	To test the SOA's ability, acting as the new service provider, to acknowledge the cancellation of a cancel-pending subscription version after the old service provider has requested the action to cancel. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	R



<b>Severity Explanation</b>	Direct impact on providing LNP service.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.subscriptionVersionCancel
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange with the subscriptionStatus set to “cancel-pending”.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> <li>3. SOA issues a valid subscriptionVersionNewSP-CancellationAcknowledge M-ACTION request.</li> <li>4. NPAC SMS Simulator responds successfully to the M-ACTION.</li> <li>5. NPAC SMS Simulator issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange with the subscriptionStatus set to “canceled”.</li> <li>6. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA receives the first M-EVENT-REPORT (cancel-pending), issues a valid M-ACTION request and receives the NPAC SMS Simulator's M-ACTION response and second M-EVENT-REPORT (canceled) properly.

11.4.13 MOC.SOA.CAP.ACT.subscriptionVersionDisconnect

<b>Purpose</b>	To test the SOA’s ability to disconnect an active subscription version immediately. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.subscriptionVersionActivate-VersionId or MOC.SOA.CAP.ACT.subscriptionVersionActivate-TN
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA sends a valid subscriptionVersionDisconnect M-ACTION request, populating the subscriptionEffectiveReleaseDate.</li> <li>2. NPAC SMS Simulator responds with a successful M-ACTION reply.</li> <li>3. NPAC SMS Simulator sends the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange with the subscriptionVersionStatus set to “disconnect-pending”.</li> <li>4. SOA confirms the M-EVENT-REPORT.</li> <li>5. NPAC SMS Simulator issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange with the subscriptionVersionStatus set to “old”.</li> <li>6. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-ACTION request and receives the NPAC SMS Simulator's M-ACTION response and M-EVENT-REPORTs properly.

11.4.14 MOC.SOA.CAP.ACT.subscriptionVersionRemoveFromConflict

<b>Purpose</b>	To test the SOA’s ability to remove a subscription version from conflict. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
----------------	--

<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service.
<b>Prerequisites</b>	A subscriptionVersion with a status of 'conflict'.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues a valid subscriptionVersionRemoveFromConflict M-ACTION request.</li> <li>2. NPAC SMS Simulator responds with a successful M-ACTION reply.</li> <li>3. NPAC SMS issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange specifying the subscriptionVersionStatus as 'pending'.</li> <li>4. SOA confirms the M-EVENT-REPORT.</li> <li>5. NPAC SMS issues an attributeValueChange or subscriptionVersionRangeAttributeValueChange for the subscriptionVersion with the subscriptionOldSP-Authorization set to TRUE.</li> <li>6. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-ACTION request and receives the NPAC SMS Simulator's M-ACTION response and M-EVENT-REPORTs properly.

11.4.15 MOC.SOA.INV.GET.InpSubscriptions

<b>Purpose</b>	To test the SOA's ability to handle the M-GET error response operationCancelled error to a previously initiated and valid M-GET request for all attributes of the InpSubscriptions object.
<b>Severity</b>	C
<b>Severity Explanation</b>	May be performed to validate SOA's error handling.
<b>Prerequisites</b>	MOC.SOA.CAP.OP.GET.InpSubscriptions
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues a valid M-GET request to retrieve all the of the InpSubscriptions object.</li> <li>2. NPAC SMS Simulator responds with the operationCancelled error.</li> </ol>
<b>Expected Results</b>	The SOA correctly handles the error response operationCancelled error from the NPAC SMS Simulator.

11.4.16 MOC.SOA.INV.ACT.subscriptionVersionNewSP-Create

<b>Purpose</b>	To test the SOA's ability to handle an error response for an M-ACTION request. This will be accomplished by returning the soa-not-authorized error in response to the subscriptionVersionNewSP-Create action.
<b>Severity</b>	R
<b>Severity Explanation</b>	Should be performed to validate the SOA's error handling.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.InpSubscriptionVersionNewSP-Create-Initial
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues a valid subscriptionVersionNewSP-Create action.</li> <li>2. NPAC SMS Simulator responds with a soa-not-authorized error.</li> </ol>
<b>Expected Results</b>	The SOA will correctly handle the error response received from the NPAC SMS Simulator.

11.4.17 MOC.SOA.INV.ACT.subscriptionVersionOldSP-Create

<b>Purpose</b>	To test the SOA's ability to handle an error response for an M-ACTION request. This will be accomplished by returning the error status 'version-create-already-exists' in response to the subscriptionVersionOldSP-Create action.
<b>Severity</b>	R
<b>Severity Explanation</b>	Should be performed to validate the SOA's error handling.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.InpSubscriptionVersionOldSP-Create-Initial
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues a valid subscriptionVersionOldSP-Create action.</li> <li>2. NPAC SMS Simulator responds with a 'version-create-already-exists' error.</li> </ol>
<b>Expected Results</b>	The SOA will correctly handle the error response received from the NPAC SMS Simulator.

11.4.18 MOC.SOA.INV.ACT.subscriptionVersionActivate

<b>Purpose</b>	To test the SOA's ability to handle an error response for an M-ACTION request. This will be accomplished by returning the soa-not-authorized error in response to the subscriptionVersionActivate action.
<b>Severity</b>	R
<b>Severity Explanation</b>	Should be performed to validate the SOA's error handling.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.InpSubscriptionVersionActivate-VersionId
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues a valid subscriptionVersionActivate action.</li> <li>2. NPAC SMS Simulator responds with a soa-not-authorized error.</li> </ol>
<b>Expected Results</b>	The SOA will correctly handle the error response received from the NPAC SMS Simulator.

11.4.19 MOC.SOA.INV.ACT.subscriptionVersionModify

<b>Purpose</b>	To test the SOA's ability to handle an error response for an M-ACTION request. This will be accomplished by returning the invalidArgumentValue error in response to the subscriptionVersionModify action.
<b>Severity</b>	R
<b>Severity Explanation</b>	Should be performed to validate the SOA's error handling.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.InpSubscriptionVersionModify
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues a valid subscriptionVersionModify action.</li> <li>2. NPAC SMS Simulator responds with an invalidArgumentError error.</li> </ol>
<b>Expected Results</b>	The SOA will correctly handle the error response received from the NPAC SMS Simulator.

11.4.20 MOC.SOA.INV.ACT.subscriptionVersionCancel

<b>Purpose</b>	To test the SOA's ability to handle an error response for an M-ACTION request. This will be accomplished by returning the error status 'failed' in response to the subscriptionVersionCancel action.
<b>Severity</b>	R
<b>Severity Explanation</b>	Should be performed to validate the SOA's error handling.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.InpSubscriptionVersionCancel
<b>Procedure</b>	1. SOA issues a valid subscriptionVersionCancel action. 2. NPAC SMS Simulator responds with a 'failed' error.
<b>Expected Results</b>	The SOA will correctly handle the error response received from the NPAC SMS Simulator.

11.4.21 MOC.SOA.INV.ACT.subscriptionVersionOldSP-CancellationAcknowledge

<b>Purpose</b>	To test the SOA's ability to handle an error response for an M-ACTION request. This will be accomplished by returning the error status "no-version-found" in response to the subscriptionVersionOldSP-CancellationAcknowledge action.
<b>Severity</b>	R
<b>Severity Explanation</b>	Should be performed to validate the SOA's error handling.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.InpSubscriptionVersionOldSP-CancellationAcknowledge
<b>Procedure</b>	1. SOA issues a valid subscriptionVersionOldSP-CancellationAcknowledge action. 2. NPAC SMS Simulator responds with a 'no-version-found' error.
<b>Expected Results</b>	The SOA will correctly handle the error response received from the NPAC SMS Simulator.

11.4.22 MOC.SOA.INV.ACT.subscriptionVersionNewSP-CancellationAcknowledge

<b>Purpose</b>	To test the SOA's ability to handle an error response for an M-ACTION request. This will be accomplished by returning the noSuchAction error in response to the subscriptionVersionNewSP-CancellationAcknowledge action.
<b>Severity</b>	R
<b>Severity Explanation</b>	Should be performed to validate the SOA's error handling.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.InpSubscriptionVersionNewSP-CancellationAcknowledge
<b>Procedure</b>	1. SOA issues a valid subscriptionVersionNewSP-CancellationAcknowledge action. 2. NPAC SMS Simulator responds with noSuchAction error.
<b>Expected Results</b>	The SOA will correctly handle the error response received from the NPAC SMS Simulator.

11.4.23 MOC.SOA.INV.ACT.subscriptionVersionDisconnect

<b>Purpose</b>	To test the SOA's ability to handle an error response for an M-ACTION request. This will be accomplished by returning the noSuchArgument error in response to the subscriptionVersionDisconnect action.
<b>Severity</b>	R
<b>Severity Explanation</b>	Should be performed to validate the SOA's error handling.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.InpSubscriptionVersionDisconnect
<b>Procedure</b>	1. SOA issues a valid subscriptionVersionDisconnect action. 2. NPAC SMS Simulator responds with noSuchArgument error.
<b>Expected Results</b>	The SOA will correctly handle the error response received from the NPAC SMS Simulator.

11.4.24 MOC.SOA.INV.ACT.subscriptionVersionRemoveFromConflict

<b>Purpose</b>	To test the SOA's ability to handle an error response for an M-ACTION request. This will be accomplished by returning the error status, "soa-not-authorized" in response to the subscriptionVersionRemoveFromConflict action.
<b>Severity</b>	R
<b>Severity Explanation</b>	Should be performed to validate the SOA's error handling.
<b>Prerequisites</b>	
<b>Procedure</b>	1. SOA issues a valid subscriptionVersionDisconnect action. 2. NPAC SMS Simulator responds with 'no-not-authorized' error.
<b>Expected Results</b>	The SOA will correctly handle the error response received from the NPAC SMS Simulator.

11.4.25 MOC.SOA.CAP.ACT.numberPoolBlockCreateAction

<b>Purpose</b>	Verify the SOA's ability to issue the numberPoolBlockCreate action.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if SOA will be supporting numberPoolBlock data.
<b>Prerequisites</b>	N/A
<b>Procedure</b>	1. SOA issues a valid numberPoolBlockCreate M-ACTION request. 2. NPAC SMS Simulator responds with a successful M-ACTION response.
<b>Expected Results</b>	SOA issues a valid M-ACTION request and retrieves the data successfully from the NPAC SMS Simulator.

11.4.26 MOC.SOA.INV.ACT.numberPoolBlockCreateAction

<b>Purpose</b>	Verify the SOA's ability to handle an error response to the numberPoolBlockCreate action. This will be accomplished by issuing a soa-not-authorized error in response to the numberPoolBlockCreate action.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if SOA will be supporting numberPoolBlock data.

<b>Prerequisites</b>	N/A
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues a valid numberPoolBlockCreate M-ACTION request.</li> <li>2. NPAC SMS Simulator responds with a soa-not-authorized error.</li> </ol>
<b>Expected Results</b>	SOA correctly handles the error response received from the NPAC SMS Simulator.

11.4.27 MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeStatusAttributeValueChange

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeStatusAttributeValueChange M-EVENT-REPORT using the range-data CHOICE field in the ASN.1.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if SOA will be supporting the subscriptionVersionRangeStatusAttributeValueChange M-EVENT-REPORT.
<b>Prerequisites</b>	1 or more subscription versions exist on the NPAC SMS Simulator with a subscriptionVersionStatus of 'pending'.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionRangeStatusAttributeValueChange specifying the subscriptionVersionStatus as 'active' for 1 or more subscription versions with consecutive TNs and subscription version Ids.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and acknowledges it correctly.

11.4.28 MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeAttributeValueChange

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeAttributeValueChange M-EVENT-REPORT using the range-data CHOICE field in the ASN.1.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if SOA will be supporting the subscriptionVersionRangeAttributeValueChange M-EVENT-REPORT.
<b>Prerequisites</b>	The service provider SOA has issued a subscriptionVersionNewSP-Create action and created 1 or more 'pending' subscription versions on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionRangeAttributeValueChange M-EVENT-REPORT, simulating the Old Service Provider create, for 1 or more 'pending' subscription versions with consecutive TNs and subscription version Ids.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and acknowledges it correctly.

11.4.29 MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeObjectCreation

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeObjectCreation M-EVENT-REPORT using the range-data CHOICE field in the ASN.1.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if SOA will be supporting the subscriptionVersionRangeObjectCreation notification.
<b>Prerequisites</b>	A InpSubscriptions managed object instance has been inherently created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionRangeObjectCreation M-EVENT-REPORT for 1 or more subscription versions with consecutive TNs and subscription version Ids.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and acknowledges it correctly.

11.4.30 MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeDonorSP-CustomerDisconnectDate

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeDonorSP-CustomerDisconnectDate M-EVENT-REPORT using the range-data CHOICE field in the ASN.1.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if SOA will be supporting the subscriptionVersionRangeDonorSP-CustomerDisconnectDate notification.
<b>Prerequisites</b>	1 or more subscription versions exist on the NPAC SMS Simulator with a subscriptionVersionStatus of 'active'.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionRangeDonorSP-CustomerDisconnectDate for 1 or more subscription versions with consecutive TNs and subscription version Ids.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and acknowledges it correctly.

11.4.31 MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeCancellationAcknowledgeRequest

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeCancellationAcknowledgeRequest M-EVENT-REPORT using the range-data CHOICE field in the ASN.1.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if SOA will be supporting the subscriptionVersionRangeCancellationAcknowledgeRequest notification.
<b>Prerequisites</b>	1 or more subscription versions exist on the NPAC SMS with a subscriptionVersionStatus of 'cancel-pending'.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the</li> </ol>

	<p>subscriptionVersionRangeCancellationAcknowledgeRequest for 1 or more subscription versions with consecutive TNs and subscription version Ids.</p> <p>2. SOA confirms the M-EVENT-REPORT.</p>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and acknowledges it correctly.

11.4.32 MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeNewSP-CreateRequest

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeNewSP-CreateRequest M-EVENT-REPORT using the range-data CHOICE field in the ASN.1.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if SOA will be supporting the subscriptionVersionRangeNewSP-CreateRequest notification.
<b>Prerequisites</b>	1 or more subscription versions exist on the NPAC SMS with a subscriptionVersionStatus of 'pending' that were created by the Old Service Provider.
<b>Procedure</b>	<p>1. NPAC SMS Simulator issues the subscriptionVersionRangeNewSP-CreateRequest notification for 1 or more subscription versions with consecutive TNs and subscription version Ids to the New Service Provider.</p> <p>2. SOA confirms the M-EVENT-REPORT.</p>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and acknowledges it correctly.

11.4.33 MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeOldSP-ConcurrenceRequest

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeOldSP-ConcurrenceRequest M-EVENT-REPORT using the range-data CHOICE field in the ASN.1.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if SOA will be supporting the subscriptionVersionRangeOldSP-ConcurrenceRequest notification.
<b>Prerequisites</b>	1 or more subscription versions exist on the NPAC SMS with a subscriptionVersionStatus of 'pending' that were created by the New Service Provider.
<b>Procedure</b>	<p>1. NPAC SMS Simulator issues the subscriptionVersionRangeOldSP-ConcurrenceRequest for 1 or more subscription versions with consecutive TNs and subscription version Ids to the Old Service Provider.</p> <p>2. SOA confirms the M-EVENT-REPORT.</p>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and acknowledges it correctly.

11.4.34 MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeOldSPFinalConcurrenceWindowExpiration

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeOldSPFinalConcurrenceWindowExpiration M-EVENT-REPORT using the range-data CHOICE field in the ASN.1.
----------------	--



<b>Severity</b>	C
<b>Severity Explanation</b>	Required if SOA will be supporting the subscriptionVersionRangeOldSPFinalConcurrenceWindowExpiration notification.
<b>Prerequisites</b>	1 or more subscription versions exist on the NPAC SMS with a subscriptionVersionStatus of 'pending' that were created by the New Service Provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionRangeOldSPFinalConcurrenceWindowExpiration for 1 or more subscription versions with consecutive TNs and subscription version Ids.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and acknowledges it correctly.

11.4.35 MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeNewSPFinalCreateWindowExpiration

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeNewSPFinalCreateWindowExpiration M-EVENT-REPORT using the range-data CHOICE field in the ASN.1.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if SOA will be supporting the subscriptionVersionRangeNewSPFinalCreateWindowExpiration notification.
<b>Prerequisites</b>	1 or more subscription versions exist on the NPAC SMS with a subscriptionVersionStatus of 'pending' that were created by the Old Service Provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionRangeNewSPFinalCreateWindowExpiration for 1 or more subscription versions with consecutive TNs and subscription version Ids to the New Service Provider.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and acknowledges it correctly.

11.4.36 MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeStatusAttribute ValueChange

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeStatusAttribute ValueChange M-EVENT-REPORT using the list-data CHOICE field in the ASN.1.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if SOA will be supporting the subscriptionVersionRangeStatusAttribute ValueChange M-EVENT-REPORT.
<b>Prerequisites</b>	2 or more subscription versions exist on the NPAC SMS Simulator with a subscriptionVersionStatus of 'pending' with consecutive TNs and non-consecutive subscription version Ids.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS issues the subscriptionVersionRangeStatusAttribute ValueChange specifying the subscriptionVersionStatus as 'active' for 2 or more subscription versions with consecutive TNs and non-consecutive subscription version Ids.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>

<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and acknowledges it correctly.
-------------------------	---

11.4.37 MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeAttributeValueChange

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeStatusAttributeValueChange M-EVENT-REPORT using the list-data CHOICE field in the ASN.1.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if SOA will be supporting the subscriptionVersionRangeAttributeValueChange M-EVENT-REPORT.
<b>Prerequisites</b>	The service provider SOA has issued a subscriptionVersionNewSP-Create action and created 2 or more 'pending' subscription versions on the NPAC SMS Simulator with consecutive TNs and non-consecutive subscription version Ids.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionRangeRangeAttributeValueChange M-EVENT-REPORT, simulating the Old Service Provider.</li> <li>2. SOA confirms the M-EVENT-REPORT for 2 or more subscription versions with consecutive TNs and non-consecutive subscription version Ids.</li> </ol>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and acknowledges it correctly.

11.4.38 MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeObjectCreation

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeObjectCreation M-EVENT-REPORT using the list-data CHOICE field in the ASN.1.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if SOA will be supporting the subscriptionVersionRangeObjectCreation notification.
<b>Prerequisites</b>	An InpSubscriptions managed object instance has been inherently created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionRangeObjectCreation M-EVENT-REPORT for 2 or more subscription versions with consecutive TNs and non-consecutive subscription version Ids.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and acknowledges it correctly.

11.4.39 MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeDonorSP-CustomerDisconnectDate

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeDonorSP-CustomerDisconnectDate M-EVENT-REPORT using the list-data CHOICE field in the ASN.1.
<b>Severity</b>	C

<b>Severity Explanation</b>	Required if SOA will be supporting the subscriptionVersionRangeDonorSP-CustomerDisconnectDate notification.
<b>Prerequisites</b>	2 or more subscription versions exist on the NPAC SMS Simulator with a subscriptionVersionStatus of 'active' with consecutive TNs and non-consecutive subscription version Ids.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionRangeDonorSP-CustomerDisconnectDate for 2 or more subscription versions with consecutive TNs and non-consecutive subscription version Ids.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and acknowledges it correctly.

11.4.40 MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeCancellationAcknowledgeRequest

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeCancellationAcknowledgeRequest M-EVENT-REPORT using the list-data CHOICE field in the ASN.1.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if SOA will be supporting the subscriptionVersionRangeCancellationAcknowledgeRequest notification.
<b>Prerequisites</b>	2 or more subscription versions exist on the NPAC SMS with a subscriptionVersionStatus of 'pending' with consecutive TNs and non-consecutive subscription version Ids.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionRangeCancellationAcknowledgeRequest for 2 or more subscription versions with consecutive TNs and non-consecutive subscription version Ids.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and acknowledges it correctly.

11.4.41 MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeNewSP-CreateRequest

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeNewSP-CreateRequest M-EVENT-REPORT using the list-data CHOICE field in the ASN.1.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if SOA will be supporting the subscriptionVersionRangeNewSP-CreateRequest notification.
<b>Prerequisites</b>	2 or more subscription versions exist on the NPAC SMS with a subscriptionVersionStatus of 'pending' that were created by the Old Service Provider with consecutive TNs and non-consecutive subscription version Ids.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionRangeNewSP-CreateRequest for 2 or more subscription versions with consecutive TNs and non-consecutive subscription version Ids</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT

	and acknowledges it correctly.
--	--------------------------------

11.4.42 MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeOldSP-ConcurrenceRequest

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeOldSP-ConcurrenceRequest M-EVENT-REPORT using the list-data CHOICE field in the ASN.1.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if SOA will be supporting the subscriptionVersionRangeOldSP-ConcurrenceRequest notification.
<b>Prerequisites</b>	2 or more subscription versions exist on the NPAC SMS with a subscriptionVersionStatus of 'pending' that were created by the New Service Provider with consecutive TNs and non-consecutive subscription version Ids.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionRangeOldSP-ConcurrenceRequest for 2 or more subscription versions with consecutive TNs and non-consecutive subscription version Ids to the Old Service Provider.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and acknowledges it correctly.

11.4.43 MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeOldSPFinalConcurrenceWindowExpiration

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeOldSPFinalConcurrenceWindowExpiration M-EVENT-REPORT using the list-data CHOICE field in the ASN.1.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if SOA will be supporting the subscriptionVersionRangeOldSPFinalConcurrenceWindowExpiration notification.
<b>Prerequisites</b>	2 or more subscription versions exist on the NPAC SMS with a subscriptionVersionStatus of 'pending' that were created by the New Service Provider with consecutive TNs and non-consecutive subscription version Ids.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionRangeOldSPFinalConcurrenceWindowExpiration for 2 or more subscription versions with consecutive TNs and non-consecutive subscription version Ids to the Old Service Provider.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and acknowledges it correctly.

11.4.44 MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeNewSPFinalCreateWindowExpiration

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeNewSPFinalCreateWindowExpiration M-EVENT-REPORT using the list-data CHOICE field in the ASN.1.
<b>Severity</b>	C

<b>Severity Explanation</b>	Required if SOA will be supporting the subscriptionVersionRangeNewSPFinalCreateWindowExpiration notification.
<b>Prerequisites</b>	2 or more subscription versions exist on the NPAC SMS with a subscriptionVersionStatus of 'pending' that were created by the Old Service Provider with consecutive TNs and non-consecutive subscription version Ids.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionRangeNewSPFinalCreateWindowExpiration for 2 or more subscription versions with consecutive TNs and non-consecutive subscription version Ids to the New Service Provider.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and acknowledges it correctly.

11.4.45 MOC.SOA.INV.NOT.RANGE.subscriptionVersionRangeStatusAttributeValueChange

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeStatusAttributeValueChange M-EVENT-REPORT with invalid syntax for the failed-service-provs field.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	The service provider SOA has issued a subscriptionVersionNewSP-Create action and created 2 or more 'pending' subscription versions on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS issues the subscriptionVersionRangeStatusAttributeValueChange specifying the subscriptionVersionStatus as 'active'.</li> <li>2. SOA rejects the M-EVENT-REPORT with an invalidArgument error.</li> </ol>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and returns the invalidArgument or other appropriate error.

11.4.46 MOC.SOA.INV.NOT.RANGE.subscriptionVersionRangeAttributeValueChange

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeStatusAttributeValueChange M-EVENT-REPORT with invalid syntax for the version-id field.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	The service provider SOA has issued a subscriptionVersionNewSP-Create action and created 2 or more 'pending' subscription versions on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionRangeRangeAttributeValueChange M-EVENT-REPORT, simulating the Old Service Provider.</li> <li>2. SOA rejects the M-EVENT-REPORT with an invalidArgument error.</li> </ol>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and returns the invalidArgument or other appropriate error.

11.4.47 MOC.SOA.INV.NOT.[RANGE](#).subscriptionVersionRangeObjectCreation

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeObjectCreation M-EVENT-REPORT with invalid syntax for the object-info field.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	A InpSubscriptions managed object instance has been inherently created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionRangeObjectCreation M-EVENT-REPORT.</li> <li>2. SOA rejects the M-EVENT-REPORT with an invalidArgument error.</li> </ol>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and returns the invalidArgument or other appropriate error.

11.4.48 MOC.SOA.INV.NOT.[RANGE](#).subscriptionVersionRangeDonorSP-CustomerDisconnectDate

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeDonorSP-CustomerDisconnectDate M-EVENT-REPORT with invalid syntax for the customer-disconnect-date field.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	2 or more subscription versions exist on the NPAC SMS Simulator with a subscriptionVersionStatus of 'active'.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionRangeDonorSP-CustomerDisconnectDate .</li> <li>2. SOA rejects the M-EVENT-REPORT with an invalidArgument error.</li> </ol>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and returns the invalidArgument or other appropriate error.

11.4.49 MOC.SOA.INV.NOT.[RANGE](#).subscriptionVersionRangeCancellationAcknowledgeRequest

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeCancellationAcknowledgeRequest M-EVENT-REPORT with invalid syntax for the tn-version-id field.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	2 or more subscription versions exist on the NPAC SMS with a subscriptionVersionStatus of 'pending'.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionRangeCancellationAcknowledgeRequest.</li> <li>2. SOA rejects the M-EVENT-REPORT with an invalidArgument error.</li> </ol>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT

	and acknowledges it correctly.
--	--------------------------------

11.4.50 MOC.SOA.INV.NOT.RANGE.subscriptionVersionRangeNewSP-CreateRequest

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeNewSP-CreateRequest M-EVENT-REPORT with invalid syntax for the service-prov-id field.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	2 or more subscription versions exist on the NPAC SMS with a subscriptionVersionStatus of 'pending' that were created by the Old Service Provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionRangeNewSP-CreateRequest to the New Service Provider.</li> <li>2. SOA rejects the M-EVENT-REPORT with an invalidArgument error.</li> </ol>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and returns the invalidArgument or other appropriate error.

11.4.51 MOC.SOA.INV.NOT.RANGE.subscriptionVersionRangeOldSP-ConcurrenceRequest

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeOldSP-ConcurrenceRequest M-EVENT-REPORT with invalid syntax for the tn-version-id field.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	2 or more subscription versions exist on the NPAC SMS with a subscriptionVersionStatus of 'pending' that were created by the New Service Provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionRangeOldSP-ConcurrenceRequest to the Old Service Provider.</li> <li>2. SOA rejects the M-EVENT-REPORT with an invalidArgument error.</li> </ol>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and returns the invalidArgument or other appropriate error.

11.4.52 MOC.SOA.INV.NOT.RANGE.subscriptionVersionRangeOldSPFinalConcurrenceWindowExpiration

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeOldSPFinalConcurrenceWindowExpiration M-EVENT-REPORT with invalid syntax for the tn-version-id field.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	2 or more subscription versions exist on the NPAC SMS with a subscriptionVersionStatus of 'pending' that were created by the New Service Provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionRangeOldSPFinalConcurrenceWindowExpiration to the Old Service Provider.</li> </ol>

	2. SOA rejects the M-EVENT-REPORT with an invalidArgument error.
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and returns the invalidArgument or other appropriate error.

11.4.53 MOC.SOA.INV.NOT.[RANGE](#).subscriptionVersionRangeNewSPFinalCreateWindowExpiration

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeNewSPFinalCreateWindowExpiration M-EVENT-REPORT with invalid syntax for the service-prov-old-authorization field.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	2 or more subscription versions exist on the NPAC SMS with a subscriptionVersionStatus of 'pending' that were created by the Old Service Provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionRangeNewSPFinalCreateWindowExpiration to the New Service Provider.</li> <li>2. SOA rejects the M-EVENT-REPORT with an invalidArgument error.</li> </ol>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and returns the invalidArgument or other appropriate error.

11.5 **InpNetwork**

<b>MO</b>	InpNetwork
<b>Purpose</b>	This section contains the test cases for the InpNetwork Managed Object Class pertaining to the SOA to NPAC SMS Interface, as part of the Managed Object Conformance testing of the NPAC SMS Interoperability Test.
<b>Prerequisite</b>	A Service Provider and Network Data Management association function is established. A InpNPAC-SMS and a InpNetwork Managed Object Instances have been created inherently.

11.5.1 MOC.SOA.CAP.OP.GET.InpNetwork

<b>Purpose</b>	To test the SOA's ability to GET all the attributes of the InpNetwork managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	No impact to providing LNP service. May be performed to verify the InpNetwork object.
<b>Prerequisites</b>	A InpNetwork managed object instance has been inherently created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA sends a valid M-GET request to retrieve all attributes of the InpNetwork object.</li> <li>2. NPAC SMS Simulator responds with the M-GET result containing all attributes.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-GET request and retrieves the attributes successfully from the NPAC SMS Simulator.



11.5.2 MOC.SOA.INV.GET.InpNetwork

<b>Purpose</b>	To test the SOA's ability to handle the M-GET error response processingFailure error to a previously initiated and valid M-GET request for all attributes of the InpNetwork object.
<b>Severity</b>	C
<b>Severity Explanation</b>	Should be performed if MOC.SOA.CAP.OP.GET.InpNetwork performed.
<b>Prerequisites</b>	MOC.SOA.CAP.OP.GET.InpNetwork
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA sends a valid M-GET request to retrieve all the attributes from the InpNetwork object.</li> <li>NPAC SMS Simulator responds with processingFailure error.</li> </ol>
<b>Expected Results</b>	The SOA correctly handles the error response processingFailure error from the NPAC SMS Simulator.

11.5.3 MOC.SOA.CAP.ACT.InpNetwork.InpDownload

<b>Purpose</b>	To test the SOA 's ability to download the serviceProvNetwork, serviceProvNPA-NXX and serviceProvLRN objects instantiated on the NPAC SMS Simulator. This will be accomplished by the SOA issuing the confirmed M-ACTION request for InpDownload via the InpNetwork object and subsequently handling the NPAC SMS Simulator M-ACTION response.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the SOA is to support network data recovery.
<b>Prerequisites</b>	Network data to be recovered exists. The data to be recovered includes data to be added, modified, or deleted for each type of network data to be recovered.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA sends a InpDownload M-ACTION request with criteria as supported by the product.</li> <li>NPAC SMS Simulator responds with a InpDownload M-ACTION response.</li> </ol>
<b>Expected Results</b>	The SOA sends a valid M-ACTION request and receives the NPAC SMS Simulator M-ACTION response properly.

11.5.4 MOC.SOA.INV.ACT.InpNetwork.InpDownload

<b>Purpose</b>	To test the SOA 's ability to handle an error response for the InpDownload action.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the SOA supports network data recovery.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA sends a InpDownload M-ACTION request for network data with criteria as supported by the product.</li> <li>NPAC SMS Simulator responds with error status 'failed'.</li> </ol>
<b>Expected Results</b>	The SOA will correctly handle the error response received from the NPAC SMS Simulator.

11.5.5 MOC.SOA.VAL.InpDownload-NPA-NXX-X

<b>Purpose</b>	Verify the SOA's ability to issue the InpDownload action for
----------------	--

	serviceProvNPA-NXX-X data.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if SOA will be supporting serviceProvNPA-NXX-X data.
<b>Prerequisites</b>	serviceProvNPA-NXX-X objects exist on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid InpDownload M-ACTION request for all network data or specific serviceProvNPA-NXX-X objects.</li> <li>NPAC SMS Simulator responds with a successful M-ACTION response containing the requested data.</li> </ol>
<b>Expected Results</b>	SOA issues a valid M-ACTION request and retrieves the data successfully from the NPAC SMS Simulator.

### 11.6 serviceProv

<b>MO</b>	serviceProv
<b>Purpose</b>	This section contains the test cases for the serviceProv Managed Object Class pertaining to the SOA to NPAC SMS Interface, as part of the Managed Object Conformance testing of the NPAC SMS Interoperability Test.
<b>Prerequisite</b>	A Service Provider and Network Data Management association function is established. A InpNPAC-SMS and a InpServiceProvs Managed Object Instances have been created inherently. A serviceProv Managed Object Instance has been created locally by the NPAC SMS Simulator personnel.

#### 11.6.1 MOC.SOA.CAP.OP.SET.serviceProv

<b>Purpose</b>	To test the SOA's ability to SET all of the mandatory attributes on which the M-SET operation is allowed in the serviceProv managed object instance.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if the service provider will manage their service provider profile from the SOA.
<b>Prerequisites</b>	A serviceProv managed object instance has been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues the M-SET to update the serviceProvAddress, serviceProvSysLinkInfo and serviceProvName.</li> <li>NPAC SMS Simulator responds with a successful M-SET result.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-SET request and sets the serviceProvAddress, serviceProvName and serviceProvSysLinkInfo attributes successfully in the NPAC SMS Simulator.

#### 11.6.2 MOC.SOA.CAP.OP.GET.serviceProv

<b>Purpose</b>	To test the SOA's ability to GET all the attributes of the serviceProv managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if service provider will be updating the serviceProv object from the SOA.
<b>Prerequisites</b>	MOC.SOA.CAP.OP.SET.serviceProv

<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues the M-GET to retrieve all attributes from the serviceProv object.</li> <li>2. NPAC SMS Simulator responds with a successful M-GET result containing all attributes.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-GET request and retrieves the attributes successfully from the NPAC SMS Simulator.

### 11.6.3 MOC.SOA.VAL.SET.SING.serviceProv

<b>Purpose</b>	To test the SOA's ability to SET a single attribute, namely the serviceProvAddress in the serviceProv managed object instance. Note: Although only a single attribute is updated, all modifiable attributes of the serviceProv object must be sent.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if the service provider will be updating their serviceProv object from the SOA, but may be satisfied by MOC.SOA.CAP.OP.SET.serviceProv
<b>Prerequisites</b>	A serviceProv managed object instance has been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues the M-SET to update the serviceProvAddress.</li> <li>2. NPAC SMS Simulator responds with a successful M-SET result.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-SET request and sets the serviceProvAddress attribute successfully in the NPAC SMS Simulator.

### 11.6.4 MOC.SOA.VAL.SET.SING.COND.serviceProv

<b>Purpose</b>	To test the SOA's ability to SET a single conditional attribute, namely the serviceProvBillingAddress in the serviceProv managed object instance. Note: Although only a single attribute is updated, all modifiable attributes of the serviceProv object must be sent.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if service provider elects to update their serviceProv object from the SOA.
<b>Prerequisites</b>	MOC.SOA.VAL.SET.SING.serviceProv
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues the M-SET to update the serviceProvBillingAddress.</li> <li>2. NPAC SMS Simulator responds with a successful M-SET result.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-SET request and sets the serviceProvBillingAddress attribute successfully in the NPAC SMS Simulator.

### 11.6.5 MOC.SOA.VAL.SET.MULT.serviceProv

<b>Purpose</b>	To test the SOA's ability to SET a group of attributes, namely the serviceProvAddress, npacCustomerAllowableFunctions, and serviceProvSOA-Address in the serviceProv managed object instance.
----------------	---

	Note: Although only a subset of the attributes is being updated, all modifiable attributes of the serviceProv object must be sent.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if service provider elects to update their serviceProv object from the SOA.
<b>Prerequisites</b>	MOC.SOA.VAL.SET.SING.COND.serviceProv
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues the M-SET to update the serviceProvAddress, npacCustomerAllowableFunctions and serviceProvSOA-Address.</li> <li>NPAC SMS Simulator responds with a successful M-SET result.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-SET request and sets the attribute group successfully in the NPAC SMS Simulator.

#### 11.6.6 MOC.SOA.INV.SET.serviceProv

<b>Purpose</b>	To test the SOA's ability to handle the M-SET error response processingFailure error to a previously initiated and valid M-SET request for the serviceProvName attribute.
<b>Severity</b>	C
<b>Severity Explanation</b>	Should be performed if MOC.SOA.VAL.SET.SING.serviceProv, MOC.SOA.VAL.SET.COND.serviceProv or MOC.SOA.VALSET.MULT.serviceProv is performed..
<b>Prerequisites</b>	MOC.SOA.VAL.SET.SING.serviceProv
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-SET for the serviceProvName attribute on the serviceProv object.</li> <li>NPAC SMS Simulator responds with a processingFailure error.</li> </ol>
<b>Expected Results</b>	The SOA correctly handles the error response processingFailure error from the NPAC SMS Simulator.

#### 11.6.7 MOC.SOA.INV.GET.serviceProv

<b>Purpose</b>	To test the SOA's ability to handle the M-GET error response processingFailure error to a previously initiated and valid M-GET request for all the attributes of the serviceProv object.
<b>Severity</b>	C
<b>Severity Explanation</b>	Should be performed if MOC.SOA.CAP.OP.GET.serviceProv is performed.
<b>Prerequisites</b>	MOC.SOA.CAP.OP.GET.serviceProv
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-GET for all the attributes of the serviceProv object.</li> <li>NPAC SMS Simulator responds with a processingFailure error.</li> </ol>
<b>Expected Results</b>	The SOA correctly handles the error response processingFailure error from the NPAC SMS Simulator.

#### 11.6.8 MOC.SOA.BND.MIN.SET.serviceProv

<b>Purpose</b>	To test the behaviour of the SOA when setting the city field of the serviceProvAddress attribute to a value of length 1 octet which is
----------------	--

	the lower bound of the range for the city size.
<b>Severity</b>	C
<b>Severity Explanation</b>	Should be performed if MOC.SOAVAL.SET.SING.serviceProv is performed.
<b>Prerequisites</b>	MOC.SOA.VAL.SET.SING.serviceProv
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-SET for the serviceProvAddress attribute with the city equal to a string value of length 1 on the serviceProv object.</li> <li>NPAC SMS Simulator responds with a valid M-SET result.</li> </ol>
<b>Expected Results</b>	SOA handles the M-SET response and the city field is set accordingly in the NPAC SMS Simulator.

11.6.9 MOC.SOA.BND.MAX.SET.serviceProv

<b>Purpose</b>	To test the behaviour of the SOA when setting the city field of the serviceProvAddress attribute to a value of length 20 which is the higher bound of the range for the city size.
<b>Severity</b>	C
<b>Severity Explanation</b>	Should be performed if MOC.SOAVAL.SET.SING.serviceProv is performed.
<b>Prerequisites</b>	MOC.SOA.VAL.SET.SING.serviceProv
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-SET for the serviceProvAddress attribute with the city equal to a string value of length 20 on the serviceProv object.</li> <li>NPAC SMS Simulator responds with a valid M-SET result.</li> </ol>
<b>Expected Results</b>	The SOA handles the M-SET response and the city field is set accordingly in the NPAC SMS Simulator.

11.7 *subscriptionAudit*

<b>MO</b>	subscriptionAudit
<b>Purpose</b>	This section contains the test cases for the subscriptionAudit Managed Object Class pertaining to the SOA to NPAC SMS Interface, as part of the Managed Object Conformance testing of the NPAC SMS Interoperability Test.
<b>Prerequisite</b>	A SOA Management association function is established. A lnpNPAC-SMS and a lnpAudits Managed Object Instances have been created inherently.

11.7.1 MOC.SOA.CAP.OP.CRE.subscriptionAudit

<b>Purpose</b>	To test the SOA's ability to CREATE a managed object instance of the subscriptionAudit class and subsequently handle the objectCreation notification generated by that operation.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if the SOA will be supporting audits.
<b>Prerequisites</b>	A lnpAudits managed object instance has been inherently created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA sends the M-CREATE request for the subscriptionAudit object.</li> <li>NPAC SMS Simulator responds with a successful M-CREATE</li> </ol>

	<p>response.</p> <ol style="list-style-type: none"> <li>3. NPAC SMS issues the objectCreation M-EVENT-REPORT for the subscriptionAudit object.</li> <li>4. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-CREATE request causing the audit object to be successfully created in the NPAC SMS Simulator, receives the M-EVENT-REPORT for objectCreation from the NPAC SMS Simulator and responds with the M-EVENT-REPORT confirmation to the NPAC SMS Simulator.

11.7.2 MOC.SOA.CAP.OP.GET.subscriptionAudit

<b>Purpose</b>	To test the SOA's ability to GET all the attributes of the subscriptionAudit managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. SOA may perform to verify functionality works.
<b>Prerequisites</b>	MOC.SOA.CAP.OP.CRE.subscriptionAudit
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA sends M-GET request for all attributes of the subscriptionAudit object.</li> <li>2. NPAC SMS Simulator responds with the M-GET result containing all attributes.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-GET request and retrieves the attributes successfully from the NPAC SMS Simulator.

11.7.3 MOC.SOA.CAP.OP.DEL.subscriptionAudit

<b>Purpose</b>	To test the SOA's ability to DELETE an existing managed object instance of the subscriptionAudit class and subsequently handle the objectDeletion notification generated by that operation.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. SOA may perform to verify functionality.
<b>Prerequisites</b>	MOC.SOA.CAP.OP.CRE.subscriptionAudit
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues M-DELETE request for subscriptionAudit object.</li> <li>2. NPAC SMS Simulator responds with successful M-DELETE response.</li> <li>3. NPAC SMS issues objectDeletion M-EVENT-REPORT for the subscriptionAudit object.</li> <li>4. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-DELETE request and removes the managed object instance successfully from the NPAC SMS Simulator, receives the M-EVENT-REPORT for objectDeletion from the NPAC SMS Simulator and responds with the M-EVENT-REPORT confirmation to the NPAC SMS Simulator.

11.7.4 MOC.SOA.CAP.NOT.subscriptionAuditResults

<b>Purpose</b>	To test the SOA's ability to receive the M-EVENT-REPORT for the subscriptionAudit's subscriptionAuditResults notification.
----------------	--

<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if the SOA is supporting audits.
<b>Prerequisites</b>	MOC.SOA.CAP.OP.CRE.subscriptionAudit
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends the subscriptionAuditResults M-EVENT-REPORT.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA responds with a valid M-EVENT-REPORT confirmation.

#### 11.7.5 MOC.SOA.CAP.NOT.subscriptionAudit-DiscrepancyReport

<b>Purpose</b>	To test the SOA's ability to receive the M-EVENT-REPORT for the subscriptionAudit's subscriptionAudit-DiscrepancyReport notification.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if the SOA is supporting audits.
<b>Prerequisites</b>	MOC.SOA.CAP.OP.CRE.subscriptionAudit
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends the subscriptionAudit-DiscrepancyReport M-EVENT-REPORT to the SOA.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA responds with a valid M-EVENT-REPORT confirmation.

#### 11.7.6 MOC.SOA.VAL.CRE.AUTO.subscriptionAudit

<b>Purpose</b>	To test the SOA's ability to CREATE a managed object instance of the subscriptionAudit class using AUTOMATIC INSTANCE NAMING and subsequently handle the objectCreation notification generated by that operation.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	MOC.SOA.CAP.OP.CRE.subscriptionAudit
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA sends the M-CREATE request for the subscriptionAudit object.</li> <li>2. NPAC SMS Simulator responds with a successful M-CREATE response.</li> <li>3. NPAC SMS issues the objectCreation M-EVENT-REPORT for the subscriptionAudit object.</li> <li>4. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-CREATE request with AUTOMATIC INSTANCE NAMING, causing the managed object instance to be successfully created in the NPAC SMS Simulator, receives the M-EVENT-REPORT for objectCreation from the NPAC SMS Simulator and responds with the M-EVENT-REPORT confirmation to the NPAC SMS Simulator.

#### 11.7.7 MOC.SOA.VAL.GET.SCOP.FILT.subscriptionAudit

<b>Purpose</b>	To test the SOA's ability to initiate a valid scoped and filtered M-
----------------	--

	GET request for all attributes of the subscriptionAudit object.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. If the service provider opts to implement Audits, there is no impact due to not supporting this operation. Only impact is on the ability of the SOA to examine audits in-progress as a group instead of one at a time.
<b>Prerequisites</b>	Multiple subscriptionAudit managed object instances have been created and MOC.SOA.VAL.GET.MULT.subscriptionAudit
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA sends the M-GET request for all the attributes and filtered for the given serviceProvId.</li> <li>2. NPAC SMS Simulator responds with the M-GET results containing all the attributes for the matching subscriptionAudit objects.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-GET request and retrieves all the attributes successfully from the NPAC SMS Simulator.

#### 11.7.8 MOC.SOA.VAL.DEL.SCOP.subscriptionAudit

<b>Purpose</b>	To test the SOA's ability to issue a scoped DELETE for a group of existing managed object instances of the subscriptionAudit class and subsequently handle the objectDeletion notifications generated by that operation.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Functionality may be satisfied by deleting the audit instances one at a time.
<b>Prerequisites</b>	MOC.SOA.CAP.OP.CRE.subscriptionAudit, and multiple instances created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA sends the M-DELETE request for the subscriptionAudit filtered for the given serviceProvId.</li> <li>2. NPAC SMS Simulator responds with the M-DELETE results.</li> <li>3. NPAC SMS issues the objectDeletion M-EVENT-REPORTs for all the deleted objects.</li> <li>4. SOA confirms all the M-EVENT-REPORTs.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid scoped M-DELETE request starting at the InpAudits Managed Object and removes the subscriptionAudit managed object instances successfully from the NPAC SMS Simulator, receives the M-EVENT-REPORTs for objectDeletion from the NPAC SMS Simulator and responds with the M-EVENT-REPORT confirmations to the NPAC SMS Simulator.

#### 11.7.9 MOC.SOA.INV.CRE.subscriptionAudit

<b>Purpose</b>	To test the SOA's ability to handle the M-CREATE error response processingFailure error to a previously initiated and valid M-CREATE request for a subscriptionAudit managed object instance.
<b>Severity</b>	C
<b>Severity Explanation</b>	Should be performed if MOC.SOA.CAP.OP.CRE.subscriptionAudit is performed.
<b>Prerequisites</b>	MOC.SOA.CAP.OP.CRE.subscriptionAudit
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA sends a valid M-CREATE request for a subscriptionAudit</li> </ol>



	object. 2. NPAC SMS Simulator responds with a processingFailure error.
<b>Expected Results</b>	The SOA correctly handles the error response processingFailure error from the NPAC SMS Simulator.

#### 11.7.10 MOC.SOA.INV.GET.subscriptionAudit

<b>Purpose</b>	To test the SOA's ability to handle the M-GET error response getListError error to a previously initiated and valid M-GET request for all attributes of the subscriptionAudit object.
<b>Severity</b>	C
<b>Severity Explanation</b>	Should be performed if MOC.SOA.CAP.OP.GET.subscriptionAudit is performed.
<b>Prerequisites</b>	MOC.SOA.CAP.OP.GET.subscriptionAudit
<b>Procedure</b>	1. SOA sends a valid M-GET request for all of a subscriptionAudit object. 2. NPAC SMS Simulator responds with a getListError error.
<b>Expected Results</b>	The SOA correctly handles the error response getListError error from the NPAC SMS Simulator.

#### 11.7.11 MOC.SOA.INV.DEL.subscriptionAudit

<b>Purpose</b>	To test the SOA's ability to handle the M-DELETE error response processingFailure error to a previously initiated and valid M-DELETE request for a subscriptionAudit managed object instance.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. SOA may perform to verify error handling.
<b>Prerequisites</b>	MOC.SOA.CAP.OP.DEL.subscriptionAudit
<b>Procedure</b>	1. SOA sends a valid M-DELETE request for a subscriptionAudit object. 2. NPAC SMS Simulator responds with a processingFailure error.
<b>Expected Results</b>	The SOA correctly handles the error response processingFailure error from the NPAC SMS Simulator.

#### 11.7.12 MOC.SOA.INV.NOT.subscriptionAuditResults

<b>Purpose</b>	To test the SOA's ability to handle an invalid M-EVENT-REPORT for the subscriptionAuditResults notification. This will be accomplished by setting an attribute of that notification to an invalid value.
<b>Severity</b>	O
<b>Severity Explanation</b>	Should be performed if MOC.SOA.CAP.OP.CRE.subscriptionAudit is performed.
<b>Prerequisites</b>	MOC.SOA.CAP.OP.CRE.subscriptionAudit
<b>Procedure</b>	1. NPAC SMS Simulator issues the subscriptionAuditResults M-EVENT-REPORT containing an invalid attribute. 2. SOA responds with an invalidArgumentValue error.
<b>Expected Results</b>	The SOA will correctly handle the invalid M-EVENT-REPORT received from the NPAC SMS Simulator and return the

	invalidArgumentValue error.
--	-----------------------------

11.7.13 MOC.SOA.INV.NOT.subscriptionAudit-DiscrepancyReport

<b>Purpose</b>	To test the SOA's ability to handle an invalid M-EVENT-REPORT for the subscriptionAudit-DiscrepancyReport notification. This will be accomplished by setting the auditDiscrepancyVersionId attribute of that notification to a value with invalid ASN syntax.
<b>Severity</b>	O
<b>Severity Explanation</b>	Should be performed if MOC.SOA.CAP.OP.CRE.subscriptionAudit is performed.
<b>Prerequisites</b>	MOC.SOA.CAP.OP.CRE.subscriptionAudit
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionAudit-DiscrepancyReport M-EVENT-REPORT containing an invalid attribute.</li> <li>2. SOA responds with an invalidArgumentValue error.</li> </ol>
<b>Expected Results</b>	The SOA will correctly handle the invalid M-EVENT-REPORT received from the NPAC SMS Simulator and return the invalidArgumentValueEr error.

11.7.14 MOC.SOA.INV.CAP.OP.CRE.subscriptionAudit

<b>Purpose</b>	To verify a SOA can handle an error when an audit is created with an invalid subscriptionAuditRequestingSP.
<b>Severity</b>	O
<b>Severity Explanation</b>	Test case should be executed if the SOA will be supporting audits.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues the M-CREATE for an audit with the subscriptionAuditRequestingSP set to a value other than a service provider id specified in the access control.</li> <li>2. The NPAC SMS Simulator responds with the M-CREATE error response of invalidAttributeValue.</li> </ol>
<b>Expected Results</b>	The SOA successfully initiates the audit M-CREATE and successfully handles the M-CREATE error response.

**11.8 subscriptionVersionNPAC**

<b>MO</b>	SubscriptionVersionNPAC
<b>Purpose</b>	This section contains the test cases for the subscriptionVersionNPAC Managed Object Class pertaining to the SOA to NPAC SMS Interface, as part of the Managed Object Conformance testing of the NPAC SMS Interoperability Test.
<b>Prerequisite</b>	A SOA Management association function is established. A InpNPAC-SMS and InpSubscriptions Managed Object Instances have been created inherently.

11.8.1 MOC.SOA.CAP.OP.SET.OldSP.subscriptionVersionNPAC

<b>Purpose</b>	To test the SOA's ability to SET all the attributes which may be set by an old service provider for the subscriptionVersionNPAC managed object instance (i.e. subscriptionOldSP-DueDate, and
----------------	--

	subscriptionOldSP-Authorization) using an M-SET.
<b>Severity</b>	O
<b>Severity Explanation</b>	Impacts providing LNP service. Requirement exists but it may be satisfied using the Modify M-ACTION.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.InpSubscriptionVersionOldSP-Create-Initial
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA sends a valid M-SET request for an existing subscriptionVersionNPAC object updating the subscriptionOldSSP-DueDate, subscriptionOldSP-Authorization and subscripitonStatusChangeCauseCode.</li> <li>NPAC SMS Simulator responds with a successful M-SET reply.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-SET request, updates the attribute values successfully in the NPAC SMS Simulator and correctly handles the M-SET response.

### 11.8.2 MOC.SOA.CAP.OP.SET.NewSP.subscriptionVersionNPAC

<b>Purpose</b>	To test the SOA's ability to SET all the attributes which may be set by a new service provider for the subscriptionVersionNPAC managed object instance (i.e. subscriptionLRN, subscriptionNewSP-DueDate, subscriptionCLASS-DPC, subscriptionCLASS-SSN, subscriptionLIDB-DPC, subscriptionLIDB-SSN, subscriptionCNAM-DPC, subscriptionCNAM-SSN, subscriptionISVM-DPC, subscriptionISVM-SSN, subscriptionEndUserLocationValue, subscriptionEndUserLocationType, and subscriptionBillingId).
<b>Severity</b>	O
<b>Severity Explanation</b>	Impacts providing LNP service. Requirement exists but it may be satisfied using the Modify M-ACTION.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.InpSubscriptionVersionNewSP-Create-Initial
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA sends a valid M-SET request for an existing subscriptionVersionNPAC object updating the subscriptionLRN, subscriptionNewSP-DueDate, subscriptionCLASS-DPC, subscriptionCLASS-SSN, subscriptionLIDB-DPC, subscriptionLIDB-SSN, subscriptionCNAM-DPC, subscriptionCNAM-SSN, subscriptionISVM-DPC, subscriptionISVM-SSN, subscriptionEndUserLocationValue, subscriptionEndUserLocationType, and subscriptionBillingId.</li> <li>NPAC SMS Simulator responds with a successful M-SET reply.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-SET request, updates the attribute values successfully in the NPAC SMS Simulator and correctly handles the M-SET response..

### 11.8.3 MOC.SOA.CAP.OP.GET.subscriptionVersionNPAC

<b>Purpose</b>	To test the SOA's ability to GET all the attributes of the subscriptionVersionNPAC managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact providing LNP service. Requirement exists in the

	GDMO. If not implemented SOA may not be able to retrieve any information on existing versions given that Audits are not implemented either.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.InpSubscriptionVersionNewSP-Create-Initial and MOC.SOA.CAP.ACT.InpSubscriptionVersionOldSP-Create-Initial
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA sends valid M-GET request for all attributes of the subscriptionVersionNPAC object.</li> <li>2. NPAC SMS Simulator responds with M-GET result containing all attributes.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-GET request and retrieves the attributes successfully from the NPAC SMS Simulator.

#### 11.8.4 MOC.SOA.CAP.NOT.subscriptionVersionOldSP-ConcurrenceRequest

<b>Purpose</b>	To test the SOA's ability to receive the subscriptionVersionNPAC's subscriptionVersionOldSP-ConcurrenceRequest notification.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if the SOA is supporting the subscriptionVersionOldSP-ConcurrenceRequest notification.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.InpSubscriptionVersionNewSP-Create-Initial
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the M-EVENT-REPORT, subscriptionVersionOldSP-ConcurrenceRequest.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA responds with a valid M-EVENT-REPORT confirmation.

#### 11.8.5 MOC.SOA.CAP.NOT.subscriptionVersionOldSP-FinalConcurrenceWindowExpiration

<b>Purpose</b>	To test the SOA's ability to receive the subscriptionVersionNPAC's subscriptionVersionOldSP-FinalConcurrenceWindowExpiration notification.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if the SOA is supporting the subscriptionVersionOldSP-FinalConcurrenceWindowExpiration notification..
<b>Prerequisites</b>	MOC.SOA.CAP.NOT.subscriptionVersionOldSP-ConcurrenceRequest
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the M-EVENT-REPORT, subscriptionVersionOldSP-FinalConcurrenceWindowExpiration.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA responds with a valid M-EVENT-REPORT confirmation.

#### 11.8.6 MOC.SOA.CAP.NOT.subscriptionVersionNewSP-CreateRequest

<b>Purpose</b>	To test the SOA's ability to receive the subscriptionVersionNPAC's subscriptionVersionNewSP-CreateRequest notification.
<b>Severity</b>	C

<b>Severity Explanation</b>	Required if the SOA is supporting the subscriptionVersionNewSP-CreateRequest notification.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.InpSubscriptionVersionOldSP-Create-Initial
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the M-EVENT-REPORT, subscriptionVersionNewSP-CreateRequest.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA responds with a valid M-EVENT-REPORT confirmation.

11.8.7 MOC.SOA.CAP.NOT.subscriptionVersionCancellationAcknowledgeRequest

<b>Purpose</b>	To test the SOA's ability to receive the subscriptionVersionNPAC's subscriptionVersionCancellationAcknowledgeRequest notification.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if the SOA is supporting the CancellationAcknowledgeRequest notification..
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.subscriptionVersionOldSP-CancellationAcknowledge
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the M-EVENT-REPORT, subscriptionVersionCancellationAcknowledgeRequest.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA responds with a valid M-EVENT-REPORT confirmation.

11.8.8 MOC.SOA.CAP.NOT.subscriptionVersionDonorSP-CustomerDisconnectDate

<b>Purpose</b>	To test the SOA's ability to receive the subscriptionVersionNPAC's subscriptionVersionDonorSP-CustomerDisconnectDate notification.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if the SOA is supporting the DonorSP-CustomerDisconnectDate notification.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.subscriptionVersionDisconnect
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the M-EVENT-REPORT, subscriptionVersionDonorSP-CustomerDisconnectDate.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA responds with a valid M-EVENT-REPORT confirmation.

11.8.9 MOC.SOA.VAL.SET.SING.subscriptionVersionNPAC

<b>Purpose</b>	To test the SOA's ability to initiate a valid M-SET request for a single attribute, namely the subscriptionVersionOldSP attribute.
<b>Severity</b>	O
<b>Severity Explanation</b>	Impacts providing LNP service. Requirement exists but it may be satisfied using the Modify M-ACTION.
<b>Prerequisites</b>	A subscriptionVersionNPAC instance has been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA sends a valid M-SET request for the subscriptionVersionOldSP attribute.</li> <li>2. NPAC SMS Simulator sends a successful M-SET reply.</li> </ol>

<b>Expected Results</b>	The SOA issues a valid M-SET request and updates the attribute successfully in the NPAC SMS Simulator.
-------------------------	--

11.8.10 MOC.SOA.VAL.SET.MULT.subscriptionVersionNPAC

<b>Purpose</b>	To test the SOA's ability to initiate a valid M-SET request for a group of attributes, namely the subscriptionCNAM-DPC, subscriptionCNAM-SSN, subscriptionCLASS-DPC, subscriptionCLASS-SSN and subscriptionNewSP-DueDate, attributes.
<b>Severity</b>	O
<b>Severity Explanation</b>	Impacts providing LNP service. Requirement exists but it may be satisfied using the Modify M-ACTION.
<b>Prerequisites</b>	A subscriptionVersionNPAC instance has been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA sends a valid M-SET request for the subscriptionCNAM-DPC, subscriptionCNAM-SSN, subscriptionCLASS-DPC, subscriptionCLASS-SSN and subscriptionNewSP-DueDate attributes.</li> <li>NPAC SMS Simulator sends a successful M-SET reply.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-SET request, updates the attributes successfully in the NPAC SMS Simulator and correctly handles the response.

11.8.11 MOC.SOA.VAL.GET.SCOP.subscriptionVersionNPAC

<b>Purpose</b>	To test the SOA's ability to initiate a valid scoped M-GET request for all attributes of a subscriptionVersionNPAC object. This will be accomplished by retrieving all the attributes starting at the base managed object InpSubscriptions and ending at the subscriptionVersionNPAC level with filtering on the TN range.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact providing LNP service.
<b>Prerequisites</b>	Multiple subscriptionVersionNPAC managed object instances have been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA sends a valid scope and filtered M-GET request for all the attributes with a filter reflecting a TN-Range.</li> <li>NPAC SMS Simulator responds with the successful M-GET results containing the attribute.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-GET request and retrieves the attributes successfully from the NPAC SMS Simulator.

11.8.12 MOC.SOA.VAL.NOT.subscriptionVersionNewNPA-NXX

<b>Purpose</b>	To test the SOA's ability to receive the subscriptionVersionNewNPA-NXX notification from the NPAC SMS Simulator.
<b>Severity</b>	R
<b>Severity Explanation</b>	Needed to inform the SOA of opening a new NPA-NXX for porting.

<b>Prerequisites</b>	subscriptionVersionNPAC managed object instance exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the M-EVENT-REPORT, subscriptionVersionNewNPA-NXX.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA confirms the M-EVENT-REPORT request.

11.8.13 MOC.SOA.VAL.NOT.subscriptionVersionStatusAttributeValueChange

<b>Purpose</b>	To test the SOA's ability to handle the M-EVENT-REPORT for the subscriptionVersionNPAC's subscriptionVersionStatusAttributeValueChange notification.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if the SOA is supporting the subscriptionVersionStatusAttributeValueChange notification.
<b>Prerequisites</b>	subscriptionVersionNPAC managed object instance exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues subscriptionVersionStatusAttributeValueChange M-EVENT-REPORT with the subscriptionVersionStatus set to "download-failed" and the subscriptionVersionFailedSP-List.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA confirms the M-EVENT-REPORT.

11.8.14 MOC.SOA.INV.SET.SING.subscriptionVersionNPAC

<b>Purpose</b>	To test the SOA's ability to handle the M-SET error response setListError error to a previously initiated and valid M-SET request for a single attribute, namely the subscriptionLRN attribute.
<b>Severity</b>	C
<b>Severity Explanation</b>	Should be performed if prerequisite is performed.
<b>Prerequisites</b>	MOC.SOA.VAL.SET.SING.subscriptionVersionNPAC
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA sends a valid M-SET request for the subscriptionLRN attribute.</li> <li>2. NPAC SMS Simulator responds with a setListError error response.</li> </ol>
<b>Expected Results</b>	The SOA correctly handles the error response setListError error from the NPAC SMS Simulator.

11.8.15 MOC.SOA.INV.GET.subscriptionVersionNPAC

<b>Purpose</b>	To test the SOA's ability to handle the M-GET error response noSuchObjectInstance error to a previously initiated and valid M-GET request for all attributes of the subscriptionVersionNPAC object.
<b>Severity</b>	C
<b>Severity Explanation</b>	Should be performed if prerequisite is performed.
<b>Prerequisites</b>	MOC.SOA.CAP.OP.GET.subscriptionVersionNPAC
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA sends a valid M-GET request for all the attributes of the subscriptionVersionNPAC object.</li> <li>2. NPAC SMS Simulator responds with a noSuchObjectInstance</li> </ol>

	error response.
<b>Expected Results</b>	The SOA correctly handles the error response noSuchObjectInstance error from the NPAC SMS Simulator.

11.8.16 MOC.SOA.INV.NOT.subscriptionVersionOldSp-ConcurrenceRequest

<b>Purpose</b>	To test the SOA's ability to handle the M-EVENT-REPORT for the subscriptionVersionNPAC's subscriptionVersionOldSP-ConcurrenceRequest notification with an invalid syntax for the subscriptionNewSP-DueDate attribute.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.SOA.CAP.NOT.subscriptionVersionOldSP-ConcurrenceRequest
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionOldSP-ConcurrenceRequest M-EVENT-REPORT with invalid syntax for the subscriptionNewSP-DueDate.</li> <li>2. SOA rejects the M-EVENT-REPORT with invalidArgumentValue error.</li> </ol>
<b>Expected Results</b>	The SOA rejects the M-EVENT-REPORT with invalid syntax.

11.8.17 MOC.SOA.INV.NOT.subscriptionVersionNewSP-CreateRequest

<b>Purpose</b>	To test the SOA's ability to handle the M-EVENT-REPORT for the subscriptionVersionNPAC's subscriptionVersionNewSP-CreateRequest notification with an invalid syntax for the subscriptionTN attribute.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.SOA.CAP.NOT.subscriptionVersionNewSP-CreateRequest
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionNewSP-CreateRequest M-EVENT-REPORT with invalid syntax for the subscriptionTN.</li> <li>2. SOA rejects the M-EVENT-REPORT with invalidArgumentValue error.</li> </ol>
<b>Expected Results</b>	The SOA rejects the M-EVENT-REPORT with invalid syntax.

11.8.18 MOC.SOA.INV.NOT.subscriptionVersionCancellationAcknowledgeRequest

<b>Purpose</b>	To test the SOA's ability to handle the M-EVENT-REPORT for the subscriptionVersionNPAC's subscriptionVersionCancellationAcknowledgeRequest notification with an invalid syntax for the subscriptionVersionId attribute.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.SOA.CAP.NOT.subscriptionVersionCancellationAcknowledgeRequest



	eRequest
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues subscriptionVersionCancellationAcknowledgeRequest M-EVENT-REPORT with invalid syntax for the subscriptionVersionId.</li> <li>2. SOA rejects the M-EVENT-REPORT with invalidArgumentValue error.</li> </ol>
<b>Expected Results</b>	The SOA rejects the M-EVENT-REPORT with invalid syntax.

11.8.19 MOC.SOA.INV.NOT.subscriptionVersionDonorSP-  
CustomerDisconnectDate

<b>Purpose</b>	To test the SOA's ability to handle the M-EVENT-REPORT for the subscriptionVersionNPAC's subscriptionVersionDonorSP-CustomerDisconnectDate notification with an invalid syntax for the subscriptionEffectiveReleaseDate attribute.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.SOA.CAP.NOT.subscriptionVersionDonorSP-CustomerDisconnectDate
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues subscriptionVersionDonorSP-CustomerDisconnectDate M-EVENT-REPORT with invalid syntax for the subscriptionEffectiveReleaseDate attribute.</li> <li>2. SOA rejects the M-EVENT-REPORT with invalidArgumentValue error.</li> </ol>
<b>Expected Results</b>	The SOA rejects the M-EVENT-REPORT with invalid syntax.

11.8.20 MOC.SOA.INV.NOT.subscriptionVersionStatusAttributeValueChange

<b>Purpose</b>	To test the SOA's ability to handle the M-EVENT-REPORT for the subscriptionVersionNPAC's subscriptionVersionStatusAttributeValueChange notification with an invalid syntax for the subscriptionVersionAttributeValueChangeInfo attribute.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	InpSubscriptions test cases .
<b>Procedure</b>	<ol style="list-style-type: none"> <li>3. NPAC SMS Simulator issues subscriptionVersionStatusAttributeValueChange M-EVENT-REPORT with invalid syntax for the subscriptionVersionAttributeValueChangeInfo attribute.</li> <li>4. SOA rejects the M-EVENT-REPORT with invalidArgumentValue error.</li> </ol>
<b>Expected Results</b>	The SOA rejects the M-EVENT-REPORT with invalid syntax.

11.8.21 MOC.SOA.INV.NOT.attributeValueChange.subscriptionVersion

<b>Purpose</b>	To test the SOA's ability to handle the M-EVENT-REPORT for the subscriptionVersionNPAC's attributeValueChange notification for a subscriptionVersion with an invalid accessControl attribute.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	InpSubscriptions test cases
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues attributeValueChange M-EVENT-REPORT for a subscriptionVersion with invalid syntax for the accessControl attribute.</li> <li>2. SOA rejects the M-EVENT-REPORT with an abort.</li> </ol>
<b>Expected Results</b>	The SOA rejects the M-EVENT-REPORT with an abort.

11.8.22 MOC.SOA.INV.NOT.subscriptionVersionNewNPA-NXX

<b>Purpose</b>	To test the SOA's ability to handle the M-EVENT-REPORT request for an invalid subscriptionVersionNewNPA-NXX notification with an invalid NPA-NXX value.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.SOA.VAL.NOT.subscriptionVersionNewNPA-NXX
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues subscriptionVersionNewNPA-NXX M-EVENT-REPORT with invalid value for the NPA-NXX attribute.</li> <li>2. SOA rejects the M-EVENT-REPORT with an invalidArgumentValue error.</li> </ol>
<b>Expected Results</b>	The SOA responds with the appropriate M-EVENT-REPORT error.

11.8.23 MOC.SOA.BND.GET.MAXQ.subscriptionVersionNPAC

<b>Purpose</b>	To test the behaviour of the SOA when it receives the responses to a valid scoped M-GET, which will return the maximum number of records specified in the NPAC SMS Simulator <Max Subscriber Query> parameter. This will be accomplished by requesting all attributes for all the existing subscriptionVersionNPAC managed object instances.
<b>Severity</b>	R
<b>Severity Explanation</b>	Must be performed filter M-GET requests are being used.
<b>Prerequisites</b>	The number of subscriptionVersionNPAC managed object instances created is equal to the Max Subscriber Query parameter.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA sends a valid scoped and filtered M-GET request for subscriptionVersionNPAC data that will result in 'Max Subscriber Query' objects being returned.</li> <li>2. NPAC SMS Simulator responds with the linked M-GET result replies.</li> </ol>
<b>Expected Results</b>	The SOA handles the linked replies properly.

11.8.24 MOC.SOA.INV.QUERY.SCOPE.subsriptionVersion

<b>Purpose</b>	To verify a SOA can handle a scoped filtered query request error.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case should be executed if the SOA will be supporting scoped filtered subscription version query.
<b>Prerequisites</b>	subscriptionVersionNPACs exist on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues a scoped filtered M-GET for a range of subscription versions where the number of subscription versions that satisfy the request exceeds the maximum number of subscription versions that can be retrieved in one request.</li> <li>2. The NPAC SMS Simulator responds with an M-GET error of complexityLimitation.</li> </ol>
<b>Expected Results</b>	The SOA successfully initiates the M-GET and successfully handles the M-GET error response.

11.8.25 MOC.SOA.CAP.NOT.subsriptionVersionNewSP-FinalConcurrenceWindowExpiration

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionNewSP-FinalConcurrenceWindowExpiration M-EVENT-REPORT.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if SOA will be supporting the subscriptionVersionNewSP-FinalConcurrenceWindowExpiration notification.
<b>Prerequisites</b>	A subscription version exists on the NPAC SMS with a subscriptionVersionStatus of 'pending' that was created by the Old Service Provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionNewSP-FinalConcurrenceWindowExpiration to the New Service Provider.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and acknowledges it correctly.

11.8.26 MOC.SOA.INV.NOT.subsriptionVersionNewSP-FinalConcurrenceWindowExpiration

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionNewSP-FinalConcurrenceWindowExpiration M-EVENT-REPORT with invalid syntax for the version-create-request field.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	A subscription version exists on the NPAC SMS with a subscriptionVersionStatus of 'pending' that was created by the Old Service Provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionNewSP-FinalConcurrenceWindowExpiration to the New Service Provider.</li> <li>2. SOA rejects the M-EVENT-REPORT with an invalidArgument error.</li> </ol>
<b>Expected Results</b>	The SOA receives the NPAC SMS Simulator's M-EVENT-REPORT and returns the invalidArgument or other appropriate error.

### 11.9 *serviceProvNetwork*

<b>MO</b>	serviceProvNetwork
<b>Purpose</b>	This section contains the test cases for the serviceProvNetwork Managed Object Class pertaining to the SOA to NPAC SMS Interface, as part of the Managed Object Conformance testing of the NPAC SMS Interoperability Test.
<b>Prerequisite</b>	A Service Provider and Network Data Management association function is established. A InpNPAC-SMS and a InpNetwork Managed Object Instances have been created inherently. A serviceProvNetwork Managed Object Instance has been created locally by the NPAC SMS Simulator Personnel.

#### 11.9.1 MOC.SOA.CAP.OP.GET.serviceProvNetwork

<b>Purpose</b>	To test the SOA's ability to GET all the attributes of the serviceProvNetwork managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-GET request to retrieve all attributes of the serviceProvNetwork object.</li> <li>NPAC SMS Simulator responds with a successful M-GET reply containing all the attributes.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-GET request and retrieves the attributes (i.e. serviceProvId and serviceProvName) successfully from the NPAC SMS Simulator.

#### 11.9.2 MOC.SOA.INV.GET.serviceProvNetwork

<b>Purpose</b>	To test the SOA's ability to handle the M-GET error response invalidFilter error to a previously initiated and valid scoped M-GET request for all the attributes of the serviceProvNetwork object.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. SOA may perform to verify error-handling capabilities.
<b>Prerequisites</b>	MOC.SOA.CAP.OP.GET.serviceProvNetwork
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-GET request to retrieve all the attributes of the serviceProvNetwork object.</li> <li>NPAC SMS Simulator responds with an invalidFilter error.</li> </ol>
<b>Expected Results</b>	The SOA correctly handles the error response invalidFilter error from the NPAC SMS Simulator.

### 11.10 *serviceProvNPA-NXX*

<b>MO</b>	serviceProvNPA-NXX
<b>Purpose</b>	This section contains the test cases for the serviceProvNPA-NXX Managed Object Class pertaining to the SOA to NPAC SMS Interface, as part of the

	Managed Object Conformance testing of the NPAC SMS Interoperability Test.
<b>Prerequisite</b>	A Service Provider and Network Data Management association function is established. A InpNPAC-SMS and a InpNetwork Managed Object Instances have been created inherently.

11.10.1 MOC.SOA.CAP.OP.GET.serviceProvNPA-NXX

<b>Purpose</b>	To test the SOA's ability to GET all the attributes of the serviceProvNPA-NXX managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	MOC.SOA.VAL.CRE.AUTO.serviceProvNPA-NXX
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-GET request for all attributes of the serviceProvNPA-NXX object.</li> <li>NPAC SMS Simulator responds with a successful M-GET result containing all attributes.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-GET request and retrieves the attributes successfully from the NPAC SMS Simulator.

11.10.2 MOC.SOA.CAP.OP.DEL.serviceProvNPA-NXX

<b>Purpose</b>	To test the SOA's ability to DELETE an existing serviceProvNPA-NXX managed object instance from the NPAC SMS Simulator.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if the SOA is managing network data.
<b>Prerequisites</b>	MOC.SOA.VAL.CRE.AUTO.serviceProvNPA-NXX
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-DELETE request for a serviceProvNPA-NXX object.</li> <li>NPAC SMS Simulator responds with a successful M-DELETE reply.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-DELETE request and removes the object successfully from the NPAC SMS Simulator.

11.10.3 MOC.SOA.VAL.CRE.AUTO.serviceProvNPA-NXX

<b>Purpose</b>	To test the SOA's ability to CREATE a serviceProvNPA-NXX managed object instance in the NPAC SMS Simulator using automatic instance naming.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if the SOA is managing network data.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-CREATE request for a serviceProvNPA-NXX object.</li> <li>NPAC SMS Simulator responds with a successful M-CREATE reply.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-CREATE request with automatic

	instance naming causing the serviceProvNPA-NXX instance to be created and its attributes populated successfully in the NPAC SMS Simulator.
--	--

11.10.4 MOC.SOA.VAL.GET.SCOP.FILT.serviceProvNPA-NXX

<b>Purpose</b>	To test the SOA's ability to initiate a valid scoped and filtered M-GET request for all attributes. This will be accomplished by retrieving all the attributes for an agreed upon NPA-NXX value (i.e. filtering on serviceProvNPA-NXX-Value equal to that number) starting at the base managed object serviceProvNetwork and ending at the serviceProvNPA-NXX.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. This functionality may be satisfied by getting one instance at a time.
<b>Prerequisites</b>	A serviceProvNPA-NXX managed object instance with agreed upon serviceProvNPA-NXX-Value attribute has been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid scope and filtered M-GET request for the attributes of a serviceProvNPA-NXX object with a filter for equality on the serviceProvNPA-NXX-Value.</li> <li>NPAC SMS Simulator responds with a successful M-GET reply containing the attributes.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-GET request and retrieves the attributes successfully from the NPAC SMS Simulator.

11.10.5 MOC.SOA.VAL.DEL.SCOP.FILT.serviceProvNPA-NXX

<b>Purpose</b>	To test the SOA's ability to initiate a valid scoped and filtered M-DELETE request for an existing managed object instance. This will be accomplished by deleting the serviceProvNPA-NXX instance with the serviceProvNPA-NXX-Value equal to a specified value starting at the base managed object serviceProvNetwork and ending at the serviceProvNPA-NXX.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. This functionality may be satisfied by deleting one instance at a time.
<b>Prerequisites</b>	A serviceProvNPA-NXX managed object instance with above serviceProvNPA-NXX-Value attribute has been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid scope and filtered M-DELETE request for the serviceProvNPA-NXX object with a filter for equality on the serviceProvNPA-NXX-Value.</li> <li>NPAC SMS Simulator responds with a successful M-DELETE reply.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-DELETE request with a correct scope and filter causing the above instance to be removed successfully from the NPAC SMS Simulator.

11.10.6 MOC.SOA.INV.CRE.serviceProvNPA-NXX

<b>Purpose</b>	To test the SOA's ability to handle the M-CREATE error response
----------------	---

	duplicateManagedObjectInstance error to a previously initiated and valid M-CREATE request for a serviceProvNPA-NXX.
<b>Severity</b>	C
<b>Severity Explanation</b>	Must be performed if prerequisite is performed.
<b>Prerequisites</b>	MOC.SOA.VAL.CRE.AUTO.serviceProvNPA-NXX
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues an M-CREATE request for the serviceProvNPA-NXX object.</li> <li>NPAC SMS Simulator responds with a duplicateManagedObjectInstance error.</li> </ol>
<b>Expected Results</b>	The SOA correctly handles the error response duplicateManagedObjectInstance error from the NPAC SMS Simulator.

11.10.7 MOC.SOA.INV.GET.serviceProvNPA-NXX

<b>Purpose</b>	To test the SOA's ability to handle the M-GET error response operationCancelled error to a previously initiated and valid M-GET request for all the attributes of the serviceProvNPA-NXX object.
<b>Severity</b>	C
<b>Severity Explanation</b>	Must be performed if prerequisite is performed.
<b>Prerequisites</b>	MOC.SOA.CAP.OP.GET.serviceProvNPA-NXX
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-GET request for all the attributes of a serviceProvNPA-NXX object.</li> <li>NPAC SMS Simulator responds with an operationCancelled error.</li> </ol>
<b>Expected Results</b>	The SOA correctly handles the error response operationCancelled error from the NPAC SMS Simulator.

11.10.8 MOC.SOA.INV.DEL.serviceProvNPA-NXX

<b>Purpose</b>	To test the SOA's ability to handle the M-DELETE error response processingFailure error to a previously initiated and valid M-DELETE request for an existing serviceProvNPA-NXX managed object instance.
<b>Severity</b>	C
<b>Severity Explanation</b>	Must be performed if prerequisite is performed.
<b>Prerequisites</b>	MOC.SOA.CAP.OP.DEL.serviceProvNPA-NXX
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-DELETE request for a serviceProvNPA-NXX object.</li> <li>NPAC SMS Simulator responds with a processingFailure error.</li> </ol>
<b>Expected Results</b>	The SOA correctly handles the error response processingFailure error from the NPAC SMS Simulator.

11.11 serviceProvLRN

<b>MO</b>	serviceProvLRN
<b>Purpose</b>	This section contains the test cases for the serviceProvLRN Managed Object Class pertaining to the SOA to NPAC SMS Interface, as part of the Managed

	Object Conformance testing of the NPAC SMS Interoperability Test.
<b>Prerequisite</b>	A Service Provider and Network Data Management is established. A InpNPAC-SMS and a InpNetwork Managed Object Instances have been created inherently.

11.11.1 MOC.SOA.CAP.OP.GET.serviceProvLRN

<b>Purpose</b>	To test the SOA's ability to GET all the attributes of the serviceProvLRN managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	MOC.SOA.VAL.CRE.AUTO.serviceProvLRN
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-GET request to retrieve all attributes of a serviceProvLRN.</li> <li>NPAC SMS Simulator responds with the M-GET result.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-GET request and retrieves the attributes successfully from the NPAC SMS Simulator.

11.11.2 MOC.SOA.CAP.OP.DEL.serviceProvLRN

<b>Purpose</b>	To test the SOA's ability to DELETE an existing serviceProvLRN managed object instance from the NPAC SMS Simulator.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if the SOA is managing network data.
<b>Prerequisites</b>	MOC.SOA.VAL.CRE.AUTO.serviceProvLRN
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-DELETE request for a serviceProvLRN.</li> <li>NPAC SMS Simulator responds with the successful M-DELETE reply.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-DELETE request and removes the object successfully from the NPAC SMS Simulator.

11.11.3 MOC.SOA.VAL.CRE.AUTO.serviceProvLRN

<b>Purpose</b>	To test the SOA's ability to CREATE a serviceProvLRN managed object instance in the NPAC SMS Simulator using automatic instance naming.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if SOA is managing network data.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-CREATE request for a serviceProvLRN.</li> <li>NPAC SMS Simulator responds with the successful M-CREATE reply.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-CREATE request with automatic instance naming causing the serviceProvLRN instance to be created and its attributes populated successfully in the NPAC SMS Simulator.



11.11.4 MOC.SOA.VAL.GET.SCOP.FILT.serviceProvLRN

<b>Purpose</b>	To test the SOA's ability to initiate a valid scoped and filtered M-GET request for a single attribute. This will be accomplished by retrieving all the attributes for an agreed upon LRN value (i.e. filtering on serviceProvLRN-Value equal to that number) starting at the serviceProvNetwork and ending at the serviceProvLRN.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. This functionality may be satisfied by getting one instance at a time.
<b>Prerequisites</b>	A serviceProvLRN managed object instance with above serviceProvLRN-Value attribute has been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-GET request for the attributes of a serviceProvLRN.</li> <li>NPAC SMS Simulator responds with the successful M-GET result containing the attribute.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-GET request and retrieves the attribute successfully from the NPAC SMS Simulator.

11.11.5 MOC.SOA.VAL.DEL.SCOP.FILT.serviceProvLRN

<b>Purpose</b>	To test the SOA's ability to initiate a valid scoped and filtered M-DELETE request for an existing managed object instance. This will be accomplished by deleting all the serviceProvLRN instances with the serviceProvLRN-Value equal to a specified LRN value starting at the serviceProvNetwork and ending at the serviceProvLRN.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. May be used to satisfy the requirements instead of Test Case MOC.SOA.CAP.OP.DEL.serviceProvLRN.
<b>Prerequisites</b>	A serviceProvLRN managed object instance with above serviceProvLRN-Value attribute have been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-DELETE request for the serviceProvLRN-Value attribute of a serviceProvLRN with the filter set to equality for the specified LRN value.</li> <li>NPAC SMS Simulator responds with the successful M-DELETE reply.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-DELETE request with a correct scope and filter causing the above instance to be removed successfully from the NPAC SMS Simulator and successfully handles the reply.

11.11.6 MOC.SOA.INV.CRE.serviceProvLRN

<b>Purpose</b>	To test the SOA's ability to handle the M-CREATE error response duplicateManagedObjectInstance error to a previously initiated and valid M-CREATE request for a serviceProvLRN.
<b>Severity</b>	C
<b>Severity Explanation</b>	Must be performed if prerequisite is performed.
<b>Prerequisites</b>	MOC.SOA.VAL.CRE.AUTO.serviceProvLRN

<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues a valid M-CREATE request a serviceProvLRN.</li> <li>2. NPAC SMS Simulator responds with the duplicateManagedObjectInstance error.</li> </ol>
<b>Expected Results</b>	The SOA correctly handles the error response duplicateManagedObjectInstance error from the NPAC SMS Simulator.

11.11.7 MOC.SOA.INV.GET.serviceProvLRN

<b>Purpose</b>	To test the SOA's ability to handle the M-GET error response operationCancelled error to a previously initiated and valid M-GET request for all the attributes of the serviceProvLRN object.
<b>Severity</b>	C
<b>Severity Explanation</b>	Must be performed if prerequisite is performed.
<b>Prerequisites</b>	MOC.SOA.CAP.OP.GET.serviceProvLRN
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues a valid M-GET request for all the attributes of a serviceProvLRN object.</li> <li>2. NPAC SMS Simulator responds with the operationCancelled error.</li> </ol>
<b>Expected Results</b>	The SOA correctly handles the error response operationCancelled error from the NPAC SMS Simulator.

11.11.8 MOC.SOA.INV.DEL.serviceProvLRN

<b>Purpose</b>	To test the SOA's ability to handle the M-DELETE error response processingFailure error to a previously initiated and valid M-DELETE request for an existing serviceProvLRN managed object instance.
<b>Severity</b>	C
<b>Severity Explanation</b>	Must be performed if prerequisite is performed.
<b>Prerequisites</b>	MOC.SOA.CAP.OP.DEL.serviceProvLRN
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues a valid M-DELETE request for a serviceProvLRN object.</li> <li>2. NPAC SMS Simulator responds with the processingFailure error.</li> </ol>
<b>Expected Results</b>	The SOA correctly handles the error response processingFailure error from the NPAC SMS Simulator.

**11.12 numberPoolBlockNPAC**

<b>MO</b>	numberPoolBlockNPAC
<b>Purpose</b>	This section contains the test cases for the numberPoolBlockNPAC Managed Object Class pertaining to the SOA to NPAC SMS Interface as part of the MOC testing of the NPAC SMS Simulator Interoperability Test.
<b>Prerequisite</b>	<ul style="list-style-type: none"> <li>• A LSMS Network and Subscription Data Download Management association function is established.</li> <li>• InpNPAC-SMS and InpSubscription Managed Object Instances exist.</li> </ul>

11.12.1 MOC.SOA.CAP.OP.GET.numberPoolBlockNPAC

<b>Purpose</b>	Verify the SOA's ability to GET all the attributes of the numberPoolBlockNPAC managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	A numberPoolBlockNPAC object exists on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-GET request for all attributes of the numberPoolBlockNPAC object.</li> <li>NPAC SMS Simulator responds with a successful M-GET result containing all attributes.</li> </ol>
<b>Expected Results</b>	SOA issues a valid M-GET request and retrieves the attributes successfully from the NPAC SMS Simulator.

11.12.2 MOC.SOA.CAP.OP.SET.numberPoolBlockNPAC

<b>Purpose</b>	Verify the SOA's ability to SET all the modifiable attributes of the numberPoolBlockNPAC managed object instance.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if the SOA is supporting modification of number pool blocks.
<b>Prerequisites</b>	A numberPoolBlockNPAC object exists on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-SET request for all modifiable attributes of the numberPoolBlockNPAC object.</li> <li>NPAC SMS Simulator responds with a successful M-SET result containing all modifiable attributes.</li> </ol>
<b>Expected Results</b>	SOA issues a valid M-SET request and updates the attributes successfully on the NPAC SMS Simulator.

11.12.3 MOC.SOA.VAL.GET.SCOP.numberPoolBlockNPAC

<b>Purpose</b>	Verify the SOA's ability to initiate a valid scoped M-GET request for all the attributes of the numberPoolBlockNPAC managed object instance. This will be accomplished by retrieving all the attributes starting at the base managed object InpSubscriptions and ending at the numberPoolBlockNPAC.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	Multiple numberPoolBlockNPAC objects exist on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid scoped and filtered M-GET request for the numberPoolBlockNPAC object(s).</li> <li>NPAC SMS Simulator responds with a successful M-GET result containing all attributes.</li> </ol>
<b>Expected Results</b>	SOA issues a valid M-GET request and retrieves the object(s) successfully from the NPAC SMS Simulator.

11.12.4 MOC.SOA.INV.GET.numberPoolBlockNPAC

<b>Purpose</b>	Verify the SOA's ability to handle an error response, processingFailure error, to a previously initiated and valid M-GET
----------------	--

	request for all the attributes of a numberPoolBlockNPAC managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Required if MOC.SOA.CAP.OP.GET.numberPoolBlockNPAC is performed.
<b>Prerequisites</b>	A numberPoolBlockNPAC exists on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-GET request for all attributes of the numberPoolBlockNPAC object.</li> <li>NPAC SMS Simulator responds with a processingFailure error.</li> </ol>
<b>Expected Results</b>	SOA successfully handles the error response from the NPAC SMS Simulator.

#### 11.12.5 MOC.SOA.INV.SET.numberPoolBlockNPAC

<b>Purpose</b>	Verify the SOA's ability to handle an error response, processingFailure error, to a previously initiated and valid M-SET request for all the attributes of a numberPoolBlockNPAC managed object instance.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if MOC.SOA.CAP.OP.SET.numberPoolBlockNPAC is performed.
<b>Prerequisites</b>	A numberPoolBlockNPAC exists on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-SET request to update all attributes of the numberPoolBlockNPAC object.</li> <li>NPAC SMS Simulator responds with a processingFailure error.</li> </ol>
<b>Expected Results</b>	SOA successfully handles the error response from the NPAC SMS Simulator.

#### 11.12.6 MOC.SOA.INV.GET.SCOPE.numberPoolBlockNPAC

<b>Purpose</b>	Verify the SOA's ability to handle an error response, processingFailure, to a previously initiated and valid scope and filtered M-GET request for all the attributes of a numberPoolBlockNPAC managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Required if MOC.SOA.CAP.OP.GET.SCOPE.numberPoolBlockNPAC is performed.
<b>Prerequisites</b>	A numberPoolBlockNPAC exists on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid scope and filtered M-GET request for all attributes of the numberPoolBlockNPAC object.</li> <li>NPAC SMS Simulator responds with a processingFailure error.</li> </ol>
<b>Expected Results</b>	SOA successfully handles the error response from the NPAC SMS Simulator.

### 11.13 serviceProvNPA-NXX-X

<b>MO</b>	numberPoolBlockNPAC
<b>Purpose</b>	This section contains the test cases for the serviceProvNPA-NXX-X Managed Object Class pertaining to the SOA to NPAC SMS Interface as part of the MOC testing of the NPAC SMS Simulator Interoperability Test.
<b>Prerequisite</b>	<ul style="list-style-type: none"> <li>A SOA Service Provider and Network Data Download Management</li> </ul>

	association function is established. • InpNPAC-SMS and InpNetwork Managed Object Instances exist.
--	--

11.13.1 MOC.SOA.CAP.OP.GET.serviceProvNPA-NXX-X

<b>Purpose</b>	Verify the SOA's ability to GET all the attributes of the serviceProvNPA-NXX-X managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	A serviceProvNPA-NXX-X object exists on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues a valid M-GET request for all attributes of the serviceProvNPA-NXX-X object.</li> <li>2. NPAC SMS Simulator responds with a successful M-GET result containing all attributes.</li> </ol>
<b>Expected Results</b>	SOA issues a valid M-GET request and retrieves the attributes successfully from the NPAC SMS Simulator.

11.13.2 MOC.SOA.VAL.GET.SCOP.serviceProvNPA-NXX-X

<b>Purpose</b>	Verify the SOA's ability to initiate a valid scoped M-GET request for all the attributes of the serviceProvNPA-NXX-X managed object instance. This will be accomplished by retrieving all the attributes starting at the base managed object InpSubscriptions and ending at the serviceProvNPA-NXX-X.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	Multiple numberPoolBlockNPAC objects exist on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues a valid scoped and filtered M-GET request for the serviceProvNPA-NXX-X object(s).</li> <li>2. NPAC SMS Simulator responds with a successful M-GET result containing all attributes.</li> </ol>
<b>Expected Results</b>	SOA issues a valid M-GET request and retrieves the object(s) successfully from the NPAC SMS Simulator.

11.13.3 MOC.SOA.INV.GET.serviceProvNPA-NXX-X

<b>Purpose</b>	Verify the SOA's ability to handle an error response, processingFailure error, to a previously initiated and valid M-GET request for all the attributes of a serviceProvNPA-NXX-X managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Required if MOC.SOA.CAP.OP.GET.serviceProvNPA-NXX-X is performed.
<b>Prerequisites</b>	A serviceProvNPA-NXX-X exists on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues a valid M-GET request for all attributes of the serviceProvNPA-NXX-X object.</li> <li>2. NPAC SMS Simulator responds with a processingFailure error.</li> </ol>
<b>Expected Results</b>	SOA successfully handles the error response from the NPAC SMS Simulator.

11.13.4 MOC.SOA.INV.GET.SCOP.serviceProvNPA-NXX-X

<b>Purpose</b>	Verify the SOA's ability to handle an error response, processingFailure, to a previously initiated and valid scope and filtered M-GET request for all the attributes of a serviceProvNPA-NXX-X managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Required if MOC.SOA.CAP.OP.GET.SCOP.serviceProvNPA-NXX-X is performed.
<b>Prerequisites</b>	A serviceProvNPA-NXX-X exists on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues a valid scope and filtered M-GET request for all attributes of the serviceProvNPA-NXX-X object.</li> <li>2. NPAC SMS Simulator responds with a processingFailure error.</li> </ol>
<b>Expected Results</b>	SOA successfully handles the error response from the NPAC SMS Simulator.

## 12 NPAC SMS to SOA MOC Test Cases

### 12.1 InpSOA

<b>MO</b>	InpSOA
<b>Purpose</b>	These test cases are for the InpSOA Managed Object Class pertaining to the NPAC SMS to SOA Interface, as part of the MO Conformance testing of the interoperability test. This capability test package checks the SOA existence and basic validity of the specified capabilities. This object is used to support network data download to the SOA.
<b>Prerequisite</b>	<ul style="list-style-type: none"> <li>• A NPAC Management association function is established with the NPAC SMS Simulator.</li> <li>• The SOA has successfully completed the Stack-to-Stack Interoperability testing.</li> </ul>

#### 12.1.1 MOC.NPAC.CAP.OP.GET.InpSOA

<b>Purpose</b>	Verifies the SOA capability to correctly respond to a InpSOA MO class M-GET request with all the attributes.
<b>Severity</b>	O
<b>Severity Explanation</b>	Needed for NPAC SMS Simulator to verify correct creation of inherent instance by SOA for network data download. The NPAC SMS will not issue such a request.
<b>Prerequisites</b>	A InpSOA instance has been inherently created on the SOA.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC sends an M-GET request for InpSOA.</li> <li>2. SOA responds with getResult.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a getResult with the correct attribute information for all attributes.

#### 12.1.2 MOC.NPAC.INV.CRE.INH.InpSOA

<b>Purpose</b>	This test case checks the SOA ability of responding to a semantically invalid CMIP request. The NPAC SMS Simulator sends M-CREATE request intending to create an instance that can only be created inherently on the SOA.
<b>Severity</b>	O
<b>Severity Explanation</b>	Needed to test error-handling capabilities of SOA if the SOA supports network data download. NPAC will not issue such a request.
<b>Prerequisites</b>	A InpSOA instance exists on the SOA.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends M-CREATE request.</li> <li>2. SOA responds with an error response of processingFailure or duplicateManagedObjectInstance error.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives an M-CREATE error response. No instance is created on the SOA.

#### 12.1.3 MOC.NPAC.INV.SET.InpSOA

<b>Purpose</b>	This test case checks the SOA ability to respond to a semantically invalid CMIP request. The NPAC SMS Simulator sends out an M-
----------------	---

	SET intending to override the read-only attribute InpSOA-Name.
<b>Severity</b>	O
<b>Severity Explanation</b>	Needed to test error-handling capabilities of SOA if the SOA supports network data download. NPAC SMS may not issue such a request.
<b>Prerequisites</b>	A InpSOA instance exists on the SOA.
<b>Procedure</b>	1. NPAC SMS Simulator sends the M-SET request. 2. SOA responds with setListError error response.
<b>Expected Results</b>	The NPAC SMS Simulator receives a setListError error response. The attribute is not updated.

12.1.4 MOC.NPAC.INV.DEL.InpSOA

<b>Purpose</b>	This test case checks the SOA ability to respond to a semantically invalid CMIP request. The NPAC SMS Simulator sends out an M-DELETE request intending to delete the InpSOA instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Needed to test error-handling capabilities of SOA if the SOA supports network data download.
<b>Prerequisites</b>	A InpSOA instance exists on the SOA.
<b>Procedure</b>	1. NPAC SMS Simulator sends the M-DELETE request. 2. SOA responds with processingFailureEr error response.
<b>Expected Results</b>	The NPAC SMS Simulator receives a processingFailureEr error response. No instance is removed from the SOA.

12.2 InpNetwork

<b>MO</b>	InpNetwork
<b>Purpose</b>	This section contains test cases for the InpNetwork Managed Object Class pertaining to the NPAC SMS manager to SOA Interface, as part of the MO Conformance testing of the interoperability test. This capability test package checks the SOA existence and basic validity of the specified capabilities. This object is used to support network data download to the SOA.
<b>Prerequisite</b>	<ul style="list-style-type: none"> <li>• A NPAC Management association function is established.</li> <li>• The agent has successfully completed the Stack-to-Stack Interoperability testing.</li> </ul>

12.2.1 MOC.NPAC.SOA.CAP.OP.GET.InpNetwork

<b>Purpose</b>	Tests the capability of the SOA to correctly respond to an M-GET request for the InpNetwork MO class. The NPAC SMS Simulator will get all attributes of the MO instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Needed for the NPAC SMS Simulator to verify correct creation of inherent instance by SOA if the SOA is supporting network data download.
<b>Prerequisites</b>	A InpNetwork instance exists on SOA.
<b>Procedure</b>	1. NPAC SMS Simulator sends an M-GET request for InpNetwork for all attributes. 2. SOA responds with getResult.
<b>Expected Results</b>	The NPAC SMS Simulator receives a getResult with all attribute



	values.
--	---------

### 12.2.2 MOC.NPAC.SOA.INV.CRE.INH.InpNetwork

<b>Purpose</b>	Tests the ability of the SOA to respond to an invalid CMIP M-CREATE request. The NPAC SMS Simulator sends out an M-CREATE intending to create an instance that can only be created inherently on the SOA.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact the SOA ability to provide LNP service for network data download. SOA may perform to verify error handling.
<b>Prerequisites</b>	A InpSOA instance exists on SOA.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an M-CREATE request for InpNetwork.</li> <li>2. SOA responds with an M-CREATE error.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives an error response with error type set to processingFailureEr or duplicateObjectInstanceEr.

### 12.2.3 MOC.NPAC.SOA.INV.SET.InpNetwork

<b>Purpose</b>	Tests the ability of the SOA to respond to an invalid CMIP M-SET request. The NPAC SMS Simulator sends out an M-SET intending to override the read-only attribute InpNetworkName.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact the SOA ability to provide LNP service for network data download. SOA may perform to verify error handling.
<b>Prerequisites</b>	A InpNetwork instance exists on the SOA.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an M-SET request for InpNetwork InpNetworkName attribute.</li> <li>2. SOA responds with an M-SET error.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives an error response with the error type set to setListErrorEr.

### 12.2.4 MOC.NPAC.SOA.INV.ACT.InpNetwork

<b>Purpose</b>	Tests the ability of the SOA to respond to an invalid CMIP M-ACTION request. The manager sends a InpDownload action.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact the SOA ability to provide LNP service for network data download. SOA may perform to verify error handling..
<b>Prerequisites</b>	A InpNetwork instance exists on the SOA.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a InpDownload M-ACTION request.</li> <li>2. SOA responds with an M-ACTION error.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives an error response with error type set to noSuchActionEr.

12.2.5 MOC.NPAC.SOA.INV.DEL.InpNetwork

<b>Purpose</b>	Tests the ability of the SOA to respond to an invalid CMIP M-DELETE request. The manager sends a delete for the InpNetwork MO.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact the SOA ability to provide LNP service for network data download. SOA may perform to verify error handling.
<b>Prerequisites</b>	A InpNetwork instance exists on the SOA.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends M-DELETE request for the InpNetwork MO.</li> <li>2. SOA responds with an M-DELETE error.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives the error response with error type set to processingFailureEr.

12.3 serviceProvNetwork

<b>MO</b>	serviceProvNetwork
<b>Purpose</b>	This section contains test cases for the InpLocalSMS Managed Object Class pertaining to the NPAC SMS manager to Local SMS agent Interface, as part of the MO Conformance testing of the interoperability test. This capability test package checks the agent's existence and basic validity of the specified capabilities.
<b>Prerequisite</b>	A NPAC Management association function is established. The agent has successfully completed the Stack-to-Stack Interoperability testing. The agent has successfully completed the MOC.NPAC.CAP.InpLocalSMS test.

12.3.1 MOC.NPAC.SOA.CAP.OP.CRE.serviceProvNetwork

<b>Purpose</b>	Test the capability of the SOA to correctly respond to an M-CREATE request for the serviceProvNetwork MO.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the SOA is to supports network data download.
<b>Prerequisites</b>	A InpNetwork instance exists on the SOA
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a serviceProvNetwork M-CREATE request.</li> <li>2. SOA responds successfully to the M-CREATE.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator sends a valid M-CREATE request and receives the SOA M-CREATE response indicating successful creation of the serviceProvNetwork.

12.3.2 MOC.NPAC.SOA.CAP.OP.GET.serviceProvNetwork

<b>Purpose</b>	Tests the capability of the SOA to correctly respond to an M-GET request for the serviceProvNetwork MO class. The NPAC SMS Simulator will get all attributes of the MO instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Needed for the NPAC SMS Simulator to verify correct instantiation by SOA if the SOA is supporting network data download.

<b>Prerequisites</b>	A serviceProvNetwork instance exists on SOA.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an M-GET request for serviceProvNetwork for all attributes.</li> <li>2. SOA responds with getResult.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a getResult with all attribute values.

### 12.3.3 MOC.NPAC.SOA.CAP.OP.SET.serviceProvNetwork

<b>Purpose</b>	Tests the capability of the SOA to support the M-SET of the serviceProvName attribute in the serviceProvNetwork MO class.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if the SOA supports network data download.
<b>Prerequisites</b>	A serviceProvNetwork exists on the SOA.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an M-SET request for serviceProvNetwork serviceProvName attribute.</li> <li>2. SOA responds successfully to the M-SET</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator sends an M-SET request to the SOA and the SOA responds successfully.

### 12.3.4 MOC.NPAC.SOA.CAP.OP.DEL.serviceProvNetwork

<b>Purpose</b>	Tests the capability of the agent to support the M-DELETE request for the serviceProvNetworkMO class.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if the SOA supports network data download.
<b>Prerequisites</b>	A serviceProvNetwork exists on the SOA.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an M-DELETE request for the serviceProvNetwork MO.</li> <li>2. SOA responds successfully to the M-DELETE.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator sends an M-DELETE request to the SOA and the SOA responds successfully.

### 12.3.5 MOC.NPAC.SOA.INV.CRE.DUP.serviceProvNetwork

<b>Purpose</b>	Test the capability of the SOA to correctly respond to a duplicate M-CREATE request for the serviceProvNetwork MO.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact the SOA ability to provide LNP service for network data download. SOA may perform to verify error handling.
<b>Prerequisites</b>	A serviceProvNetwork instance exists on the SOA.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a serviceProvNetwork M-CREATE request for a serviceProvNetwork MO that already exists.</li> <li>2. SOA responds to the M-CREATE.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator sends an M-CREATE request for an

	existing serviceProvNetwork MO and receives the SOA M-CREATE error response of duplicateObjectInstanceEr.
--	---

### 12.3.6 MOC.NPAC.SOA.INV.SET.RO.serviceProvNetwork

<b>Purpose</b>	Tests the capability of the SOA to respond to an M-SET for the read-only attribute serviceProvID.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact the SOA ability to provide LNP service for network data download. SOA may perform to verify error handling.
<b>Prerequisites</b>	A serviceProvNetwork instance exists on the SOA.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an M-SET request for a serviceProvNetwork serviceProvID attribute.</li> <li>2. SOA responds with an M-SET error.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives an error response with the error type set to setListErrorEr.

### 12.3.7 MOC.NPAC.SOA.INV.SET.SYN.serviceProvNetwork

<b>Purpose</b>	Tests the capability of the SOA to respond to an M-SET for a syntactically invalid CMIP.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact the SOA ability to provide LNP service for network data download. SOA may perform to verify error handling.
<b>Prerequisites</b>	A serviceProvNetwork instance exists on the SOA.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an M-SET request for a serviceProvNetwork serviceProvName attribute with a length of 41.</li> <li>2. SOA responds with an M-SET error.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives an error response with the error type set to processingFailureEr.

### 12.3.8 MOC.NPAC.SOA.INV.SET.serviceProvNetwork

<b>Purpose</b>	Tests the capability of the SOA to respond to M-SET for a syntactically invalid CMIP request for the serviceProvName attribute.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact the SOA ability to provide LNP service for network data download. SOA may perform to verify error handling..
<b>Prerequisites</b>	A serviceProvNetwork instance exists on the SOA.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an M-SET request for a serviceProvNetwork serviceProvName attribute with a length of 0.</li> <li>2. SOA responds with an M-SET error.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives an error response with the error type set to processingFailureEr.

12.3.9 MOC.NPAC.SOA.INV.GET.serviceProvNetwork

<b>Purpose</b>	Tests the capability of the SOA to respond to an invalid M-GET for an attribute that does not exist.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. SOA may perform to verify error handling..
<b>Prerequisites</b>	A serviceProvNetwork instance exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an M-GET request for a serviceProvNetwork for an attribute that does not exist.</li> <li>2. SOA responds with an M-SET error.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives an error response with the error type set to getListErrorEr.

12.3.10 MOC.NPAC.SOA.INV.DEL.serviceProvNetwork

<b>Purpose</b>	Tests the ability of the SOA to respond to an invalid CMIP M-DELETE request. The manager sends a delete for a serviceProvNetwork MO that does not exist.
<b>Severity</b>	C
<b>Severity Explanation</b>	Must be performed if prerequisite is performed.
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• MOC.NPAC.SOA.CAP.OP.DEL.serviceProvNetwork</li> <li>• The serviceProvNetwork MO to be deleted does not exist.</li> </ul>
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends M-DELETE request for the serviceProvNetwork MO.</li> <li>2. SOA responds with an M-DELETE error.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives the error response with error type set to noSuchObjectInstanceEr.

12.3.11 MOC.NPAC.SOA.INV.DEL.CO.serviceProvNetwork

<b>Purpose</b>	Tests the ability of the SOA to respond to an invalid CMIP M-DELETE request. The manager sends a delete for the serviceProvNetwork MO that is a container for subordinate objects.
<b>Severity</b>	C
<b>Severity Explanation</b>	Must be performed if prerequisite is performed.
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• MOC.NPAC.SOA.CAP.OP.DEL.serviceProvNetwork</li> <li>• The serviceProvNetwork MO to be deleted does not exist.</li> </ul>
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends M-DELETE request for the serviceProvNetwork MO that is a container for subordinate objects.</li> <li>2. SOA responds with an M-DELETE error.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives the error response with error type set to processingFailureEr.

12.3.12 MOC.NPAC.SOA.BND.SET.MIN.serviceProvNetwork

<b>Purpose</b>	Tests the capability of the SOA to support the M-SET of the serviceProvName attribute to a string length of one in the
----------------	--

	serviceProvNetwork MO class.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case should be executed if the SOA supports network data download.
<b>Prerequisites</b>	A serviceProvNetwork exists on the SOA.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an M-SET request for serviceProvNetwork serviceProvName attribute to a value with the minimum length of 1.</li> <li>2. SOA responds successfully to the M-SET</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator sends an M-SET request to the SOA and the SOA responds successfully.

12.3.13 MOC.NPAC.SOA.BND.SET.MAX.serviceProvNetwork

<b>Purpose</b>	Tests the capability of the SOA to support the M-SET of the serviceProvName attribute to a string length of 40 in the serviceProvNetwork MO class.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case should be executed if the SOA supports network data download.
<b>Prerequisites</b>	A serviceProvNetwork exists on the SOA.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an M-SET request for serviceProvNetwork serviceProvName attribute to a value with the maximum length of 40.</li> <li>2. SOA responds successfully to the M-SET</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator sends an M-SET request to the SOA and the SOA responds successfully.

12.4 serviceProvNPA-NXX

<b>MO</b>	serviceProvNPA-NXX
<b>Purpose</b>	This section contains test cases for the serviceProvNPA-NXX Managed Object Class pertaining to the NPAC SMS manager to SOA Interface, as part of the MO Conformance testing of the interoperability test. This capability test checks the existence and the basic validity of the SOA capability. This object is used to support network data download to the SOA.
<b>Prerequisite</b>	<ul style="list-style-type: none"> <li>• A NPAC Management association function is established.</li> <li>• The agent has successfully completed serviceProvNetwork MOC testing.</li> <li>• A serviceProvNetwork instance exists on Local SMS.</li> </ul>

12.4.1 MOC.NPAC.SOA.CAP.OP.CRE.serviceProvNPA-NXX

<b>Purpose</b>	Test the capability of the SOA to correctly respond to an M-CREATE request for the serviceProvNPA-NXX MO.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the SOA is to supports network data download.
<b>Prerequisites</b>	A serviceProvNetwork instance exists on the SOA
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a serviceProvNPA-NXX M-CREATE request.</li> </ol>

	2. SOA responds successfully to the M-CREATE.
<b>Expected Results</b>	The NPAC SMS Simulator sends a valid M-CREATE request and receives the SOA M-CREATE response indicating successful creation of the serviceProvNetwork.

#### 12.4.2 MOC.NPAC.SOA.CAP.OP.DEL.serviceProvNPA-NXX

<b>Purpose</b>	Tests the capability of the agent to support the M-DELETE request for the serviceProvNPA-NXX MO class.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if the SOA supports network data download.
<b>Prerequisites</b>	A serviceProvNPA-NXX exists on the SOA.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an M-DELETE request for the serviceProvNPA-NXX MO.</li> <li>2. SOA responds successfully to the M-DELETE</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator sends an M-DELETE request to the SOA and the SOA responds successfully.

#### 12.4.3 MOC.NPAC.SOA.INV.CRE.DUP.serviceProvNPA-NXX

<b>Purpose</b>	Test the capability of the SOA to correctly respond to a duplicate M-CREATE request for the serviceProvNetwork MO.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact the SOA ability to provide LNP service for network data download. SOA may perform to verify error handling.
<b>Prerequisites</b>	A serviceProvNPA-NXX instance exists on the SOA
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a serviceProvNPA-NXX M-CREATE request for a serviceProvNetwork MO that already exists.</li> <li>2. SOA responds to the M-CREATE.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator sends an M-CREATE request for an existing serviceProvNPA-NXX MO and receives the SOA M-CREATE error response of duplicateObjectInstanceEr.

#### 12.4.4 MOC.NPAC.SOA.INV.SET.serviceProvNPA-NXX

<b>Purpose</b>	Tests the capability of the SOA to respond to an M-SET for a syntactically invalid CMIP request for the read-only attribute serviceProvNPA-NXX-ID.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact the SOA ability to provide LNP service for network data download. SOA may perform to verify error handling.
<b>Prerequisites</b>	A serviceProvNPA-NXX instance exists on the SOA.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an M-SET request for a serviceProvNPA-NXX-ID.</li> </ol>

	2. SOA responds with an M-SET error.
<b>Expected Results</b>	The NPAC SMS Simulator receives an error response with the error type set to setListErrorEr.

#### 12.4.5 MOC.NPAC.SOA.INV.DEL.serviceProvNPA-NXX

<b>Purpose</b>	Tests the ability of the SOA to respond to an invalid CMIP M-DELETE request. The manager sends a delete for the serviceProvNPA-NXX MO that does not exist.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the SOA is to support network data download.
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• MOC.NPAC.SOA.CAP.OP.DEL.serviceProvNPA-NXX</li> <li>• The serviceProvNPA-NXX MO to be deleted does not exist.</li> </ul>
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends M-DELETE request for the serviceProvNPA-NXX MO.</li> <li>2. SOA responds with an M-DELETE error.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives the error response with error type set to noSuchObjectInstanceEr.

### 12.5 ServiceProvLRN

<b>MO</b>	serviceProvLRN
<b>Purpose</b>	This file contains test cases for the serviceProvLRN Managed Object Class pertaining to the NPAC SMS manager to SOA Interface, as part of the MO Conformance testing of the interoperability test. This capability test package checks the SOA existence and basic validity of the specified capabilities. This object is used to support network data download to the SOA.
<b>Prerequisite</b>	<ul style="list-style-type: none"> <li>• A NPAC Management association function is established.</li> <li>• The agent has successfully completed the Stack-to-Stack Interoperability testing.</li> <li>• The SOA has successfully completed the serviceProvNetwork MOC tests.</li> <li>• There is a serviceProvNetwork existing on SOA.</li> </ul>

#### 12.5.1 MOC.NPAC.SOA.CAP.OP.CRE.serviceProvLRN

<b>Purpose</b>	Test the capability of the SOA to correctly respond to an M-CREATE request for the serviceProvLRN MO.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the SOA is to supports network data download.
<b>Prerequisites</b>	A serviceProvNetwork instance exists on the SOA
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a serviceProvLRN M-CREATE request.</li> <li>2. SOA responds successfully to the M-CREATE.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator sends a valid M-CREATE request and receives the SOA M-CREATE response indicating successful creation of the serviceProvNetwork.



12.5.2 MOC.NPAC.SOA.CAP.OP.DEL.serviceProvLRN

<b>Purpose</b>	Tests the capability of the agent to support the M-DELETE request for the serviceProvLRN MO class.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if the SOA supports network data download.
<b>Prerequisites</b>	A serviceProvLRN exists on the SOA.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an M-DELETE request for the serviceProvLRN MO.</li> <li>2. SOA responds successfully to the M-DELETE.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator sends an M-DELETE request to the SOA and the SOA responds successfully.

12.5.3 MOC.NPAC.SOA.INV.CRE.DUP.serviceProvLRN

<b>Purpose</b>	Test the capability of the SOA to correctly respond to a duplicate M-CREATE request for the serviceProvNetwork MO.
<b>Severity</b>	C
<b>Severity Explanation</b>	Must be performed if prerequisite is performed.
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• MOC.NPAC.SOA.CAP.OP.CRE.serviceProvLRN</li> <li>• A serviceProvLRN instance exists on the SOA</li> </ul>
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a serviceProvLRN M-CREATE request for a serviceProvNetwork MO that already exists.</li> <li>2. SOA responds to the M-CREATE.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator sends an M-CREATE request for an existing serviceProvLRN MO and receives the SOA M-CREATE error response of duplicateObjectInstanceEr.

12.5.4 MOC.NPAC.SOA.INV.SET.serviceProvLRN

<b>Purpose</b>	Tests the capability of the SOA to respond to M-SET for a syntactically invalid CMIP request for the read-only attribute serviceProvLRN-ID.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact the SOA ability to provide LNP service for network data download. SOA may perform to verify error handling.
<b>Prerequisites</b>	A serviceProvLRN instance exists on the SOA.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an M-SET request for a serviceProvLRN-ID.</li> <li>2. SOA responds with an M-SET error.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives an error response with the error type set to setListErrorEr.

12.5.5 MOC.NPAC.SOA.INV.DEL.serviceProvLRN

<b>Purpose</b>	Tests the ability of the SOA to respond to an invalid CMIP M-DELETE request. The manager sends a delete for the serviceProvLRN MO that does not exist.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the SOA is to support network data download.
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>MOC.NPAC.SOA.CAP.OP.DEL.serviceProvLRN</li> <li>The serviceProvLRN MO to be deleted does not exist.</li> </ul>
<b>Procedure</b>	<ol style="list-style-type: none"> <li>NPAC SMS Simulator sends M-DELETE request for the serviceProvLRN MO.</li> <li>SOA responds with an M-DELETE error.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives the error response with error type set to noSuchObjectInstanceEr.

12.6 numberPoolBlockNPAC

12.6.1 MOC.SOA.CAP.NOT.numberPoolBlockAttributeValueChange

<b>Purpose</b>	Verify the SOA's ability to receive the numberPoolBlockAttributeValueChange notification for the numberPoolBlockNPAC object.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if the SOA is supporting the numberPoolBlockNPAC managed object instance.
<b>Prerequisites</b>	N/A
<b>Procedure</b>	<ol style="list-style-type: none"> <li>NPAC SMS Simulator issues the M-EVENT-REPORT, numberPoolBlockAttributeValueChange, for a numberPoolBlockNPAC object.</li> <li>SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	SOA responds with a valid M-EVENT-REPORT confirmation.

12.6.2 MOC.SOA.CAP.NOT.numberPoolBlockStatusAttributeValueChange

<b>Purpose</b>	Verify the SOA's ability to receive the numberPoolBlockStatusAttributeValueChange notification for the numberPoolBlockNPAC object.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if the SOA is supporting the numberPoolBlockNPAC managed object instance.
<b>Prerequisites</b>	N/A
<b>Procedure</b>	<ol style="list-style-type: none"> <li>NPAC SMS Simulator issues the M-EVENT-REPORT, numberPoolBlockStatusAttributeValueChange, for a numberPoolBlockNPAC object.</li> <li>SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	SOA responds with a valid M-EVENT-REPORT confirmation.

12.7 serviceProvNPA-NXX-X

<b>MO</b>	serviceProvNPA-NXX-X
-----------	----------------------

<b>Purpose</b>	This section contains test cases for the serviceProvNPA-NXX-X Managed Object Class pertaining to the NPAC SMS manager to SOA Interface, as part of the MO Conformance testing of the interoperability test. This capability test package checks the SOA existence and basic validity of the specified capabilities. This object is used to support network data download to the SOA.
<b>Prerequisite</b>	<ul style="list-style-type: none"> <li>• A NPAC Management association function is established.</li> <li>• The SOA has successfully completed the S2S Interoperability testing.</li> <li>• There is a serviceProvNetwork existing on SOA.</li> </ul>

12.7.1 MOC.NPAC.SOA.CAP.OP.CRE.serviceProvNPA-NXX-X

<b>Purpose</b>	Verify SOA's ability to respond correctly to an M-CREATE request for the serviceProvNPA-NXX-X managed object instance.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if the SOA is to support network data download and the serviceProvNPA-NXX-X managed object.
<b>Prerequisites</b>	A serviceProvNetwork instance exists on the SOA.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a serviceProvNPA-NXX-X M-CREATE request.</li> <li>2. SOA responds successfully to the M-CREATE.</li> </ol>
<b>Expected Results</b>	NPAC SMS Simulator sends a valid M-CREATE request and receives the SOA M-CREATE response indicating successful creation of the serviceProvNetwork.

12.7.2 MOC.NPAC.SOA.CAP.OP.SET.serviceProvNPA-NXX-X

<b>Purpose</b>	Verify the SOA's ability to respond correctly to an M-SET request for the serviceProvNPA-NXX-X managed object instance.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if the SOA is to support network data download and the serviceProvNPA-NXX-X managed object.
<b>Prerequisites</b>	A serviceProvNetwork instance exists on the SOA.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a serviceProvNPA-NXX-X M-SET request.</li> <li>2. SOA responds successfully to the M-SET.</li> </ol>
<b>Expected Results</b>	NPAC SMS Simulator sends a valid M-SET request and receives the SOA M-SET response indicating successful modification of the serviceProvNPA-NXX-X.

12.7.3 MOC.NPAC.SOA.CAP.OP.DEL.serviceProvNPA-NXX-X

<b>Purpose</b>	Verify the SOA's ability to respond correctly to an M-DELETE request for the serviceProvNPA-NXX-X managed object class.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if the SOA supports network data download and the serviceProvNPA-NXX-X managed object.
<b>Prerequisites</b>	A serviceProvNPA-NXX-X exists on the SOA.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an M-DELETE request for the serviceProvNPA-NXX-X managed object.</li> <li>2. SOA responds successfully to the M-DELETE.</li> </ol>

<b>Expected Results</b>	NPAC SMS Simulator sends an M-DELETE request to the SOA and the SOA responds successfully.
-------------------------	--

12.7.4 MOC.NPAC.SOA.INV.CRE.DUP.serviceProvNPA-NXX-X

<b>Purpose</b>	Verify the SOA's ability to respond correctly to a duplicate M-CREATE request for the serviceProvNetwork managed object.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if the SOA is to support network data download and the serviceProvNPA-NXX-X managed object.
<b>Prerequisites</b>	A serviceProvNPA-NXX-X instance exists on the SOA
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a serviceProvNPA-NXX-X M-CREATE request for a serviceProvNetwork managed object that already exists.</li> <li>2. SOA responds to the M-CREATE.</li> </ol>
<b>Expected Results</b>	NPAC SMS Simulator sends an M-CREATE request for an existing serviceProvNPA-NXX-X managed object and receives the SOA M-CREATE error response of duplicateObjectInstanceEr.

12.7.5 MOC.NPAC.SOA.INV.SET.serviceProvNPA-NXX-X

<b>Purpose</b>	Verify the SOA's ability to respond to an M-SET for a syntactically invalid CMIP request for the read-only attribute serviceProvNPA-NXX-X-ID.
<b>Severity</b>	O
<b>Severity Explanation</b>	Required if the SOA is to support network data download and the serviceProvNPA-NXX-X managed object.
<b>Prerequisites</b>	A serviceProvNPA-NXX-X exists on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an M-SET request for a serviceProvNPA-NXX-X-ID.</li> <li>2. SOA responds with an M-SET error.</li> </ol>
<b>Expected Results</b>	NPAC SMS Simulator receives an error response with the error type set to setListErrorEr.

12.7.6 MOC.NPAC.SOA.INV.DEL.serviceProvNPA-NXX-X

<b>Purpose</b>	Verify the SOA correctly handles an invalid CMIP M-DELETE request. NPAC SMS Simulator sends a delete for the serviceProvNPA-NXX-X managed object that does not exist on the SOA.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if the SOA is to support network data download and the serviceProvNPA-NXX-X managed object.
<b>Prerequisites</b>	The serviceProvNPA-NXX-X managed object to be deleted does not exist on the SOA.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends M-DELETE request for the serviceProvNPA-NXX-X managed object.</li> <li>2. SOA responds with an M-DELETE error.</li> </ol>
<b>Expected Results</b>	NPAC SMS Simulator receives the error response with error type set to noSuchObjectInstanceEr.



## 13 LSMS to NPAC MOC Test Cases

### 13.1 InpNPAC-SMS

<b>MO</b>	InpNPAC-SMS
<b>Purpose</b>	This section contains the test cases for the InpNPAC-SMS Managed Object Class pertaining to the LSMS to NPAC SMS Interface, as part of the Managed Object Conformance testing of the NPAC SMS Interoperability Test.
<b>Prerequisite</b>	A Service Provider and Network Data Management association function is established. A InpNPAC-SMS Managed Object Instance has been created.

#### 13.1.1 MOC.LSMS.CAP.OP.GET.InpNPAC-SMS

<b>Purpose</b>	To test the LSMS's ability to GET all the attributes of the InpNPAC-SMS managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	LSMS does not need to issue this request to provide LNP service.
<b>Prerequisites</b>	A InpNPAC-SMS managed object instance has been inherently created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends the M-GET request for the InpNPAC-SMS object requesting all attributes.</li> <li>2. NPAC SMS Simulator responds with the M-GET result containing all attributes.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-GET request and retrieves the attributes successfully from the NPAC SMS Simulator.

#### 13.1.2 MOC.LSMS.CAP.OP.ACT.InpRecoveryComplete

<b>Purpose</b>	To test the LSMS's ability to indicate that the recovery mode for the Local SMS is complete.
<b>Severity</b>	R
<b>Severity Explanation</b>	This action is needed to allow the LSMS to download subscription version and network data, which will be used to provide LNP service.
<b>Prerequisites</b>	A InpNPAC-SMS managed object instance has been inherently created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends the InpRecoveryComplete M-ACTION request.</li> <li>2. NPAC SMS Simulator responds with a successful M-ACTION result.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-ACTION request and receives the NPAC SMS Simulator's M-ACTION response properly.

#### 13.1.3 MOC.LSMS.CAP.NOT.InpNPAC-SMS-Operational-Information

<b>Purpose</b>	To test the LSMS's ability to receive the M-EVENT-REPORT for the InpNPAC-SMS-Operational-Information notification.
<b>Severity</b>	R

<b>Severity Explanation</b>	LSMS is required to handle this notification which informs it of NPAC down time.
<b>Prerequisites</b>	A InpNPAC-SMS managed object instance has been inherently created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the InpNPAC-SMS-Operational-Information M-EVENT-REPORT.</li> <li>2. LSMS confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The LSMS responds with a valid M-EVENT-REPORT confirmation.

#### 13.1.4 MOC.LSMS.INV.GET.InpNPAC-SMS

<b>Purpose</b>	To test the LSMS's ability to handle the M-GET error response processingFailure error to a previously initiated and valid M-GET request for all attributes of the InpNPAC-SMS object.
<b>Severity</b>	O
<b>Severity Explanation</b>	LSMS does not need to issue this request to provide LNP service.
<b>Prerequisites</b>	MOC.LSMS.CAP.OP.GET.InpNPAC-SMS
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends the valid M-GET request for all attributes of the InpNPAC-SMS object.</li> <li>2. NPAC SMS Simulator responds with a processingFailure error.</li> </ol>
<b>Expected Results</b>	The LSMS correctly handles the error response processingFailure error from the NPAC SMS Simulator.

#### 13.1.5 MOC.LSMS.INV.ACT.InpRecoveryComplete

<b>Purpose</b>	To test the LSMS's ability to handle an error response for an M-ACTION request. This will be accomplished by returning the noSuchAction error in response to the InpRecoveryComplete action.
<b>Severity</b>	R
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.LSMS.CAP.ACT.InpNPAC-SMS
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends the valid InpRecoveryComplete M-ACTION request to the InpNPAC-SMS object.</li> <li>2. NPAC SMS Simulator responds with a noSuchAction error.</li> </ol>
<b>Expected Results</b>	The LSMS will correctly handle the error response received from the NPAC SMS Simulator.

#### 13.1.6 MOC.LSMS.INV.NOT.InpNPAC-SMS-Operational-Information

<b>Purpose</b>	To test the LSMS's ability to handle an invalid M-EVENT-REPORT for the InpNPAC-SMS-Operational-Information notification. This will be accomplished by setting the stop time attribute of that notification to a value that is before the start time.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• MOC.LSMS.CAP.NOT.InpNPAC-SMS-Operational-Information</li> <li>• A InpNPAC-SMS managed object instance has been inherently</li> </ul>

	created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the InpNPAC-SMS-Operational-Information M-EVENT-REPORT.</li> <li>2. LSMS responds with an invalidArgumentValue error.</li> </ol>
<b>Expected Results</b>	The LSMS will correctly handle the invalid M-EVENT-REPORT received from the NPAC SMS Simulator and return the invalidArgumentValue error.

### 13.1.7 MOC.LSMS.CAP.NOT.subscriptionVersionNewNPA-NXX

<b>Purpose</b>	To test the LSMS's ability to receive the M-EVENT-REPORT for subscriptionVersionNewNPA-NXX notification.
<b>Severity</b>	R
<b>Severity Explanation</b>	.
<b>Prerequisites</b>	A InpNPAC-SMS managed object instance has been inherently created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionNewNPA-NXX M-EVENT-REPORT.</li> <li>2. LSMS confirms M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The LSMS responds with a valid M-EVENT-REPORT confirmation.

### 13.1.8 MOC.LSMS.INV.NOT.subscriptionVersionNewNPA-NXX

<b>Purpose</b>	To test the LSMS's ability to handle an invalid M-EVENT-REPORT for the subscriptionVersionNewNPA-NXX notification. This will be accomplished by sending the subscriptionVersionNewNPA-NXX notification with an invalid NPA-NXX value.
<b>Severity</b>	O
<b>Severity Explanation</b>	Optional.
<b>Prerequisites</b>	A InpNPAC-SMS managed object instance has been inherently created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues the subscriptionVersionNewNPA-NXX M-EVENT-REPORT.</li> <li>2. LSMS returns an invalidArgumentValue error.</li> </ol>
<b>Expected Results</b>	The LSMS will correctly handle the invalid M-EVENT-REPORT received from the NPAC SMS Simulator and return appropriate error.

### 13.1.9 MOC.LSMS.CAP.ACT.InpNotificationRecovery

<b>Purpose</b>	Verify LSMS can successfully process the InpNotificationRecovery action.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the service provider LSMS supports notification recovery.
<b>Prerequisites</b>	Notifications exist of each type of notification that can be recovered for the requesting service provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends the InpNotificationRecovery action to the NPAC SMS Simulator to start notification data download for a</li> </ol>



	specified period of time. 2. NPAC SMS Simulator responds with M-ACTION InpNotificationRecovery response.
<b>Expected Results</b>	LSMS sends the M-ACTION and receives the action response with notification data.

13.1.10 MOC.LSMS.INV.ACT.InpNotificationRecovery

<b>Purpose</b>	Verify LSMS can successfully process an error response to the InpNotificationRecovery action.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the service provider LSMS supports notification recovery.
<b>Prerequisites</b>	
<b>Procedure</b>	1. LSMS sends action request to NPAC SMS Simulator to start notification data download for a specified period of time. 2. NPAC SMS Simulator responds with error status 'failed'.
<b>Expected Results</b>	LSMS sends the M-ACTION and receives the action response with the error successfully.

13.2 InpServiceProvs

<b>MO</b>	InpServiceProvs
<b>Purpose</b>	This section contains the test cases for the InpServiceProvs Managed Object Class pertaining to the LSMS to NPAC SMS Interface, as part of the Managed Object Conformance testing of the NPAC SMS Interoperability Test.
<b>Prerequisite</b>	A Service Provider and Network Data Management association function is established. A InpNPAC-SMS and a InpServiceProvs Managed Object Instances have been created inherently.

13.2.1 MOC.LSMS.CAP.OP.GET.InpServiceProvs

<b>Purpose</b>	To test the LSMS's ability to GET all the attributes of the InpServiceProvs object.
<b>Severity</b>	O
<b>Severity Explanation</b>	No impact to providing LNP service. LSMS may perform to verify InpServiceProvs object.
<b>Prerequisites</b>	A InpServiceProvs managed object instance has been inherently created.
<b>Procedure</b>	1. LSMS sends a valid M-GET request to get all attributes of the InpServiceProvs object. 2. NPAC SMS Simulator responds with a valid M-GET result containing the attributes.
<b>Expected Results</b>	The LSMS issues a valid M-GET request and retrieves the attributes successfully from the NPAC SMS Simulator.

13.2.2 MOC.LSMS.INV.GET.InpServiceProvs

<b>Purpose</b>	To test the LSMS's ability to handle the M-GET error response
----------------	---

	processingFailure error to a previously initiated and valid M-GET request for all attributes of the InpServiceProvs object.
<b>Severity</b>	O
<b>Severity Explanation</b>	No impact to providing LNP service. LSMS may perform to verify error handling.
<b>Prerequisites</b>	MOC.LSMS.CAP.OP.GET.InpServiceProvs
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-GET request to get all the attributes of the InpServiceProvs object.</li> <li>2. NPAC SMS Simulator responds with a processingFailure error.</li> </ol>
<b>Expected Results</b>	The LSMS correctly handles the error response processingFailure error from the NPAC SMS Simulator.

### 13.3 InpSubscriptions

<b>MO</b>	InpSubscriptions
<b>Purpose</b>	This section contains the test cases for the InpSubscriptions Managed Object Class pertaining to the LSMS to NPAC SMS Interface, as part of the Managed Object Conformance testing of the NPAC SMS Interoperability Test.
<b>Prerequisite</b>	A LSMS Network and Subscription Data Download Management association function is established. A InpNPAC-SMS and InpSubscriptions Managed Object Instances have been created inherently.

#### 13.3.1 MOC.LSMS.CAP.OP.GET.InpSubscriptions

<b>Purpose</b>	To test the LSMS's ability to GET all the attributes of the InpSubscriptions managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. LSMS may perform to validate InpSubscriptions object.
<b>Prerequisites</b>	A InpSubscriptions managed object instance has been inherently created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a validate M-GET request to retrieve all attributes of the InpSubscriptions object.</li> <li>2. NPAC SMS Simulator responds with a successful M-GET result containing all the attributes.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-GET request, retrieves the attributes successfully from the NPAC SMS Simulator and correctly handles the response.

#### 13.3.2 MOC.LSMS.CAP.ACT.InpSubscriptions.InpDownload

<b>Purpose</b>	To test the LSMS's ability to download the subscriptionVersionNPAC objects instantiated on the NPAC SMS Simulator.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service.
<b>Prerequisites</b>	A InpSubscriptions managed object instance has been inherently created.

<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a validate InpDownload M-ACTION request.</li> <li>2. NPAC SMS Simulator responds with a successful M-ACTION response.</li> </ol>
<b>Expected Results</b>	The LSMS sends a valid M-ACTION request, and receives the NPAC SMS Simulator's M-ACTION response properly.

### 13.3.3 MOC.LSMS.INV.GET.InpSubscriptions

<b>Purpose</b>	To test the LSMS's ability to handle the M-GET error response getListError error to a previously initiated and valid M-GET request for all the attributes of the InpSubscriptions object.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. LSMS may perform to validate InpSubscriptions object.
<b>Prerequisites</b>	MOC.LSMS.CAP.OP.GET.InpSubscriptions
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-GET request to retrieve all the attributes of the InpSubscriptions object.</li> <li>2. NPAC SMS Simulator responds with a getListError error.</li> </ol>
<b>Expected Results</b>	The LSMS correctly handles the error response getListError error from the NPAC SMS Simulator.

### 13.3.4 MOC.LSMS.INV.ACT.InpSubscriptions

<b>Purpose</b>	To test the LSMS's ability to handle an error response for an M-ACTION request. This will be accomplished by returning the complexityLimitation error in response to the InpDownload action.
<b>Severity</b>	R
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.LSMS.CAP.ACT.InpSubscriptions.InpDownload
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a validate InpDownload M-ACTION request.</li> <li>2. NPAC SMS Simulator responds with complexityLimitation error.</li> </ol>
<b>Expected Results</b>	The LSMS will correctly handle the complexityLimitation error response received from the NPAC SMS Simulator.

### 13.3.5 MOC.LSMS.VAL.InpDownload-NumberPoolBlock

<b>Purpose</b>	Verify the LSMS's ability to issue the InpDownload action for numberPoolBlock data.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if LSMS will be supporting numberPoolBlock data.
<b>Prerequisites</b>	NumberPoolBlock objects exist on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS issues a valid InpDownload M-ACTION request for all or specific numberPoolBlock objects.</li> <li>2. NPAC SMS Simulator responds with a successful M-ACTION response containing the requested data.</li> </ol>
<b>Expected Results</b>	LSMS issues a valid M-ACTION request and retrieves the data successfully from the NPAC SMS Simulator.

### 13.4 *InpNetwork*

<b>MO</b>	InpNetwork
<b>Purpose</b>	This section contains the test cases for the InpNetwork Managed Object Class pertaining to the LSMS to NPAC SMS Interface, as part of the Managed Object Conformance testing of the NPAC SMS Interoperability Test.
<b>Prerequisite</b>	A LSMS Network and Subscription Data Download Management association function is established. A InpNPAC-SMS and a InpNetwork Managed Object Instances have been created inherently.

#### 13.4.1 MOC.LSMS.CAP.OP.GET.InpNetwork

<b>Purpose</b>	To test the LSMS's ability to GET all the attributes of the InpNetwork managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	No impact to providing LNP service. LSMS may perform to validate InpNetwork object.
<b>Prerequisites</b>	A InpNetwork managed object instance has been inherently created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-GET request for all attribute of the InpNetwork object.</li> <li>2. NPAC SMS Simulator responds with a successful M-GET result containing all the attributes.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-GET request and retrieves the attributes successfully from the NPAC SMS Simulator.

#### 13.4.2 MOC.LSMS.CAP.ACT.InpNetwork.InpDownload

<b>Purpose</b>	To test the LSMS's ability to download the serviceProvNetwork, serviceProvNPA-NXX and serviceProvLRN objects instantiated on the NPAC SMS Simulator.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service.
<b>Prerequisites</b>	A InpNetwork managed object instance has been inherently created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid InpDownload M-ACTION request.</li> <li>2. NPAC SMS Simulator responds with a successful M-ACTION reply.</li> </ol>
<b>Expected Results</b>	The LSMS sends a valid M-ACTION request, and receives the NPAC SMS Simulator's M-ACTION response properly.

#### 13.4.3 MOC.LSMS.INV.GET.InpNetwork

<b>Purpose</b>	To test the LSMS's ability to handle the M-GET error response processingFailure error to a previously initiated and valid M-GET request for all attributes of the InpNetwork object.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.LSMS.VAL.GET.InpNetwork
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-GET request for all the attributes of the InpNetwork object.</li> </ol>

	2. NPAC SMS Simulator responds with a processingFailure error.
<b>Expected Results</b>	The LSMS correctly handles the error response processingFailure error from the NPAC SMS Simulator.

#### 13.4.4 MOC.LSMS.INV.ACT.InpNetwork

<b>Purpose</b>	To test the LSMS's ability to handle an error response for an M-ACTION request. This will be accomplished by returning the processingFailure error in response to the InpDownload action.
<b>Severity</b>	R
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.LSMS.CAP.ACT.InpNetwork.InpDownload
<b>Procedure</b>	1. LSMS sends a valid InpDownload M-ACTION request. 2. NPAC SMS Simulator responds with a processingFailure error.
<b>Expected Results</b>	The LSMS will correctly handle the error response received from the NPAC SMS Simulator.

#### 13.4.5 MOC.LSMS.VAL.InpDownload-NPA-NXX-X

<b>Purpose</b>	Verify the LSMS's ability to issue the InpDownload action for serviceProvNPA-NXX-X data.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if LSMS will be supporting serviceProvNPA-NXX-X data.
<b>Prerequisites</b>	serviceProvNPA-NXX-X objects exist on the NPAC SMS Simulator.
<b>Procedure</b>	1. LSMS issues a valid InpDownload M-ACTION request for all network data or specific serviceProvNPA-NXX-X objects. 2. NPAC SMS Simulator responds with a successful M-ACTION response containing the requested data.
<b>Expected Results</b>	LSMS issues a valid M-ACTION request and retrieves the data successfully from the NPAC SMS Simulator.

### 13.5 *serviceProv*

<b>MO</b>	serviceProv
<b>Purpose</b>	This section contains the test cases for the serviceProv Managed Object Class pertaining to the LSMS to NPAC SMS Interface, as part of the Managed Object Conformance testing of the NPAC SMS Interoperability Test.
<b>Prerequisite</b>	A Service Provider and Network Data Management is established. A InpNPAC-SMS and a InpServiceProvs Managed Object Instances have been created inherently. A serviceProv Managed Object Instance has been created locally by the NPAC SMS Simulator personnel.

#### 13.5.1 MOC.LSMS.CAP.OP.SET.serviceProv

<b>Purpose</b>	To test the LSMS's ability to SET all of the mandatory attributes on which the M-SET operation is allowed in the serviceProv managed object instance.
----------------	---

<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if LSMS will be used to manage service provider profile.
<b>Prerequisites</b>	A serviceProv managed object instance has been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-SET request for the serviceProvAddress, serviceProvSysLinkInfo, and serviceProvTunables attributes.</li> <li>2. NPAC SMS Simulator responds with a successful M-SET response.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-SET request and sets the serviceProvAddress, serviceProvSysLinkInfo, and serviceProvTunables attributes successfully in the NPAC SMS Simulator.

13.5.2 MOC.LSMS.CAP.OP.GET.serviceProv

<b>Purpose</b>	To test the LSMS's ability to GET all the attributes of the serviceProv managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	MOC.LSMS.CAP.OP.SET.serviceProv
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-GET request for all attributes.</li> <li>2. NPAC SMS Simulator responds with a successful M-GET result containing all attributes.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-GET request and retrieves the attributes successfully from the NPAC SMS Simulator.

13.5.3 MOC.LSMS.VAL.SET.SING.serviceProv

<b>Purpose</b>	To test the LSMS's ability to SET a single attribute, namely the serviceProvAddress in the serviceProv managed object instance.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if the service provider will be managing their profile from the LSMS. Requirement exists but may be satisfied by MOC.LSMS.CAP.OP.SET.serviceProv
<b>Prerequisites</b>	A serviceProv managed object instance has been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-SET request for the serviceProvAddress attribute.</li> <li>2. NPAC SMS Simulator responds with a successful M-SET response.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-SET request and sets the serviceProvAddress attribute successfully in the NPAC SMS Simulator.

13.5.4 MOC.LSMS.VAL.SET.SING.COND.serviceProv

<b>Purpose</b>	To test the LSMS's ability to SET a single conditional attribute, namely the serviceProvBillingAddress in the serviceProv managed
----------------	---

	object instance.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if the service provider will be managing their profile from the LSMS. Requirement exists but may be satisfied by MOC.LSMS.CAP.OP.SET.serviceProv
<b>Prerequisites</b>	MOC.LSMS.VAL.SET.SING.serviceProv
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-SET request for the serviceProvBillingAddress attribute.</li> <li>2. NPAC SMS Simulator responds with a successful M-SET response.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-SET request and sets the serviceProvBillingAddress attribute successfully in the NPAC SMS Simulator.

### 13.5.5 MOC.LSMS.VAL.SET.MULT.serviceProv

<b>Purpose</b>	To test the LSMS's ability to SET a group of attributes, namely the serviceProvAddress, serviceProvTunables, and serviceProvLSMS-Address in the serviceProv managed object instance.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service Requirement exists but may be satisfied by MOC.LSMS.CAP.OP.SET.serviceProv
<b>Prerequisites</b>	MOC.LSMS.VAL.SET.SING.COND.serviceProv
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-SET request for the serviceProvAddress, serviceProvTunables, and serviceProvLSMS-Address attributes.</li> <li>2. NPAC SMS Simulator responds with a successful M-SET response.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-SET request and sets the attribute group successfully in the NPAC SMS Simulator.

### 13.5.6 MOC.LSMS.INV.SET.serviceProv

<b>Purpose</b>	To test the LSMS's ability to handle the M-SET error response setListError error to a previously initiated and valid M-SET request for the serviceProvName attribute.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. May be performed by LSMS to verify error handling.
<b>Prerequisites</b>	MOC.LSMS.VAL.SET.SING.serviceProv
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-SET request for the serviceProvName, attribute.</li> <li>2. NPAC SMS Simulator responds with a setListError error.</li> </ol>
<b>Expected Results</b>	The LSMS correctly handles the error response setListError error from the NPAC SMS Simulator.

### 13.5.7 MOC.LSMS.INV.GET.serviceProv

<b>Purpose</b>	To test the LSMS's ability to handle the M-GET error response nosuchObjectInstance error to a previously initiated and valid M-GET request all the attributes of the serviceProv object.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. May be performed by LSMS to verify error handling.
<b>Prerequisites</b>	MOC.LSMS.CAP.OP.GET.serviceProv
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-GET request for all the attributes of the serviceProv object.</li> <li>2. NPAC SMS Simulator responds with a noSuchObjectInstance error.</li> </ol>
<b>Expected Results</b>	The LSMS correctly handles the error response noSuchObjectInstance error from the NPAC SMS Simulator.

### 13.5.8 MOC.LSMS.BND.MIN.SET.serviceProv

<b>Purpose</b>	To test the behaviour of the LSMS when setting the city field of the serviceProvAddress attribute to a value of length 1 octet which is the lower bound of the range for the city size.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. LSMS may perform to validate boundary conditions.
<b>Prerequisites</b>	MOC.LSMS.VAL.SET.SING.serviceProv
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-SET request for the serviceProvAddress city attribute set to a string value of length 1.</li> <li>2. NPAC SMS Simulator responds with a successful M-SET response.</li> </ol>
<b>Expected Results</b>	The city field is set accordingly in the NPAC SMS Simulator.

### 13.5.9 MOC.LSMS.BND.MAX.SET.serviceProv

<b>Purpose</b>	To test the behaviour of the LSMS when setting the city field of the serviceProvAddress attribute to a value of length 20 which is the higher bound of the range for the city size.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. LSMS may perform to verify boundary conditions.
<b>Prerequisites</b>	MOC.LSMS.VAL.SET.SING.serviceProv
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-SET request for the serviceProvAddress city attribute set to a string value of length 20.</li> <li>2. NPAC SMS Simulator responds with a successful M-SET response.</li> </ol>
<b>Expected Results</b>	The city field is set accordingly in the NPAC SMS Simulator.



### 13.6 *IsmsFilterNPA-NXX*

<b>MO</b>	IsmsFilterNPA-NXX
<b>Purpose</b>	This section contains the test cases for the IsmsFilterNPA-NXX Managed Object Class pertaining to the LSMS to NPAC SMS Interface, as part of the Managed Object Conformance testing of the NPAC SMS Interoperability Test.
<b>Prerequisite</b>	A Service Provider and Network Data Management association function is established. A InpNPAC-SMS and a InpServiceProvs Managed Object Instances have been created inherently.

#### 13.6.1 MOC.LSMS.CAP.OP.CRE.IsmsFilterNPA-NXX

<b>Purpose</b>	To test the LSMS's ability to CREATE a IsmsFilterNPA-NXX managed object instance in the NPAC SMS Simulator.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if the service provider will be managing IsmsfilterNPA-NXX objects.
<b>Prerequisites</b>	All the test cases for the serviceProv Managed Object Class
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-CREATE request for the IsmsFilterNPA-NXX object</li> <li>2. NPAC SMS Simulator responds with a successful M-CREATE response.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-CREATE request causing the IsmsFilterNPA-NXX instance to be created and its attributes populated successfully in the NPAC SMS Simulator.

#### 13.6.2 MOC.LSMS.CAP.OP.GET.IsmsFilterNPA-NXX

<b>Purpose</b>	To test the LSMS's ability to GET all the attributes of the IsmsFilterNPA-NXX managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. May be formed by the LSMS to verify their ability to retrieve the instance.
<b>Prerequisites</b>	MOC.LSMS.CAP.OP.CRE.IsmsFilterNPA-NXX
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-GET request for the IsmsFilterNPA-NXX object</li> <li>2. NPAC SMS Simulator responds with a successful M-GET result containing all the attributes.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-GET request and retrieves the attributes successfully from the NPAC SMS Simulator.

#### 13.6.3 MOC.LSMS.CAP.OP.DEL.IsmsFilterNPA-NXX

<b>Purpose</b>	To test the LSMS's ability to DELETE an existing IsmsFilterNPA-NXX managed object instance from the NPAC SMS Simulator.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if the service provider will be managing the IsmsFilterNPA-NXX objects.
<b>Prerequisites</b>	MOC.LSMS.CAP.OP.CRE.IsmsFilterNPA-NXX

<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-DELETE request for the lsmsFilterNPA-NXX object</li> <li>2. NPAC SMS Simulator responds with a successful M-DELETE response containing all the attributes.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-DELETE request and removes the managed object instance successfully from the NPAC SMS Simulator.

13.6.4 MOC.LSMS.VAL.CRE.AUTO.lsmsFilterNPA-NXX

<b>Purpose</b>	To test the LSMS's ability to CREATE a lsmsFilterNPA-NXX managed object instance in the NPAC SMS Simulator using automatic instance naming.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if the service provider will be managing the lsmsFilterNPA-NXX objects.
<b>Prerequisites</b>	MOC.LSMS.CAP.OP.CRE.lsmsFilterNPA-NXX
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-CREATE request for the lsmsFilterNPA-NXX object</li> <li>2. NPAC SMS Simulator responds with a successful M-CREATE response.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-CREATE request with automatic instance naming causing the lsmsFilterNPA-NXX instance to be created and its attributes populated successfully in the NPAC SMS Simulator.

13.6.5 MOC.LSMS.VAL.GET.SCOP.FILT.lsmsFilterNPA-NXX

<b>Purpose</b>	To test the LSMS's ability to initiate a valid scoped and filtered M-GET request for all the attributes of a lsmsFilterNPA-NXX object. This will be accomplished by retrieving all the attributes within an agreed upon lsmsFilterNPA-NXX-Value range, starting at the base managed object serviceProv and ending at the lsmsFilterNPA-NXX.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	lsmsFilterNPA-NXX managed object instances have been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-GET request for the lsmsFilterNPA-NXXs attributes with a filter to retrieve only those in the specified lsmsFilterNPA-NXX-Value range.</li> <li>2. NPAC SMS Simulator responds with a successful M-GET result containing the attribute.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-GET request and retrieves the attribute successfully from the NPAC SMS Simulator.

13.6.6 MOC.LSMS.VAL.DEL.SCOP.FILT.lsmsFilterNPA-NXX

<b>Purpose</b>	To test the LSMS's ability to initiate a valid scoped and filtered M-DELETE request for an existing managed object instance. This will be accomplished by deleting all the lsmsFilterNPA-NXX instances with the lsmsFilterNPA-NXX-Value equal to an agreed upon value,
----------------	--

	starting at the base managed object serviceProv and ending at the lsmsFilterNPA-NXX.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. May be used to satisfy the requirements instead of Test Case MOC.LSMS.CAP.OP.DEL.lsmsFilterNPA-NXX.
<b>Prerequisites</b>	lsmsFilterNPA-NXX managed object instance with above lsmsFilterNPA-NXX-Value attribute have been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-DELETE request for the lsmsFilterNPA-NXX object with a filter to delete only that which is equal to a specified lsmsFilterNPA-NXX-Value.</li> <li>2. NPAC SMS Simulator responds with a successful M-GET result containing the attribute.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-DELETE request with a correct scope and filter causing the above instance to be removed successfully from the NPAC SMS Simulator.

13.6.7 MOC.LSMS.INV.CRE.lsmsFilterNPA-NXX

<b>Purpose</b>	To test the LSMS's ability to handle the M-CREATE error response duplicateManagedObjectInstance error to a previously initiated and valid M-CREATE request for the lsmsFilterNPA-NXX MO.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if the service provider will be managing the lsmsFilterNPA-NXX objects.
<b>Prerequisites</b>	MOC.LSMS.CAP.OP.CRE.lsmsFilterNPA-NXX
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-CREATE request for the lsmsFilterNPA-NXX object.</li> <li>2. NPAC SMS Simulator responds with a duplicateManagedObjectInstance error.</li> </ol>
<b>Expected Results</b>	The LSMS correctly handles the error response duplicateManagedObjectInstance error from the NPAC SMS Simulator.

13.6.8 MOC.LSMS.INV.GET.lsmsFilterNPA-NXX

<b>Purpose</b>	To test the LSMS's ability to handle the M-GET error response operationCancelled error to a previously initiated and valid M-GET request for all the attributes of the lsmsFilterNPA-NXX object.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if the service provider will be managing the lsmsFilterNPA-NXX objects.
<b>Prerequisites</b>	MOC.LSMS.CAP.OP.GET.lsmsFilterNPA-NXX
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-GET request for all the attributes of a lsmsFilterNPA-NXX object.</li> <li>2. NPAC SMS Simulator responds with an operationCancelled error.</li> </ol>
<b>Expected Results</b>	The LSMS correctly handles the error response operationCancelled error from the NPAC SMS Simulator.

13.6.9 MOC.LSMS.INV.DEL.IsmsFilterNPA-NXX

<b>Purpose</b>	To test the LSMS's ability to handle the M-DELETE error response processingFailure error to a previously initiated and valid M-DELETE request for an existing IsmsFilterNPA-NXX managed object instance.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if the service provider will be managing the IsmsFilterNPA-NXX objects.
<b>Prerequisites</b>	MOC.LSMS.CAP.OP.DEL.IsmsFilterNPA-NXX
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-DELETE request for the IsmsFilterNPA-NXX object.</li> <li>2. NPAC SMS Simulator responds with a processingFailure error.</li> </ol>
<b>Expected Results</b>	The LSMS correctly handles the error response processingFailure error from the NPAC SMS Simulator.

13.7 subscriptionVersionNPAC

<b>MO</b>	subscriptionVersionNPAC
<b>Purpose</b>	This section contains the test cases for the subscriptionVersionNPAC Managed Object Class pertaining to the LSMS to NPAC SMS Interface, as part of the Managed Object Conformance testing of the NPAC SMS Interoperability Test.
<b>Prerequisite</b>	A LSMS Network and Subscription Data Download association function is established. A InpNPAC-SMS and InpSubscriptions Managed Object Instances have been created inherently. All the InpSubscriptions test cases have been performed.

13.7.1 MOC.LSMS.CAP.OP.GET.subscriptionVersionNPAC

<b>Purpose</b>	To test the LSMS's ability to GET all the attributes of the subscriptionVersionNPAC managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact providing LNP service.
<b>Prerequisites</b>	A subscriptionVersionNPAC managed object instance has been created on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-GET request for the IsmsFilterNPA-NXX object.</li> <li>2. NPAC SMS Simulator responds with a successful M-GET result containing all the attributes.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-GET request and retrieves the attributes successfully from the NPAC SMS Simulator.

13.7.2 MOC.LSMS.CAP.NOT.subscriptionVersionNewNPA-NXX

<b>Purpose</b>	To test the LSMS's ability to receive the M-EVENT-REPORT for the subscriptionVersionNPAC subscriptionVersionNewNPA-NXX notification.
<b>Severity</b>	R
<b>Severity Explanation</b>	Needed to inform the LSMS of a new NPA-NXX opened porting.

<b>Prerequisites</b>	A subscriptionVersionNPAC managed object instance has been created on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid subscriptionVersionNewNPA-NXX M-EVENT-REPORT.</li> <li>2. LSMS confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The LSMS responds with a valid M-EVENT-REPORT confirmation.

### 13.7.3 MOC.LSMS.VAL.GET.SCOPE.subscriptionVersionNPAC

<b>Purpose</b>	To test the LSMS's ability to initiate a valid scoped M-GET request for all attributes of a subscriptionVersionNPAC object. This will be accomplished by retrieving all the attributes starting at the base managed object InpSubscriptions and ending at the subscriptionVersionNPAC with filtering on an agreed upon TN range.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact providing LNP service.
<b>Prerequisites</b>	Multiple subscriptionVersionNPAC managed object instances have been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-GET request for the attributes with the filter set for the specified TN-range.</li> <li>2. NPAC SMS Simulator responds with the M-GET results containing the attribute.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-GET request and retrieves the attributes successfully from the NPAC SMS Simulator.

### 13.7.4 MOC.LSMS.INV.GET.subscriptionVersionNPAC

<b>Purpose</b>	To test the LSMS's ability to handle the M-GET error response noSuchObjectInstance error to a previously initiated and valid M-GET request for all the attributes of a subscriptionVersionNPAC object.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact providing LNP service.
<b>Prerequisites</b>	MOC.LSMS.CAP.OP.GET.subscriptionVersionNPAC
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-GET request for all the attributes of a subscriptionVersionNPAC object.</li> <li>2. NPAC SMS Simulator responds with a noSuchObjectInstance error.</li> </ol>
<b>Expected Results</b>	The LSMS correctly handles the error response noSuchObjectInstance error from the NPAC SMS Simulator.

### 13.7.5 MOC.LSMS.INV.NOT.subscriptionVersionNPAC

<b>Purpose</b>	To test the LSMS's ability to handle the M-EVENT-REPORT for the subscriptionVersionNPAC's subscriptionVersionNewNPA-NXX notification with an invalid syntax for the NPA-NXX Value attribute.
----------------	--

<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact providing LNP service.
<b>Prerequisites</b>	MOC.LSMS.CAP.NOT.subscriptionVersionNewNPA-NXX
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid subscriptionVersionNewNPA-NXX M-EVENT-REPORT.</li> <li>2. LSMS responds with invalidArgumentValue or other appropriate error response.</li> </ol>
<b>Expected Results</b>	The LSMS rejects the M-EVENT-REPORT with invalid syntax.

13.7.6 MOC.LSMS.BND.GET.MAXQ.subscriptionVersionNPAC

<b>Purpose</b>	To test the behaviour of the LSMS when it receives the responses to a valid scoped M-GET, which will return the maximum number of records specified in the NPAC SMS Simulator <Max Subscriber Query> parameter. This will be accomplished by requesting any single attribute for all the existing subscriptionVersionNPAC managed object instances.
<b>Severity</b>	C
<b>Severity Explanation</b>	Must be performed if scope and filtered M-GETs are being used.
<b>Prerequisites</b>	The number of subscriptionVersionNPAC managed object instances created is equal to the Max Subscriber Query parameter.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-GET request for the attribute with a filter specified that requires Max Subscriber Query subscription versions to be returned.</li> <li>2. NPAC SMS Simulator with the linked replies for all the subscription versions.</li> </ol>
<b>Expected Results</b>	The LSMS handles the linked replies properly.

13.7.7 MOC.LSMS.INV.QUERY.SCOPE subscriptionVersion

<b>Purpose</b>	To verify a LSMS can handle a scoped filtered query request error.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case should be executed if the LSMS will be supporting scoped filtered subscription version query.
<b>Prerequisites</b>	subscriptionVersionNPACs exist on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The LSMS issues a scoped filtered M-GET for a range of subscription versions where the number of subscription versions that satisfy the request exceeds the maximum number of subscription versions that can be retrieved in one request.</li> <li>2. The NPAC SMS Simulator responds with an M-GET error of complexityLimitation.</li> </ol>
<b>Expected Results</b>	The LSMS successfully initiates the M-GET and successfully handles the M-GET error response.

13.8 serviceProvNetwork

<b>MO</b>	serviceProvNetwork
<b>Purpose</b>	This section contains the test cases for the serviceProvNetwork Managed Object Class pertaining to the LSMS to NPAC SMS Interface, as part of the Managed Object Conformance testing of the NPAC SMS Interoperability

	Test.
<b>Prerequisite</b>	A Service Provider and Network Data Management association function is established. A InpNPAC-SMS and a InpNetwork Managed Object Instances have been created inherently. A serviceProvNetwork Managed Object Instance has been created locally by the NPAC SMS Simulator personnel.

### 13.8.1 MOC.LSMS.CAP.OP.GET.serviceProvNetwork

<b>Purpose</b>	To test the LSMS's ability to GET all the attributes of the serviceProvNetwork managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	MOC.LSMS.CAP.OP.SET.serviceProvNetwork
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-GET request all attributes of the serviceProvNetwork object.</li> <li>2. NPAC SMS Simulator with the M-GET result containing all attributes.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-GET request and retrieves all the attributes (i.e. serviceProvId and serviceProvName) successfully from the NPAC SMS Simulator.

### 13.8.2 MOC.LSMS.INV.GET.serviceProvNetwork

<b>Purpose</b>	To test the LSMS's ability to handle the M-GET error response processingFailure error to a previously initiated and valid M-GET request for all the attributes of the serviceProvNetwork object.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	MOC.LSMS.CAP.OP.GET.serviceProvNetwork
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-GET request for all the attributes of the serviceProvNetwork object.</li> <li>2. NPAC SMS Simulator with a processingFailure error.</li> </ol>
<b>Expected Results</b>	The LSMS correctly handles the error response processingFailure error from the NPAC SMS Simulator.

## 13.9 serviceProvNPA-NXX

<b>MO</b>	serviceProvNPA-NXX
<b>Purpose</b>	This section contains the test cases for the serviceProvNPA-NXX Managed Object Class pertaining to the LSMS to NPAC SMS Interface, as part of the Managed Object Conformance testing of the NPAC SMS Interoperability Test.
<b>Prerequisite</b>	A Service Provider and Network Data Management is established. A InpNPAC-SMS and a InpNetwork Managed Object Instances have been created inherently. All the test cases for the serviceProvNetwork Managed Object Class have been completed.

13.9.1 MOC.LSMS.CAP.OP.GET.serviceProvNPA-NXX

<b>Purpose</b>	To test the LSMS's ability to GET all the attributes of the serviceProvNPA-NXX managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	MOC.LSMS.VAL.CRE.AUTO.serviceProvNPA-NXX
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-GET request for all the attributes of a serviceProvNPA-NXX object.</li> <li>2. NPAC SMS Simulator responds with a successful M-GET result.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-GET request and retrieves the attributes successfully from the NPAC SMS Simulator.

13.9.2 MOC.LSMS.CAP.OP.DEL.serviceProvNPA-NXX

<b>Purpose</b>	To test the LSMS's ability to DELETE an existing serviceProvNPA-NXX managed object instance from the NPAC SMS Simulator.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if the LSMS is managing network data.
<b>Prerequisites</b>	MOC.LSMS.VAL.CRE.AUTO.serviceProvNPA-NXX
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-DELETE request for the serviceProvNPA-NXX object.</li> <li>2. NPAC SMS Simulator responds with a successful M-DELETE response.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-DELETE request and removes the managed object instance successfully from the NPAC SMS Simulator.

13.9.3 MOC.LSMS.VAL.CRE.AUTO.serviceProvNPA-NXX

<b>Purpose</b>	To test the LSMS's ability to CREATE a serviceProvNPA-NXX managed object instance in the NPAC SMS Simulator using automatic instance naming.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if the LSMS is managing network data.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-CREATE request for a serviceProvNPA-NXX object.</li> <li>2. NPAC SMS Simulator responds with a successful M-CREATE response.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-CREATE request with automatic instance naming causing the serviceProvNPA-NXX instance to be created and its attributes populated successfully in the NPAC SMS Simulator.



13.9.4 MOC.LSMS.VAL.GET.SCOPE.FILT.serviceProvNPA-NXX

<b>Purpose</b>	To test the LSMS's ability to initiate a valid scoped and filtered M-GET request for a single attribute. This will be accomplished by retrieving all the attributes for the agreed upon NPA-NXX (i.e. filtering on serviceProvNPA-NXX-Value equal to that number) starting at the base managed object serviceProvNetwork and ending at the serviceProvNPA-NXX.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	A serviceProvNPA-NXX managed object instance with above serviceProvNPA-NXX-Value attribute has been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-GET request for the attributes of the serviceProvNPA-NXX object with a filter on the serviceProvNPA-NXX-Value for the agreed upon value.</li> <li>2. NPAC SMS Simulator responds with a successful M-GET result containing the specified attribute..</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-GET request and retrieves the attribute successfully from the NPAC SMS Simulator.

13.9.5 MOC.LSMS.VAL.DEL.SCOPE.FILT.serviceProvNPA-NXX

<b>Purpose</b>	To test the LSMS's ability to initiate a valid scoped and filtered M-DELETE request for an existing managed object instance. This will be accomplished by deleting all the serviceProvNPA-NXX instances with the serviceProvNPA-NXX-Value equal to an agreed upon NPA-NXX, starting at the base managed object serviceProvNetwork and ending at the serviceProvNPA-NXX.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. This function may be satisfied by deleting instances one at a time. May be used to satisfy the requirements instead of Test Case MOC.LSMS.CAP.OP.DEL.serviceProvNPA-NXX.
<b>Prerequisites</b>	A serviceProvNPA-NXX managed object instance with above serviceProvNPA-NXX-Value attribute has been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-DELETE request for the serviceProvNetwork object with a filter set to the agreed upon serviceProvNPA-NXX-Value.</li> <li>2. NPAC SMS Simulator responds with a successful M-DELETE response.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-DELETE request with a correct scope and filter causing the above instance to be removed successfully from the NPAC SMS Simulator.

13.9.6 MOC.LSMS.INV.CRE.serviceProvNPA-NXX

<b>Purpose</b>	To test the LSMS's ability to handle the M-CREATE error response duplicateManagedObjectInstance error to a previously initiated and valid M-CREATE request for the serviceProvNPA-NXX MO.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if the LSMS is managing network data.

<b>Prerequisites</b>	MOC.LSMS.VAL.CRE.AUTO.serviceProvNPA-NXX
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-CREATE request for the serviceProvNPA-NXX object.</li> <li>2. NPAC SMS Simulator responds with a duplicateManagedObjectInstance error.</li> </ol>
<b>Expected Results</b>	The LSMS correctly handles the error response duplicateManagedObjectInstance error from the NPAC SMS Simulator.

### 13.9.7 MOC.LSMS.INV.GET.serviceProvNPA-NXX

<b>Purpose</b>	To test the LSMS's ability to handle the M-GET error response operationCancelled error to a previously initiated and valid M-GET request for all the attributes of the serviceProvNPA-NXX object.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	MOC.LSMS.CAP.OP.GET.serviceProvNPA-NXX
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-GET request for all the attributes of a serviceProvNPA-NXX object.</li> <li>2. NPAC SMS Simulator with the operationCancelled error.</li> </ol>
<b>Expected Results</b>	The LSMS correctly handles the error response operationCancelled error from the NPAC SMS Simulator.

### 13.9.8 MOC.LSMS.INV.DEL.serviceProvNPA-NXX

<b>Purpose</b>	To test the LSMS's ability to handle the M-DELETE error response processingFailure error to a previously initiated and valid M-DELETE request for an existing serviceProvNPA-NXX managed object instance.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if the LSMS is managing network data.
<b>Prerequisites</b>	MOC.LSMS.CAP.OP.DEL.serviceProvNPA-NXX
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-DELETE request for serviceProvNPA-NXX object.</li> <li>2. NPAC SMS Simulator with the processingFailure error.</li> </ol>
<b>Expected Results</b>	The LSMS correctly handles the error response processingFailure error from the NPAC SMS Simulator.

## 13.10 serviceProvLRN

<b>MO</b>	serviceProvLRN
<b>Purpose</b>	This section contains the test cases for the serviceProvLRN Managed Object Class pertaining to the LSMS to NPAC SMS Interface, as part of the Managed Object Conformance testing of the NPAC SMS Interoperability Test.
<b>Prerequisite</b>	A Service Provider and Network Data Management association function is established. A InpNPAC-SMS and a InpNetwork Managed Object Instances have been created inherently. All the test cases for the serviceProvNetwork

Managed Object Class have been performed.
---

13.10.1 MOC.LSMS.CAP.OP.GET.serviceProvLRN

<b>Purpose</b>	To test the LSMS's ability to GET all the attributes of the serviceProvLRN managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	A serviceProvLRN managed object instance with above serviceProvLRN-Value attribute has been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-GET request for all the attributes of a serviceProvLRN object.</li> <li>2. NPAC SMS Simulator responds with a successful M-GET result containing all attributes.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-GET request and retrieves the attributes successfully from the NPAC SMS Simulator.

13.10.2 MOC.LSMS.CAP.OP.DEL.serviceProvLRN

<b>Purpose</b>	To test the LSMS's ability to DELETE an existing serviceProvLRN managed object instance from the NPAC SMS Simulator.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if the service provider is managing network data.
<b>Prerequisites</b>	A serviceProvLRN managed object instance with above serviceProvLRN-Value attribute has been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-DELETE request for a serviceProvLRN object.</li> <li>2. NPAC SMS Simulator responds with a successful M-DELETE response.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-DELETE request and removes the managed object instance successfully from the NPAC SMS Simulator.

13.10.3 MOC.LSMS.VAL.CRE.AUTO.serviceProvLRN

<b>Purpose</b>	To test the LSMS's ability to CREATE a serviceProvLRN managed object instance in the NPAC SMS Simulator using automatic instance naming.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if the service provider is managing network data.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-CREATE request for a serviceProvLRN object.</li> <li>2. NPAC SMS Simulator responds with a successful M-CREATE response.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-CREATE request with automatic instance naming causing the serviceProvLRN instance to be created and its attributes populated successfully in the NPAC SMS

	Simulator.
--	------------

13.10.4 MOC.LSMS.VAL.GET.SCOP.FILT.serviceProvLRN

<b>Purpose</b>	To test the LSMS's ability to initiate a valid scoped and filtered M-GET request for all attributes. This will be accomplished by retrieving all the attributes for a specified LRN value (i.e. filtering on serviceProvLRN-Value equal to that number) starting at the base managed object serviceProvNetwork and ending at the serviceProvLRN.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. This functionality may be satisfied by getting instances one at a time.
<b>Prerequisites</b>	A serviceProvLRN managed object instance with above serviceProvLRN-Value attribute has been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-GET request for the attributes of the serviceProvLRN object with a filter for the specified serviceProvLRN-Value.</li> <li>2. NPAC SMS Simulator responds with a successful M-GET result containing the attribute.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-GET request and retrieves the attribute successfully from the NPAC SMS Simulator.

13.10.5 MOC.LSMS.VAL.DEL.SCOP.FILT.serviceProvLRN

<b>Purpose</b>	To test the LSMS's ability to initiate a valid scoped and filtered M-DELETE request for an existing managed object instance. This will be accomplished by deleting all the serviceProvLRN instances with the serviceProvLRN-Value equal to the NPA-NXX, starting at the base managed object serviceProvNetwork and ending at the serviceProvLRN.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. This functionality may also be satisfied by deleting instances one at a time. May be used to satisfy the requirements instead of MOC.LSMS.CAP.OP.DEL.serviceProvLRN.
<b>Prerequisites</b>	A serviceProvLRN managed object instance with above serviceProvLRN-Value attribute has been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-DELETE request for the serviceProvLRN object with a filter for the specified serviceProvLRN-Value.</li> <li>2. NPAC SMS Simulator responds with a successful M-DELETE response.</li> </ol>
<b>Expected Results</b>	The LSMS issues a valid M-DELETE request with a correct scope and filter causing the above instance to be removed successfully from the NPAC SMS Simulator.

13.10.6 MOC.LSMS.INV.CRE.serviceProvLRN

<b>Purpose</b>	To test the LSMS's ability to handle the M-CREATE error response
----------------	--

	duplicateManagedObjectInstance error to a previously initiated and valid M-CREATE request for the serviceProvLRN MO.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if the service provider is managing network data.
<b>Prerequisites</b>	MOC.LSMS.VAL.CRE.AUTO.serviceProvLRN
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-CREATE request for a serviceProvLRN object.</li> <li>2. NPAC SMS Simulator responds with a duplicateManagedObjectInstance error.</li> </ol>
<b>Expected Results</b>	The LSMS correctly handles the error response duplicateManagedObjectInstance error from the NPAC SMS Simulator.

13.10.7 MOC.LSMS.INV.GET.serviceProvLRN

<b>Purpose</b>	To test the LSMS's ability to handle the M-GET error response operationCancelled error to a previously initiated and valid M-GET request for all the attributes of a serviceProvLRN object.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	MOC.LSMS.CAP.OP.GET.serviceProvLRN
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-GET request for all the attributes of a serviceProvLRN object.</li> <li>2. NPAC SMS Simulator responds with an operationCancelled error.</li> </ol>
<b>Expected Results</b>	The LSMS correctly handles the error response operationCancelled error from the NPAC SMS Simulator.

13.10.8 MOC.LSMS.INV.DEL.serviceProvLRN

<b>Purpose</b>	To test the LSMS's ability to handle the M-DELETE error response processingFailure error to a previously initiated and valid M-DELETE request for an existing serviceProvLRN managed object instance.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Required if the service provider is managing network data.
<b>Prerequisites</b>	MOC.LSMS.CAP.OP.DEL.serviceProvLRN
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid M-DELETE request for a serviceProvLRN object.</li> <li>2. NPAC SMS Simulator responds with a processingFailure error.</li> </ol>
<b>Expected Results</b>	The LSMS correctly handles the error response processingFailure error from the NPAC SMS Simulator.

13.11 numberPoolBlockNPAC

<b>MO</b>	numberPoolBlockNPAC
<b>Purpose</b>	This section contains the test cases for the numberPoolBlockNPAC Managed

	Object Class pertaining to the LSMS to NPAC SMS Interface as part of the MOC testing of the NPAC SMS Simulator Interoperability Test.
<b>Prerequisite</b>	<ul style="list-style-type: none"> <li>• A LSMS Network and Subscription Data Download Management association function is established.</li> <li>• InpNPAC-SMS and InpSubscription Managed Object Instances exist.</li> </ul>

### 13.11.1 MOC.LSMS.CAP.OP.GET.numberPoolBlockNPAC

<b>Purpose</b>	Verify the LSMS's ability to GET all the attributes of the numberPoolBlockNPAC managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	A numberPoolBlockNPAC object exists on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS issues a valid M-GET request for all attributes of the numberPoolBlockNPAC object.</li> <li>2. NPAC SMS Simulator responds with a successful M-GET result containing all attributes.</li> </ol>
<b>Expected Results</b>	LSMS issues a valid M-GET request and retrieves the attributes successfully from the NPAC SMS Simulator.

### 13.11.2 MOC.LSMS.VAL.GET.SCOPE.numberPoolBlockNPAC

<b>Purpose</b>	Verify the LSMS's ability to initiate a valid scoped M-GET request for all the attributes of the numberPoolBlockNPAC managed object instance. This will be accomplished by retrieving all the attributes starting at the base managed object InpSubscriptions and ending at the numberPoolBlockNPAC.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	Multiple numberPoolBlockNPAC objects exist on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS issues a valid scoped and filtered M-GET request for the numberPoolBlockNPAC object(s).</li> <li>2. NPAC SMS Simulator responds with a successful M-GET result containing all attributes.</li> </ol>
<b>Expected Results</b>	LSMS issues a valid M-GET request and retrieves the object(s) successfully from the NPAC SMS Simulator.

### 13.11.3 MOC.LSMS.INV.GET.numberPoolBlockNPAC

<b>Purpose</b>	Verify the LSMS's ability to handle an error response, noSuchObjectInstance error, to a previously initiated and valid M-GET request for all the attributes of a numberPoolBlockNPAC managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	A numberPoolBlockNPAC exists on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS issues a valid M-GET request for all attributes of the numberPoolBlockNPAC object.</li> <li>2. NPAC SMS Simulator responds with a processingFailure error.</li> </ol>
<b>Expected Results</b>	LSMS successfully handles the error response from the NPAC SMS

	Simulator.
--	------------

13.11.4 MOC.LSMS.INV.GET.SCOPE.numberPoolBlockNPAC

<b>Purpose</b>	Verify the LSMS's ability to handle an error response, processingFailure, to a previously initiated and valid scope and filtered M-GET request for all the attributes of a numberPoolBlockNPAC managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	A numberPoolBlockNPAC exists on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS issues a valid scope and filtered M-GET request for all attributes of the numberPoolBlockNPAC object.</li> <li>2. NPAC SMS Simulator responds with a processingFailure error.</li> </ol>
<b>Expected Results</b>	LSMS successfully handles the error response from the NPAC SMS Simulator.

13.12 serviceProvNPA-NXX-X

<b>MO</b>	numberPoolBlockNPAC
<b>Purpose</b>	This section contains the test cases for the serviceProvNPA-NXX-X Managed Object Class pertaining to the LSMS to NPAC SMS Interface as part of the MOC testing of the NPAC SMS Simulator Interoperability Test.
<b>Prerequisite</b>	<ul style="list-style-type: none"> <li>• A LSMS Service Provider and Network Data Download Management association function is established.</li> <li>• InpNPAC-SMS and InpNetwork Managed Object Instances exist.</li> </ul>

13.12.1 MOC.LSMS.CAP.OP.GET.serviceProvNPA-NXX-X

<b>Purpose</b>	Verify the LSMS's ability to GET all the attributes of the serviceProvNPA-NXX-X managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	A serviceProvNPA-NXX-X object exists on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS issues a valid M-GET request for all attributes of the serviceProvNPA-NXX object.</li> <li>2. NPAC SMS Simulator responds with a successful M-GET result containing all attributes.</li> </ol>
<b>Expected Results</b>	LSMS issues a valid M-GET request and retrieves the attributes successfully from the NPAC SMS Simulator.

13.12.2 MOC.LSMS.VAL.GET.SCOPE.serviceProvNPA-NXX-X

<b>Purpose</b>	Verify the LSMS's ability to initiate a valid scoped M-GET request for all the attributes of the serviceProvNPA-NXX-X managed object instance. This will be accomplished by retrieving all the attributes starting at the base managed object InpSubscriptions and ending at the serviceProvNPA-NXX-X.
<b>Severity</b>	O

<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	Multiple serviceProvNPA-NXX-X objects exist on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS issues a valid scoped and filtered M-GET request for the serviceProvNPA-NXX-X object(s).</li> <li>2. NPAC SMS Simulator responds with a successful M-GET result containing all attributes.</li> </ol>
<b>Expected Results</b>	LSMS issues a valid M-GET request and retrieves the object(s) successfully from the NPAC SMS Simulator.

13.12.3 MOC.LSMS.INV.GET.serviceProvNPA-NXX-X

<b>Purpose</b>	Verify the LSMS's ability to handle an error response, noSuchObjectInstance error, to a previously initiated and valid M-GET request for all the attributes of a serviceProvNPA-NXX-X managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	A serviceProvNPA-NXX-X exists on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS issues a valid M-GET request for all attributes of the serviceProvNPA-NXX-X object.</li> <li>2. NPAC SMS Simulator responds with a processingFailure error.</li> </ol>
<b>Expected Results</b>	LSMS successfully handles the error response from the NPAC SMS Simulator.

13.12.4 MOC.LSMS.INV.GET.SCOPE.serviceProvNPA-NXX-X

<b>Purpose</b>	Verify the LSMS's ability to handle an error response, processingFailure, to a previously initiated and valid scope and filtered M-GET request for all the attributes of a serviceProvNPA-NXX-X managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	A serviceProvNPA-NXX-X exists on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS issues a valid scope and filtered M-GET request for all attributes of the serviceProvNPA-NXX-X object.</li> <li>2. NPAC SMS Simulator responds with a processingFailure error.</li> </ol>
<b>Expected Results</b>	LSMS successfully handles the error response from the NPAC SMS Simulator.



## 14 NPAC to LSMS MOC Test Cases

### 14.1 InpLocalSMS

<b>MO</b>	InpLocalSMS
<b>Purpose</b>	This section contains test cases for the InpLocalSMS Managed Object Class pertaining to the NPAC SMS Simulator to Local SMS Interface, as part of the Managed Object Conformance testing of the interoperability test. This capability test package checks the LSMS's existence and basic validity of the specified capabilities.
<b>Prerequisite</b>	A Service Provider and Network Data Management association function is established. The LSMS has successfully completed the Stack-to-Stack Interoperability testing.

#### 14.1.1 MOC.NPAC.CAP.OP.GET.InpLocalSMS

<b>Purpose</b>	Verify the capability of the InpLocalSMS managed object class to correctly respond to an M-GET request. The NPAC SMS Simulator intends to GET all the attributes.
<b>Severity</b>	O
<b>Severity Explanation</b>	Not required for LNP provisioning. However, LSMS can perform to verify managed object class.
<b>Prerequisites</b>	An instance has been inherently created on the LSMS.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>NPAC SMS Simulator sends a valid M-GET request for the InpLocalSMS object.</li> <li>LSMS responds with a successful M-GET result.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a getResult with the correct attribute information for all attributes.

#### 14.1.2 MOC.NPAC.INV.CRE.INH.InpLocalSMS

<b>Purpose</b>	This test case checks the LSMS's ability of responding to semantically invalid CMIP request. The NPAC SMS Simulator sends M-CREATE request intending to create an instance that can only be created inherently on Local SMS.
<b>Severity</b>	O
<b>Severity Explanation</b>	Not required for LNP provisioning. LSMS may perform to verify error handling.
<b>Prerequisites</b>	A InpLocalSMS instance exists on Local SMS.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>NPAC SMS Simulator sends a syntactically valid M-CREATE request for the InpLocalSMS object for the attribute.</li> <li>LSMS responds with a DuplicateManagedObjectInstance error.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives an error response with DuplicateManagedObjectInstance error. No instance is created on LSMS.

14.1.3 MOC.NPAC.INV.SET.InpLocalSMS

<b>Purpose</b>	This test case checks the LSMS's ability to respond to semantically invalid CMIP request. The NPAC SMS Simulator sends out an M-SET intending to override the read-only attribute InpLocal-SMS-Name.
<b>Severity</b>	O
<b>Severity Explanation</b>	Not required for LNP provisioning. LSMS may perform to verify error handling.
<b>Prerequisites</b>	A InpLocalSMS instance exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a syntactically valid M-SET request to the InpLocalSMS object for the InpLocal-SMS-Name attribute.</li> <li>2. LSMS responds with a setListError error response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a SetListError error response. The attribute is not replaced.

14.1.4 MOC.NPAC.INV.DEL.InpLocalSMS

<b>Purpose</b>	This test case checks the LSMS's ability to respond to semantically invalid CMIP request. The NPAC SMS Simulator sends out an M-DELETE request intending to delete the InpLocalSMS instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Not required for LNP provisioning. LSMS may perform to verify error handling.
<b>Prerequisites</b>	A InpLocalSMS instance exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a syntactically valid M-DELETE request to the InpLocalSMS object for the InpLocal-SMS-Name attribute.</li> <li>2. LSMS responds with a processingFailure error response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a processingFailure error response. The instance is not removed from the LSMS.

14.1.5 MOC.LSMS.CAP.NOT.InpNPAC-SMS-Operational-Information

<b>Purpose</b>	To test the LSMS's ability to receive the M-EVENT-REPORT for the InpNPAC-SMS-Operational-Information Notification.
<b>Severity</b>	R
<b>Severity Explanation</b>	This test case is required for LSMS functionality.
<b>Prerequisites</b>	A InpNPAC-SMS MO instance has been inherently created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The NPAC SMS Simulator sends a InpNPAC-SMS-Operational-Information M-EVENT-REPORT to the LSMS to inform them of coming downtime.</li> <li>2. The LSMS responds with the M-EVENT-REPORT confirmation.</li> </ol>
<b>Expected Results</b>	The LSMS receives the M-EVENT-REPORT and sends notification confirmation to the NPAC SMS.

## 14.2 *InpSubscriptions*

<b>MO</b>	InpSubscriptions
<b>Purpose</b>	This section contains test cases for the InpSubscriptions Managed Object Class pertaining to the NPAC SMS Simulator NPAC SMS Simulator to Local SMS Interface, as part of the Managed Object Conformance testing of the interoperability test. This capability test package checks the LSMS's existence and basic validity of the specified capabilities.
<b>Prerequisite</b>	A LSMS Network and Subscription Data Download association function is established. The LSMS has successfully completed the Stack-to-Stack Interoperability testing. The LSMS has successfully completed the MOC.NPAC.CAP.InpLocalSMS test package.

### 14.2.1 MOC.NPAC.CAP.OP.GET.InpSubscriptions

<b>Purpose</b>	Verify the capability of the InpSubscriptions managed object class to correctly respond to an M-GET request. The NPAC SMS Simulator intends to GET all attributes.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. Not required for LNP provisioning. LSMS may perform to verify object instance.
<b>Prerequisites</b>	A InpSubscriptions instance exists on LSMS.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid M-GET request to the InpSubscriptions object for all attributes.</li> <li>2. LSMS responds with an M-GET result containing all the attributes.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a getResult with all the attribute values.

### 14.2.2 MOC.NPAC.CAP.OP.ACT.InpSubscriptions

<b>Purpose</b>	Verify the capability of the InpSubscriptions managed object class to correctly respond to a confirmed M-ACTION request. The NPAC SMS Simulator intends to perform a subscriptionVersionLocalSMS-CreateAction on the object instance.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service.
<b>Prerequisites</b>	A InpSubscriptions instance has been inherently created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends the subscriptionVersionLocalSMS-CreateAction to the InpSubscriptions object.</li> <li>2. LSMS responds with a successful LocalSMS-CreateReply M-ACTION response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a successful LocalSMS-CreateReply.

#### 14.2.3 MOC.NPAC.CAP.OP.NOT.InpSubscriptions

<b>Purpose</b>	This test case checks the capability of the InpSubscriptions to correctly send the subscriptionVersionLocalSMS-ActionResults notification for the object instance.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service.
<b>Prerequisites</b>	A InpSubscriptions instance has been inherently created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends the subscriptionVersionLocalSMS-ActionResults M-EVENT-REPORT to the InpSubscriptions object.</li> <li>2. NPAC SMS Simulator confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a subscriptionVersionLocalSMS-ActionResults notification.

#### 14.2.4 MOC.NPAC.INV.CRE.INH.InpSubscriptions

<b>Purpose</b>	This test case checks the LSMS's ability to respond to a semantically invalid CMIP request. The NPAC SMS Simulator sends out an M-CREATE intending to create an instance that can only be created inherently on the LSMS.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. LSMS may perform to verify error handling.
<b>Prerequisites</b>	A InpLocalSMS instance exists on LSMS.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid M-CREATE request for the InpSubscriptions object.</li> <li>2. LSMS responds with a processingFailure or duplicateObjectInstance error.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a processingFailure or duplicateObjectInstance response with an ObjectInstance. No instance is created on LSMS.

#### 14.2.5 MOC.NPAC.INV.SET.InpSubscriptions

<b>Purpose</b>	This test case checks the LSMS's ability to respond to semantically invalid CMIP request. The NPAC SMS Simulator sends out an M-SET intending to override the read-only attribute InpSubscriptionsName.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. LSMS may perform to verify error handling.
<b>Prerequisites</b>	A InpSubscriptions instance exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a syntactically valid M-CREATE request for the InpSubscriptions object.</li> <li>2. LSMS responds with a setListError error.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a SetListError response. Thus the attribute is not replaced.

14.2.6 MOC.NPAC.INV.ACT.SYN.ID.InpSubscriptions

<b>Purpose</b>	This test case checks the LSMS's ability to respond to syntactically invalid CMIP request. The NPAC SMS Simulator sends out a confirmed M-ACTION request with an invalid actionID.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.ACT.InpSubscriptions
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a syntactically invalid subscriptionVersionLocalSMS-CreateAction to the InpSubscriptions object.</li> <li>2. LSMS responds with an invalidArgumentValue error response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives an error response with the error type set to invalidArgumentValue. No action is performed as the result.

14.2.7 MOC.NPAC.INV.ACT.SYN.CLS.InpSubscriptions

<b>Purpose</b>	This test case checks the LSMS's ability to respond to syntactically invalid CMIP request. The NPAC SMS Simulator sends out a confirmed M-ACTION request with an invalid object class.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.ACT.InpSubscriptions
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a syntactically invalid LocalSMS-CreateAction to the InpSubscriptions object.</li> <li>2. LSMS responds with a classInstanceConflict error.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a BaseManagedObjectId response with the error type set to classInstanceConflict. No action is performed as the result.

14.2.8 MOC.NPAC.INV.ACT.InpSubscriptions

<b>Purpose</b>	This test case checks the LSMS's ability to respond to syntactically invalid CMIP request. The NPAC SMS Simulator sends out a confirmed M-ACTION request with an invalid action OID.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.ACT.InpSubscriptions
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a syntactically invalid LocalSMS-CreateAction to the InpSubscriptions object.</li> <li>2. LSMS responds with a noSuchAction error response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a NoSuchAction response. The action is not performed and the NPAC SMS Simulator gets an ActionResult, which indicates that the action failed.

14.2.9 MOC.NPAC.INV.NOT.InpSubscriptions

<b>Purpose</b>	This test case checks the LSMS's ability to handle the
----------------	--

	processingFailure error in response to the subscriptionVersionLocalSMS-ActionResults notification.
<b>Severity</b>	R
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.NOT.InpSubscriptions
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends the subscriptionVersionLocalSMS-ActionResults M-EVENT-REPORT to the InpSubscriptions object.</li> <li>2. NPAC SMS Simulator replies with a processingFailure error.</li> </ol>
<b>Expected Results</b>	The LSMS handles the processingFailure error that is sent by the NPAC SMS Simulator in response to a valid subscriptionVersionLocalSMS-ActionResults notification.

#### 14.2.10 MOC.NPAC.INV.DEL.InpSubscriptions

<b>Purpose</b>	This test case checks the LSMS's ability to respond to semantically invalid CMIP request. The NPAC SMS Simulator sends out an M-DELETE request intending to delete the inherent instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	May be performed to validate LSMS error handling.
<b>Prerequisites</b>	A InpSubscriptions instance exists on LSMS
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a syntactically valid M-DELETE request to the InpSubscriptions object.</li> <li>2. LSMS responds with a processingFailure error response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a processingFailure error response. No instance is removed from the LSMS.

### 14.3 InpNetwork

<b>MO</b>	InpNetwork
<b>Purpose</b>	This section contains test cases for the InpNetwork Managed Object Class pertaining to the NPAC SMS Simulator to Local SMS Interface, as part of the Managed Object Conformance testing of the interoperability test. This capability test package checks the LSMS's existence and basic validity of the specified capabilities.
<b>Prerequisite</b>	A LSMS Network and Subscription Data Download association function is established. The LSMS has successfully completed the Stack-to-Stack Interoperability testing.

#### 14.3.1 MOC.NPAC.CAP.OP.GET.InpNetwork

<b>Purpose</b>	This test case checks the capability of the InpNetwork managed object class to correctly respond to an M-GET request. The NPAC SMS Simulator intends to GET all the attributes of InpNetwork.
<b>Severity</b>	O
<b>Severity Explanation</b>	LSMS may perform to verify managed object.
<b>Prerequisites</b>	A InpNetwork instance exists on LSMS.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid M-GET request for all the attributes of the InpNetwork object.</li> </ol>

	2. LSMS responds with a successful M-GET result containing all the attributes.
<b>Expected Results</b>	The NPAC SMS Simulator receives a getResult with all attribute information.

#### 14.3.2 MOC.NPAC.INV.CRE.INH.InpNetwork

<b>Purpose</b>	This test case checks the LSMS's ability to respond to semantically invalid CMIP request. The NPAC SMS Simulator sends out an M-CREATE intending to create an instance that can only be created inherently on the LSMS.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. LSMS may perform to verify error handling.
<b>Prerequisites</b>	A InpLocalSMS instance exists on LSMS
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a syntactically valid M-CREATE request to the InpNetwork object.</li> <li>2. LSMS responds with a processingFailure or duplicateObjectInstance error response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a processingFailure or DuplicateObjectInstance error response. No instance is created on LSMS.

#### 14.3.3 MOC.NPAC.INV.SET.InpNetwork

<b>Purpose</b>	This test case checks the LSMS's ability to respond to a semantically invalid CMIP request. The NPAC SMS Simulator sends out an M-SET intending to override the read-only attribute InpNetworkName.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. LSMS may perform to verify error handling.
<b>Prerequisites</b>	A InpNetwork instance exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a syntactically valid M-SET request to the InpNetwork object for the InpNetworkName attribute.</li> <li>2. LSMS responds with a setListError error response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a setListError error response. The attribute is not replaced.

#### 14.3.4 MOC.NPAC.INV.ACT.InpNetwork

<b>Purpose</b>	This test case checks the LSMS's ability to respond to syntactically invalid CMIP request. The NPAC SMS Simulator sends out an M-ACTION request intending to perform the InpDownload action.
<b>Severity</b>	O
<b>Severity Explanation</b>	LSMS may perform to verify error handling.
<b>Prerequisites</b>	A InpNetwork instance exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a syntactically valid InpDownload</li> </ol>

	M-ACTION request to the InpNetwork object. 2. LSMS responds with a noSuchAction error response.
<b>Expected Results</b>	The NPAC SMS Simulator receives a noSuchAction error response. No action is performed on LSMS.

#### 14.3.5 MOC.NPAC.INV.DEL.InpNetwork

<b>Purpose</b>	This test case checks the LSMS's ability to respond to syntactically invalid CMIP request. The NPAC SMS Simulator sends out an M-DELETE request intending to delete an instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. LSMS may perform to verify error handling.
<b>Prerequisites</b>	A InpNetwork instance exists.
<b>Procedure</b>	1. NPAC SMS Simulator sends a syntactically invalid M-DELETE request to the InpNetwork object. 2. LSMS responds with a processingFailure error response.
<b>Expected Results</b>	The NPAC SMS Simulator receives an error response with error type set to processingFailure. Thus no instance is removed from the LSMS.

#### 14.4 subscriptionVersion

<b>MO</b>	subscriptionVersion
<b>Purpose</b>	This section contains capability test cases for the subscriptionVersion Managed Object Class pertaining to the NPAC SMS Simulator to Local SMS Interface. This capability test checks the existence and the basic validity of the LSMS's capability.
<b>Prerequisite</b>	A LSMS Network and Subscription Data Download association function is established. The LSMS has successfully completed MOC.NPAC.CAP.InpSubscriptions test. A InpSubscriptions instance exists on the Local SMS.

#### 14.4.1 MOC.NPAC.CAP.OP.CRE.subscriptionVersion

<b>Purpose</b>	This test case checks the capability of the LSMS to correctly respond to an M-CREATE request. The NPAC SMS Simulator intends to create a subscriptionVersion instance.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service.
<b>Prerequisites</b>	
<b>Procedure</b>	1. NPAC SMS Simulator sends a valid M-CREATE request for a subscriptionVersion object. 2. LSMS responds with a successful M-CREATE response.
<b>Expected Results</b>	The NPAC SMS Simulator receives a CreateResult and an instance is created on Local SMS.



#### 14.4.2 MOC.NPAC.CAP.OP.SET.subscriptionVersion

<b>Purpose</b>	This test case checks the capability of the subscriptionVersion managed object class to correctly respond to an M-SET request. The NPAC SMS Simulator intends to modify one attribute.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service.
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.CRE.subscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid M-SET request to a subscriptionVersion object for a specified attribute.</li> <li>2. LSMS responds with a successful M-SET result.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a setResult. The attribute value of the subscription version is replaced with the given value.

#### 14.4.3 MOC.NPAC.CAP.OP.GET.subscriptionVersion

<b>Purpose</b>	This test case checks the capability of the subscriptionVersion managed object class to correctly respond to an M-GET request. The NPAC SMS Simulator intends to GET all the attributes of the instance created above.
<b>Severity</b>	R
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.CRE.subscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid M-GET request to a subscriptionVersion object for all attributes.</li> <li>2. LSMS responds with a successful M-GET result containing all attributes.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a getResult with all the attributes of the instance.

#### 14.4.4 MOC.NPAC.CAP.OP.DEL.subscriptionVersion

<b>Purpose</b>	This test case checks the capability of the LSMS to correctly respond to an M-DELETE request. The NPAC SMS Simulator intends to delete a subscriptionVersion instance.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service.
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.CRE.subscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid M-DELETE request to a subscriptionVersion object.</li> <li>2. LSMS responds with a successful M-DELETE response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a DeleteResult. The instance is removed from the LSMS.

#### 14.4.5 MOC.NPAC.VAL.SET.SING.subscriptionVersion

<b>Purpose</b>	This behavior test case checks the capability of the subscriptionVersion managed object class to correctly respond to an M-SET request. The NPAC SMS Simulator intends to SET one
----------------	---

	attribute.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service.
<b>Prerequisites</b>	MOC.NPAC.VAL.CRE.subscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid M-SET request to a subscriptionVersion object for a specified attribute.</li> <li>2. LSMS responds with a successful M-SET result.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a setResult, The attribute value of the subscriptionVersion is replaced by the new value.

#### 14.4.6 MOC.NPAC.VAL.SET.MULT.subscriptionVersion

<b>Purpose</b>	This behavior test case checks the capability of the subscriptionVersion managed object class to correctly respond to an M-SET request. The NPAC SMS Simulator intends to modify a group of attributes of the instance.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service.
<b>Prerequisites</b>	MOC.NPAC.VAL.CRE.subscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid M-SET request to a subscriptionVersion object for the specified attributes.</li> <li>2. LSMS responds with a successful M-SET result.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a setResult. The attribute values of the subscriptionVersion are replaced by the new values.

#### 14.4.7 MOC.NPAC.VAL.SET.SCOPE.FILT.subscriptionVersion

<b>Purpose</b>	This test case checks the capability of the subscriptionVersion managed object class to correctly respond to a scoped and filtered M-SET request. The NPAC SMS Simulator will set one attribute, the subscriptionBillingId, for all instances of the subscriptionVersion which have an agreed upon subscriptionTN range. The scope begins at the base managed object InpSubscriptions and is one level down.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service.
<b>Prerequisites</b>	MOC.NPAC.VAL.CRE.subscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid M-SET request to a subscriptionVersion object for the subscriptionBillingId attribute with a filter set to equality for the subscriptionTN.</li> <li>2. LSMS responds with a successful M-SET result.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a setResult for all the modified instances.

#### 14.4.8 MOC.NPAC.VAL.GET.SCOPE.FILT.subscriptionVersion

<b>Purpose</b>	This test case checks the capability of the subscriptionVersion to correctly respond to a scoped and filtered M-GET request. The NPAC SMS Simulator will request all attributes for an instance of
----------------	--

	the subscriptionVersion which has an agreed upon subscriptionTN range. The scope begins at the base managed object InpSubscriptions and is one level down.
<b>Severity</b>	R
<b>Severity Explanation</b>	Impacts the ability to provide LNP service.
<b>Prerequisites</b>	MOC.NPAC.VAL.SET.FILT.subscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid M-GET request for a subscriptionVersion object for all the attributes with a filter set to equality for the subscriptionTN.</li> <li>2. LSMS responds with a successful M-GET result containing the attribute.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a getResult with the requested attribute for all the instances that met the filtering criteria.

14.4.9 MOC.NPAC.VAL.DEL.SCOP.FILT.subscriptionVersion

<b>Purpose</b>	This behavior test case checks the capability of the subscriptionVersion to correctly respond to a scoped M-DELETE request. The NPAC SMS Simulator intends to delete instances satisfying the following criteria: Base Managed Object is InpSubscriptions, Scope is level 1, filter is a TN range of an agreed upon range.
<b>Severity</b>	R
<b>Severity Explanation</b>	Impacts providing LNP service.
<b>Prerequisites</b>	MOC.NPAC.VAL.CRE.subscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid M-DELETE request for the subscriptionVersion objects with a filter set to a subscriptionTN range.</li> <li>2. LSMS responds with the successful M-DELETE responses.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a linked DeleteResult. The instances are removed from the LSMS.

14.4.10 MOC.NPAC.INV.CRE.subscriptionVersion

<b>Purpose</b>	The NPAC SMS Simulator sends out an M-CREATE CMIP request to LSMS with an invalid value set to an attribute. This tests the ability of the LSMS detecting the error and responding with the correct error message.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.CRE.subscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an invalid M-CREATE request for a subscriptionVersion.</li> <li>2. LSMS responds with an invalidAttributeValue error.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator should receive an invalidAttributeValue error response. No instance is created on the LSMS site as a result of the error information.

14.4.11 MOC.NPAC.INV.SET.RO.subscriptionVersion

<b>Purpose</b>	Tests the ability of the LSMS to respond to an invalid M-SET request. The NPAC SMS Simulator will attempt to M-SET the read-only attribute subscriptionVersionId.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.SET.subscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an invalid M-SET request for a subscriptionVersion for the subscriptionVersionId attribute.</li> <li>2. LSMS responds with a setListError error response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator gets a setListError response. The attribute's value is not replaced by the set request on the LSMS.

14.4.12 MOC.NPAC.INV.SET.MULT.subscriptionVersion

<b>Purpose</b>	The NPAC SMS Simulator sends out a confirmed M-SET request intending to set multiple attributes values with one invalid attribute value in the list. This tests the ability of the LSMS to correctly respond to an invalid request.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.SET.subscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an invalid M-SET request for a subscriptionVersion for several specified attributes, one of which contains an invalid syntax.</li> <li>2. LSMS responds with a setListError error response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator gets a SetListError error response. The attribute's value is not replaced by the set request.

14.4.13 MOC.NPAC.INV.SET.SYN.subscriptionVersion

<b>Purpose</b>	The NPAC SMS Simulator sends out a confirmed M-SET request intending to set the attribute subscriptionISVM-SSN to invalid value. This tests the LSMS's ability of detecting the invalid syntax and responding with correct error message.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.SET.subscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an invalid M-SET request for a subscriptionVersion for the subscriptionISVM-SSN attribute, which contains invalid syntax.</li> <li>2. LSMS responds with a setListError error response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator gets a setListError error. The attribute value is not replaced by the new value.

14.4.14 MOC.NPAC.INV.SET.SCOP.subscriptionVersion

<b>Purpose</b>	The NPAC SMS Simulator sends out a confirmed M-SET request
----------------	--

	intending to set with an invalid scope parameter. This tests the ability of the LSMS to correctly respond to an invalid request.
<b>Severity</b>	R
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.SET.subscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an invalid M-SET request for a subscriptionVersion with an incorrect parameter.</li> <li>2. LSMS responds with a setListError error response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator gets a processingFailure error. The attribute's value is not replaced by the set request.

#### 14.4.15 MOC.NPAC.INV.DEL.SCOP.subscriptionVersion

<b>Purpose</b>	The NPAC SMS Simulator sends an M-DELETE request intending to delete an instance with invalid scope. This tests the LSMS's ability of preventing the object instances from being deleted incorrectly.
<b>Severity</b>	R
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.DEL.subscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an invalid scoped M-DELETE request for a subscriptionVersion.</li> <li>2. LSMS responds with a processingFailure error response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator gets a processingFailure error response. No instance is removed from LSMS.

#### 14.4.16 MOC.NPAC.BND.SET.MIN.subscriptionVersion

<b>Purpose</b>	This test case checks the LSMS's ability of responding to an M-SET request. The NPAC SMS Simulator sends an M-SET intending to set the attributes billingId and endUserLocationValue to values with string lengths of 1 each.
<b>Severity</b>	R
<b>Severity Explanation</b>	LSMS must perform to validate bounds check.
<b>Prerequisites</b>	MOC.NPAC.VAL.subscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid M-SET request setting the subscriptionBillingId and subscriptionEndUserLocationValue to values with a length of 1.</li> <li>2. LSMS responds a successful M-SET result.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a setResult and the attributes are updated with the new values on the LSMS.

#### 14.4.17 MOC.NPAC.BND.SET.MAX.subscriptionVersion

<b>Purpose</b>	This test case checks the LSMS's ability of responding to an M-SET request. The NPAC SMS Simulator sends out an M-SET intending to set the attributes billingId and endUserLocationValue to values with string lengths of 4 and 12 respectively.
----------------	--

<b>Severity</b>	R
<b>Severity Explanation</b>	LSMS must perform to validate bounds check.
<b>Prerequisites</b>	MOC.NPAC.VAL.subscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid M-SET request setting the subscriptionBillingId and subscriptionEndUserLocationValue to values with a length of 4 and 12 respectively,</li> <li>2. LSMS responds a successful M-SET result.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a setResult and the attributes of the instance are set to the new values on the LSMS.

### 14.5 serviceProvNetwork

<b>MO</b>	serviceProvNetwork
<b>Purpose</b>	This section contains test cases for the InpLocalSMS Managed Object Class pertaining to the NPAC SMS Simulator to Local SMS Interface, as part of the Managed Object Conformance testing of the Interoperability Test. This capability test package checks the LSMS's existence and basic validity of the specified capabilities.
<b>Prerequisite</b>	A Service Provider and Network Data Management association function is established. The LSMS has successfully completed the Stack-to-Stack Interoperability testing. The LSMS has successfully completed the MOC.NPAC.CAP.InpLocalSMS tests.

#### 14.5.1 MOC.NPAC.CAP.OP.CRE.serviceProvNetwork

<b>Purpose</b>	This test case checks the capability of the LSMS to correctly respond to an M-CREATE request. The NPAC SMS Simulator intends to create a serviceProvNetwork instance.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service.
<b>Prerequisites</b>	A InpLocalSMS instance exists on Local SMS.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid M-CREATE request for a serviceProvNetwork object.</li> <li>2. LSMS responds with a successful M-CREATE response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a CreateResult and an instance is created on Local SMS.

#### 14.5.2 MOC.NPAC.CAP.OP.GET.serviceProvNetwork

<b>Purpose</b>	This test case checks the capability of the serviceProvNetwork to correctly respond to an M-GET request. The NPAC SMS Simulator intends to GET all the attributes of the instance created above.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. LSMS may perform to verify object.
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.CRE.serviceProvNetwork
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid M-GET request for all attributes of the serviceProvNetwork object.</li> <li>2. LSMS responds with a successful M-GET result containing all</li> </ol>

	the attributes.
<b>Expected Results</b>	The NPAC SMS Simulator receives a getResult with all the attributes of the instance.

#### 14.5.3 MOC.NPAC.CAP.OP.SET.serviceProvNetwork

<b>Purpose</b>	This test case checks the capability of the serviceProvNetwork managed object class to correctly respond to an M-SET request. The NPAC SMS Simulator intends to SET one attribute, the serviceProvName.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service.
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.CRE.serviceProvNetwork
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid M-SET request for the serviceProvName attribute.</li> <li>2. LSMS responds with a successful M-SET result.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a setResult, the serviceProvName is replaced by the new value.

#### 14.5.4 MOC.NPAC.CAP.OP.DEL.serviceProvNetwork

<b>Purpose</b>	This test case checks the capability of the LSMS to correctly respond to an M-DELETE request. The NPAC SMS Simulator intends to delete the serviceProvNetwork instance created above.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service.
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.CRE.serviceProvNetwork
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid M-DELETE request for the serviceProvNetwork object.</li> <li>2. LSMS responds with a successful M-DELETE response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a DeleteResult. The instance is removed from the LSMS.

#### 14.5.5 MOC.NPAC.INV.CRE.DUP.serviceProvNetwork

<b>Purpose</b>	This test case checks the LSMS's ability to respond to invalid CMIP request. The NPAC SMS Simulator sends out an M-CREATE intending to create a duplicate instance.
<b>Severity</b>	R
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.CRE.subscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid M-GET request for the serviceProvNetwork object for the specified attributes.</li> <li>2. LSMS responds with a successful M-GET result containing the attributes.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a duplicateObjectInstance error response. No instance is created on LSMS as a result.

14.5.6 MOC.NPAC.INV.SET.RO.serviceProvNetwork

<b>Purpose</b>	This test case checks the LSMS's ability to respond to invalid CMIP request. The NPAC SMS Simulator sends out an M-SET intending to override the read-only attribute serviceProvID.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.SET.subscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an invalid M-SET request for the serviceProvID attribute.</li> <li>2. LSMS responds with a setListError error response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a setListError error response. The attribute is not replaced.

14.5.7 MOC.NPAC.INV.SET.SYN.serviceProvNetwork

<b>Purpose</b>	This test case checks the LSMS's ability to respond to syntactically invalid CMIP requests. The NPAC SMS Simulator sends out an M-SET intending to set an invalid attribute value- the serviceProvName is set to length of 41.
<b>Severity</b>	O
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.SET.subscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an invalid M-SET request for the serviceProvName attribute.</li> <li>2. LSMS responds with a processingFailure error response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a processingFailure error response. The attribute is not replaced.

14.5.8 MOC.NPAC.INV.SET.serviceProvNetwork

<b>Purpose</b>	This test case checks the LSMS's ability to respond to a syntactically invalid CMIP requests. The NPAC SMS Simulator sends out an M-SET intending to set an invalid attribute value- the serviceProvName is set to length of 0.
<b>Severity</b>	O
<b>Severity Explanation</b>	LSMS may perform to verify error handling.
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.SET.subscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an invalid M-SET request for the serviceProvName attribute.</li> <li>2. LSMS responds with a processingFailure error response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator processingFailure error response. The attribute is not replaced.

14.5.9 MOC.NPAC.INV.GET.serviceProvNetwork

<b>Purpose</b>	This test case checks the LSMS's ability to respond to an invalid CMIP request. The NPAC SMS Simulator sends out an M-GET intending to get an invalid attribute value from the
----------------	--



	serviceProvNetwork object.
<b>Severity</b>	O
<b>Severity Explanation</b>	LSMS may perform to verify error handling.
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.GET.subscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an invalid M-GET request for the serviceProvNetwork object.</li> <li>2. LSMS responds with a getListError error response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives getListError error response.

14.5.10 MOC.NPAC.INV.DEL.serviceProvNetwork

<b>Purpose</b>	This test case checks the LSMS's ability to respond to a syntactically invalid CMIP request. The NPAC SMS Simulator sends out an M-DELETE request intending to delete a serviceProvNetwork object that does not exist.
<b>Severity</b>	R
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.DEL.subscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a invalid M-DELETE request for the nonexistent serviceProvNetwork object.</li> <li>2. LSMS responds with a noSuchObjectInstance error response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a noSuchObjectInstance error response. No instance is removed from the LSMS.

14.5.11 MOC.NPAC.INV.DEL.CO.serviceProvNetwork

<b>Purpose</b>	This test case checks the LSMS's ability to respond to a syntactically invalid CMIP request. The NPAC SMS Simulator sends out an M-DELETE request intending to delete a serviceProvNetwork object that contains serviceProvLRN and/or serviceProvNPA-NXX objects.
<b>Severity</b>	R
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.DEL.subscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an invalid M-DELETE request for the serviceProvNetwork object objects contained in it.</li> <li>2. LSMS responds with a processingFailure error response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a processingFailure error response. No instance is removed from the LSMS.

14.5.12 MOC.NPAC.BND.SET.MIN.serviceProvNetwork

<b>Purpose</b>	This test case checks the LSMS's ability of responding to a CMIP M-SET request. The NPAC SMS Simulator sends an M-SET intending to set the attribute ServiceProvName to a value of string length one.
<b>Severity</b>	R
<b>Severity Explanation</b>	LSMS must perform to verify bounds checking.

<b>Prerequisites</b>	MOC.NPAC.VAL.serviceProvNetwork
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid M-SET request for the serviceProvName attribute.</li> <li>2. LSMS responds with successful M-SET result.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a setResult. The attribute is updated with the new value.

#### 14.5.13 MOC.NPAC.BND.SET.MAX.serviceProvNetwork

<b>Purpose</b>	This test case checks the LSMS's ability of responding to a CMIP M-SET request. The NPAC SMS Simulator sends out an M-SET intending to set the attribute ServiceProvName to a value of string length forty.
<b>Severity</b>	R
<b>Severity Explanation</b>	LSMS must perform to verify bounds checking.
<b>Prerequisites</b>	MOC.NPAC.VAL.serviceProvNetwork
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid M-SET request for the serviceProvName attribute.</li> <li>2. LSMS responds with successful M-SET result.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a setResult, and the serviceProvName is set to the new value.

### 14.6 serviceProvNPA-NXX

<b>MO</b>	serviceProvNPA-NXX
<b>Purpose</b>	This section contains test cases for the serviceProvNPA-NXX Managed Object Class pertaining to the NPAC SMS Simulator to Local SMS Interface, as part of the Managed Object Conformance testing of the interoperability test. This capability test verifies the existence and the basic validity of the LSMS's capability.
<b>Prerequisite</b>	A Service Provider and Network Data Management association function is established. The LSMS has successfully complete MOC.NPAC.CAP.serviceProvNetwork test. A serviceProvNetwork instance exists on Local SMS.

#### 14.6.1 MOC.NPAC.CAP.OP.CRE.serviceProvNPA-NXX

<b>Purpose</b>	This test case checks the capability of the LSMS to correctly respond to an M-CREATE request. The NPAC SMS Simulator intends to create a serviceProvNPA-NXX instance.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service.
<b>Prerequisites</b>	MOC.NPAC.CAP.serviceProvNetwork
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid M-CREATE request for the serviceProvNPA-NXX object.</li> <li>2. LSMS responds with successful M-CREATE result.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a CreateResult and an instance is created on Local SMS.

14.6.2 MOC.NPAC.CAP.OP.DEL.serviceProvNPA-NXX

<b>Purpose</b>	This test case checks the capability of the LSMS to correctly respond to an M-DELETE request. The NPAC SMS Simulator intends to delete a serviceProvNPA-NXX instance.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service.
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.CRE.serviceProvNPA-NXX
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid M-DELETE request for the serviceProvNPA-NXX object.</li> <li>2. LSMS responds with successful M-DELETE result.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a DeleteResult. The instance is removed from the LSMS.

14.6.3 MOC.NPAC.INV.CRE.DUP.serviceProvNPA-NXX

<b>Purpose</b>	This test case checks the LSMS's ability to respond to a syntactically invalid CMIP request. The NPAC SMS Simulator sends out an M-CREATE intending to create a duplicate instance.
<b>Severity</b>	R
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.CRE.serviceProvNPA-NXX
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an invalid M-CREATE request for an existing serviceProvNPA-NXX object.</li> <li>2. LSMS responds with a duplicateManagedObjectInstance error response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a duplicateManagedObjectInstance error response. No instance is created as a result.

14.6.4 MOC.NPAC.INV.SET.serviceProvNPA-NXX

<b>Purpose</b>	This test case checks the LSMS's ability to respond to a semantically invalid CMIP request. The NPAC SMS Simulator sends out an M-SET intending to override the read-only attribute.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. LSMS may perform to verify error handling.
<b>Prerequisites</b>	A serviceProvNPA-NXX instance exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an invalid M-SET request for the specified attribute of the serviceProvNPA-NXX object.</li> <li>2. LSMS responds with a setListError error response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a setListError error response. The attribute is not replaced.

14.6.5 MOC.NPAC.INV.DELserviceProvNPA-NXX

<b>Purpose</b>	This test case checks the LSMS's ability to respond to invalid CMIP
----------------	---

	request. The NPAC SMS Simulator sends out an M-DELETE request intending to delete a non-existent instance.
<b>Severity</b>	R
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.DEL.serviceProvNPA-NXX
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an invalid M-DELETE request for the non-existent serviceProvNPA-NXX object.</li> <li>2. LSMS responds with a noSuchObjectInstance error response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives noSuchObjectInstance error response. No instance is removed from the LSMS.

### 14.7 serviceProvLRN

<b>MO</b>	serviceProvLRN
<b>Purpose</b>	This section contains test cases for the serviceProvLRN Managed Object Class pertaining to the NPAC SMS Simulator to Local SMS LSMS Interface, as part of the Managed Object Conformance testing of the interoperability test. This capability test package checks the LSMS's existence and basic validity of the specified capabilities.
<b>Prerequisite</b>	A Service Provider and Network Data Management association function is established. The LSMS has successfully completed the Stack-to-Stack Interoperability testing. The LSMS has successfully completed the MOC.NPAC.CAP.serviceProvNetwork test. There is a serviceProvNetwork object existing on Local SMS.

#### 14.7.1 MOC.NPAC.CAP.OP.CRE.serviceProvLRN

<b>Purpose</b>	This test case checks the capability of the LSMS to correctly respond to an M-CREATE request. The NPAC SMS Simulator intends to create an instance.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service.
<b>Prerequisites</b>	MOC.NPAC.CAP.serviceProvNetwork
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid M-CREATE request for a serviceProvLRN object.</li> <li>2. LSMS responds with a successful M-CREATE response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a CreateResult and an instance is created on Local SMS.

#### 14.7.2 MOC.NPAC.CAP.OP.DEL.serviceProvLRN

<b>Purpose</b>	This behavior test case checks the capability of the serviceProvLRN managed object class to correctly respond to an M-DELETE request. The NPAC SMS Simulator intends to delete an instance.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on providing LNP service.

<b>Prerequisites</b>	MOC.NPAC.CAP.OP.CRE.serviceProvLRN
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a valid M-DELETE request for the serviceProvLRN object.</li> <li>2. LSMS responds with a successful M-DELETE response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a DeleteResult. The instance is removed from the LSMS.

#### 14.7.3 MOC.NPAC.INV.CRE.DUP.serviceProvLRN

<b>Purpose</b>	This test case checks the LSMS's ability to respond to an invalid CMIP request. The NPAC SMS Simulator sends out an M-CREATE intending to create a duplicate instance.
<b>Severity</b>	R
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.CRE.serviceProvLRN
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an invalid M-CREATE request for the existing serviceProvLRN object.</li> <li>2. LSMS responds with a duplicateManagedObjectInstance error response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a duplicateManagedObjectInstance error response. No instance is created on LSMS as a result.

#### 14.7.4 MOC.NPAC.INV.SET.serviceProvLRN

<b>Purpose</b>	This test case checks the LSMS's ability to respond to an invalid CMIP request. The NPAC SMS Simulator sends out an M-SET intending to override the read-only attribute serviceProvLRN-ID.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. LSMS may perform to verify error handling.
<b>Prerequisites</b>	A serviceProvLRN instance exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an invalid M-SET request for the specified attribute of the serviceProvLRN object.</li> <li>2. LSMS responds with a setListErr error response.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives a setListError error response. The attribute is not replaced.

#### 14.7.5 MOC.NPAC.INV.DEL.serviceProvLRN

<b>Purpose</b>	This test case checks the LSMS's ability to respond to an invalid CMIP request. The NPAC SMS Simulator sends out an M-DELETE request intending to delete a nonexistent instance.
<b>Severity</b>	R
<b>Severity Explanation</b>	
<b>Prerequisites</b>	MOC.NPAC.CAP.OP.DEL.serviceProvLRN
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an invalid M-DELETE request for the nonexistent serviceProvLRN object.</li> <li>2. LSMS responds with a noSuchObjectInstance error response.</li> </ol>

<b>Expected Results</b>	The NPAC SMS Simulator receives a noSuchObjectInstance error response. Thus no instance is removed from the LSMS.
-------------------------	---

### 14.8 numberPoolBlock

<b>MO</b>	numberPoolBlock
<b>Purpose</b>	This section contains the test cases for the numberPoolBlock Managed Object Class pertaining to the NPAC SMS to LSMS Interface as part of the MOC testing of the NPAC SMS Simulator Interoperability Test.
<b>Prerequisite</b>	<ul style="list-style-type: none"> <li>A LSMS Network and Subscription Data Download Management association function is established.</li> <li>InpNPAC-SMS and InpSubscription Managed Object Instances exist.</li> </ul>

#### 14.8.1 MOC.NPAC.CAP.OP.CRE.numberPoolBlock

<b>Purpose</b>	Verify the LSMS's ability to respond correctly to an M-CREATE request for the numberPoolBlock managed object instance.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if LSMS is supporting numberPoolBlock objects.
<b>Prerequisites</b>	N/A
<b>Procedure</b>	<ol style="list-style-type: none"> <li>NPAC SMS Simulator issues a valid M-CREATE request for all attributes of the numberPoolBlock object.</li> <li>LSMS responds with a successful M-CREATE result containing all attributes.</li> </ol>
<b>Expected Results</b>	NPAC SMS Simulator receives a successful createResult and an instance is created on the LSMS.

#### 14.8.2 MOC.NPAC.CAP.OP.SET.numberPoolBlock

<b>Purpose</b>	Verify the LSMS's ability to respond correctly to an M-SET request for a numberPoolBlock managed object instance.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if LSMS is supporting numberPoolBlock objects.
<b>Prerequisites</b>	A numberPoolBlock exists on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>NPAC SMS Simulator issues a valid M-SET request for all attributes of the numberPoolBlock object.</li> <li>LSMS responds with a successful M-SET result</li> </ol>
<b>Expected Results</b>	NPAC SMS Simulator receives a successful setResult and the attributes are updated on the LSMS.

#### 14.8.3 MOC.NPAC.CAP.OP.GET.numberPoolBlock

<b>Purpose</b>	Verify the LSMS's ability to respond correctly to a scope and filtered M-GET request for all the attributes of the numberPoolBlock managed object instance. The filter contains an equality test for the numberPoolBlockNPA-NXX-X value.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if LSMS is supporting numberPoolBlock objects.
<b>Prerequisites</b>	A numberPoolBlock exists on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>NPAC SMS Simulator issues a valid M-GET request for all</li> </ol>

	<p>attributes of the numberPoolBlock object.</p> <p>2. LSMS responds with a successful M-GET result containing all attributes.</p>
<b>Expected Results</b>	NPAC SMS Simulator receives a getResult with all the attributes of the instance.

14.8.4 MOC.NPAC.CAP.OP.GET.MULTIPLE.numberPoolBlock

<b>Purpose</b>	Verify the LSMS's ability to respond correctly to a scope and filtered M-GET request for all the attributes of a range of numberPoolBlock managed object instances. The filter contains a range of numberPoolBlockNPA-NXX-X values.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if LSMS is supporting numberPoolBlock objects.
<b>Prerequisites</b>	Multiple numberPoolBlocks exist on the NPAC SMS Simulator.
<b>Procedure</b>	<p>1. NPAC SMS Simulator issues a valid M-GET request for all attributes of the numberPoolBlock object.</p> <p>2. LSMS responds with a series of linked M-GET replies containing all attributes.</p>
<b>Expected Results</b>	NPAC SMS Simulator receives the linked replies with all the attributes of the instance.

14.8.5 MOC.NPAC.CAP.OP.DEL.numberPoolBlock

<b>Purpose</b>	Verify the LSMS's ability to respond correctly to an M-DELETE request for the numberPoolBlock managed object instance.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if LSMS is supporting numberPoolBlock objects.
<b>Prerequisites</b>	A numberPoolBlock exists on the NPAC SMS Simulator.
<b>Procedure</b>	<p>1. NPAC SMS Simulator issues a valid M-DELETE for a numberPoolBlock object.</p> <p>2. LSMS responds with a successful M-DELETE result.</p>
<b>Expected Results</b>	NPAC SMS Simulator receives a successful deleteResult and the instance is removed on the LSMS.

14.8.6 MOC.NPAC.CAP.OP.SET.SING.numberPoolBlock

<b>Purpose</b>	Verify the LSMS's ability to respond correctly to an M-SET request for one attribute of a numberPoolBlock managed object instance.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if LSMS is supporting numberPoolBlock objects.
<b>Prerequisites</b>	A numberPoolBlock exists on the NPAC SMS Simulator.
<b>Procedure</b>	<p>1. NPAC SMS Simulator issues a valid M-SET request for a single attribute of the numberPoolBlock object.</p> <p>2. LSMS responds with a successful M-SET result</p>
<b>Expected Results</b>	NPAC SMS Simulator receives a successful setResult and the attribute is updated on the LSMS.

14.8.7 MOC.NPAC.CAP.OP.SET.MULT.numberPoolBlock

<b>Purpose</b>	Verify the LSMS's ability to respond correctly to an M-SET request
----------------	--

	for multiple attributes of a numberPoolBlock managed object instance.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if LSMS is supporting numberPoolBlock objects.
<b>Prerequisites</b>	A numberPoolBlock exists on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues a valid M-SET request for several attributes of the numberPoolBlock object.</li> <li>2. LSMS responds with a successful M-SET result.</li> </ol>
<b>Expected Results</b>	NPAC SMS Simulator receives a successful setResult and the attributes are updated on the LSMS.

14.8.8 MOC.NPAC.INV.CRE.numberPoolBlock

<b>Purpose</b>	Verify the LSMS's ability to respond correctly to an M-CREATE request that has an invalid attribute for a numberPoolBlock managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Required if LSMS is supporting numberPoolBlock objects.
<b>Prerequisites</b>	A numberPoolBlock exists on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues an M-CREATE with an invalid NPA-NXX-X for the numberPoolBlock object.</li> <li>2. LSMS responds with an invalidAttributeValue error or other appropriate error response.</li> </ol>
<b>Expected Results</b>	NPAC SMS Simulator receives the error response and no objects are created on the LSMS.

14.8.9 MOC.NPAC.INV.SET.numberPoolBlock

<b>Purpose</b>	Verify the LSMS's ability to respond correctly to an M-SET request that has an invalid attribute for a numberPoolBlock managed object instance.
<b>Severity</b>	O
<b>Severity Explanation</b>	Required if LSMS is supporting numberPoolBlock objects.
<b>Prerequisites</b>	A numberPoolBlock exists on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues an M-SET for the read-only numberPoolBlockId attribute of the numberPoolBlock object.</li> <li>2. LSMS responds with setListError error or other appropriate error response.</li> </ol>
<b>Expected Results</b>	NPAC SMS Simulator receives the error response and no attributes are updated on the LSMS.

14.8.10 MOC.NPAC.INV.DEL.numberPoolBlock

<b>Purpose</b>	Verify the LSMS's ability to respond correctly to an M-DELETE request for a numberPoolBlock managed object instance that does not exist.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if LSMS is supporting numberPoolBlock objects.
<b>Prerequisites</b>	N/A
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues an M-DELETE a numberPoolBlock object that does not exist on the LSMS.</li> </ol>



	2. LSMS responds with a noSuchObject error or other appropriate error response.
<b>Expected Results</b>	NPAC SMS Simulator receives the error response and no objects are deleted on the LSMS.

**14.9 serviceProvNPA-NXX-X**

<b>MO</b>	serviceProvNPA-NXX-X
<b>Purpose</b>	This section contains test cases for the serviceProvNPA-NXX-X Managed Object Class pertaining to the NPAC SMS manager to LSMS Interface, as part of the MO Conformance testing of the interoperability test. This capability test package checks the LSMS existence and basic validity of the specified capabilities. This object is used to support network data download to the LSMS.
<b>Prerequisite</b>	<ul style="list-style-type: none"> <li>• A NPAC Management association function is established.</li> <li>• The LSMS has successfully completed the S2S Interoperability testing.</li> <li>• A serviceProvNetwork exists on the LSMS.</li> </ul>

**14.9.1 MOC.NPAC.CAP.OP.CRE.serviceProvNPA-NXX-X**

<b>Purpose</b>	Verify the LSMS's ability to respond correctly to an M-CREATE request for the serviceProvNPA-NXX-X managed object instance.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if the LSMS is to supports network data download and the serviceProvNPA-NXX-X managed object.
<b>Prerequisites</b>	A serviceProvNetwork instance exists on the LSMS.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a serviceProvNPA-NXX-X M-CREATE request.</li> <li>2. LSMS responds successfully to the M-CREATE.</li> </ol>
<b>Expected Results</b>	NPAC SMS Simulator sends a valid M-CREATE request and receives the LSMS M-CREATE response indicating successful creation of the serviceProvNetwork.

**14.9.2 MOC.NPAC.CAP.OP.SET.serviceProvNPA-NXX-X**

<b>Purpose</b>	Verify the LSMS's ability to respond correctly to an M-SET request for the serviceProvNPA-NXX-X managed object instance.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if the LSMS is to supports network data download and the serviceProvNPA-NXX-X managed object.
<b>Prerequisites</b>	A serviceProvNetwork instance exists on the SOA.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a serviceProvNPA-NXX-X M-SET request.</li> <li>2. LSMS responds successfully to the M-SET.</li> </ol>
<b>Expected Results</b>	NPAC SMS Simulator sends a valid M-SET request and receives the LSMS M-SET response indicating successful modification of the serviceProvNetwork.

**14.9.3 MOC.NPAC.CAP.OP.DEL.serviceProvNPA-NXX-X**

<b>Purpose</b>	Verify the LSMS's ability to support the M-DELETE request for the
----------------	---

	serviceProvNPA-NXX-X managed object class.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if the LSMS supports network data download and the serviceProvNPA-NXX-X managed object.
<b>Prerequisites</b>	A serviceProvNPA-NXX-X exists on the LSMS.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an M-DELETE request for the serviceProvNPA-NXX-X managed object.</li> <li>2. LSMS responds successfully to the M-DELETE.</li> </ol>
<b>Expected Results</b>	NPAC SMS Simulator sends an M-DELETE request to the LSMS and the LSMS responds successfully.

#### 14.9.4 MOC.NPAC.INV.CRE.DUP.serviceProvNPA-NXX-X

<b>Purpose</b>	Verify the LSMS's ability to respond correctly to a duplicate M-CREATE request for the serviceProvNetwork managed object.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if the LSMS supports network data download and the serviceProvNPA-NXX-X managed object.
<b>Prerequisites</b>	A serviceProvNPA-NXX-X instance exists on the LSMS
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends a serviceProvNPA-NXX-X M-CREATE request for a serviceProvNetwork managed object that already exists.</li> <li>2. LSMS responds to the M-CREATE.</li> </ol>
<b>Expected Results</b>	NPAC SMS Simulator sends an M-CREATE request for an existing serviceProvNPA-NXX-X managed object and receives the LSMS M-CREATE error response of duplicateObjectInstanceEr.

#### 14.9.5 MOC.NPAC.INV.SET.serviceProvNPA-NXX-X

<b>Purpose</b>	Verify the LSMS's ability to handle correctly an M-SET request for a syntactically invalid CMIP request for the read-only attribute serviceProvNPA-NXX-X-ID.
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service.
<b>Prerequisites</b>	A numberPoolBlock exists on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends an M-SET request for a serviceProvNPA-NXX-X-ID.</li> <li>2. LSMS responds with an M-SET error.</li> </ol>
<b>Expected Results</b>	NPAC SMS Simulator receives an error response with the error type set to setListErrorEr.

#### 14.9.6 MOC.NPAC.INV.DEL.serviceProvNPA-NXX-X

<b>Purpose</b>	Verify the LSMS's ability to handle correctly an invalid CMIP M-DELETE request. NPAC SMS Simulator sends a delete for the serviceProvNPA-NXX-X managed object that does not exist.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if the LSMS supports network data download and the serviceProvNPA-NXX-X managed object.
<b>Prerequisites</b>	A numberPoolBlock exists on the NPAC SMS Simulator.

<b><i>Procedure</i></b>	1. NPAC SMS Simulator sends M-DELETE request for the serviceProvNPA-NXX-X managed object. 2. LSMS responds with an M-DELETE error.
<b><i>Expected Results</i></b>	NPAC SMS Simulator receives the error response with error type set to noSuchObjectInstanceEr.

## 15 Association Management Test Cases

### 15.1 Test Cases

#### 15.1.1 AMG.SOA.ASSOC.SAME and AMG.LSMS.ASSOC.SAME

<b>Purpose</b>	To verify that the SOA/LSMS retries the same NPAC SMS Simulator address after the initial association request is rejected with reason as RETRY-SAME-HOST.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service.
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• No association established between the SOA/LSMS and NPAC SMS Simulator.</li> <li>• System clocks synchronized to within 5 minutes.</li> <li>• Access Control attribute set according to Chapter 5 of NPAC SMS Interoperable Interface Specification.</li> </ul>
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA/LSMS issues association request (AARQ).</li> <li>2. NPAC SMS Simulator accepts association indication and issues an association response with errorCode = retry-same-host.</li> <li>3. SOA/LSMS receives response and issues association request (AARQ) with same address as before.</li> <li>4. NPAC SMS Simulator accepts association indication and sends association response.</li> <li>5. SOA/LSMS receives association confirmation.</li> </ol>
<b>Expected Results</b>	Association is successfully established with NPAC SMS Simulator the second time.

#### 15.1.2 AMG.SOA.ASSOC.OTHER and AMG.LSMS.ASSOC.OTHER

<b>Purpose</b>	To verify that the SOA/LSMS retries the backup NPAC SMS Simulator address after the initial association request is rejected with reason as RETRY-OTHER-HOST.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service.
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• No association established between the SOA/LSMS and NPAC SMS Simulator.</li> <li>• System clocks synchronized to within 5 minutes.</li> <li>• Access Control attribute set according to Chapter 5 of NPAC SMS Interoperable Interface Specification.</li> </ul>
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA/LSMS issues association request.</li> <li>2. NPAC SMS Simulator accepts association indication and issues an association response with errorCode = retry-other-host.</li> <li>3. SOA/LSMS issues association request to backup address of the NPAC SMS Simulator.</li> <li>4. NPAC SMS Simulator accepts association indication and sends association response.</li> <li>5. SOA/LSMS receives association confirmation.</li> </ol>
<b>Expected Results</b>	Association is successfully established with backup NPAC SMS

	Simulator.
--	------------

### 15.1.3 AMG.SOA.REQTMOT and AMG.LSMS.REQTMOT

<b>Purpose</b>	To verify that the SOA/LSMS times out a request after the configured retry interval when the NPAC SMS Simulator did not respond.
<b>Severity</b>	O
<b>Severity Explanation</b>	Current NPAC SMS guidelines do not suggest any retries.
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• An association is established between the SOA/LSMS and NPAC SMS Simulator.</li> <li>• Systems clocks are synchronized.</li> <li>• The MIB is populated with all the instance of the information model.</li> </ul>
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA/LSMS issues a CMIP request (M-GET on all attribute of a managed object instance).</li> <li>2. NPAC SMS Simulator does not respond to request.</li> <li>3. SOA/LSMS times out the request and reissues it after the configured retry interval.</li> <li>4. NPAC SMS Simulator accepts and responds to second request.</li> <li>5. SOA/LSMS receives response.</li> </ol>
<b>Expected Results</b>	First request times out. Second request is successful.

### 15.1.4 AMG.SOA.RETRY.CMIP and AMG.LSMS.RETRY.CMIP

<b>Purpose</b>	To verify that the SOA/LSMS retries a CMIP request for 3 times with a configured retry interval timeout between tries when the NPAC SMS Simulator does not respond. After the 3 <sup>rd</sup> attempt, the SOA/LSMS aborts the association.
<b>Severity</b>	O
<b>Severity Explanation</b>	Current NPAC SMS guidelines do not suggest any retries.
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• An association is established between the SOA/LSMS and NPAC SMS Simulator.</li> <li>• Systems clocks are synchronized.</li> <li>• The MIB is populated with all the instance of the information model.</li> </ul>
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA/LSMS issues a CMIP request (M-GET on all attribute of an managed object instance).</li> <li>2. NPAC SMS Simulator does not respond to request.</li> <li>3. SOA/LSMS times out the request and reissues it after the configured retry interval.</li> <li>4. NPAC SMS Simulator does not respond to request.</li> <li>5. SOA/LSMS times out the request and reissues it after the configured retry interval.</li> <li>6. NPAC SMS Simulator does not respond to request.</li> <li>7. The SOA/LSMS aborts the association and establishes a new one on which the request is tried again.</li> <li>8. SOA/LSMS receives response.</li> </ol>
<b>Expected Results</b>	SOA/LSMS successfully recovers from repeated request timeouts and the association abort.

15.1.5 AMG.SOA.RETRY.ASSOC and AMG.LSMS.RETRY.ASSOC

<b>Purpose</b>	To verify that the SOA/LSMS times out and retries when the NPAC SMS Simulator does not respond to an association request.
<b>Severity</b>	O
<b>Severity Explanation</b>	No requirements exist on how to troubleshoot an NPAC SMS connection. However, it is recommended that if an association cannot be established with the primary, the secondary NPAC SMS be attempted. If the secondary NPAC SMS replies with “retry-other-host”, then proceed to retry and troubleshoot the primary connection.
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• An association is not established between the SOA/LSMS and NPAC SMS Simulator.</li> <li>• Systems clocks are synchronized.</li> </ul>
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA/LSMS issues an A-associate request.</li> <li>2. NPAC SMS Simulator does not respond to request.</li> <li>3. SOA/LSMS times out the request after a configurable number of minutes and one of the following things has to be done: <ul style="list-style-type: none"> <li>• Automatically retries primary NPAC.</li> <li>• Automatically retries Backup NPAC.</li> </ul> </li> <li>4. Repeat the previous step until an association is established.</li> </ol>
<b>Expected Results</b>	SOA/LSMS recover from association request timeout.

15.1.6 AMG.SOA.SECVIOL and AMG.LSMS.SECVIOL

<b>Purpose</b>	To verify that the SOA/LSMS detects and recovers from security violations.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service.
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• An association is established between the SOA/LSMS and NPAC SMS Simulator.</li> <li>• Systems clocks are synchronized.</li> <li>• The MIB is populated with all the instance of the information model.</li> </ul>
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues an M-EVENT-REPORT with an invalid signature on a pre-established and active association.</li> <li>2. SOA/LSMS detects the security breach and aborts the association with no reason given.</li> <li>3. SOA/LSMS proceeds to re-establish an association with the NPAC SMS Simulator using either the same key or a different key.</li> <li>4. NPAC SMS Simulator accepts the new association and normal processing resumes.</li> </ol>
<b>Expected Results</b>	Compromised association is aborted and a new association is established with the same key or a different key.

15.1.7 AMG.SOA.LOSS and AMG.LSMS.LOSS

<b>Purpose</b>	To verify that the SOA/LSMS detects and recovers from loss of association.
----------------	--

<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service.
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• An association is established between the SOA/LSMS and NPAC SMS Simulator.</li> <li>• Systems clocks are synchronized.</li> </ul>
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. An established and active association between the NPAC SMS Simulator and SOA/LSM is manually torn down (i.e. temporarily disconnect the network connection).</li> <li>2. SOA/LSMS detects the loss of association.</li> <li>3. SOA/LSMS proceeds to re-establish an association with the NPAC SMS Simulator.</li> <li>4. NPAC SMS Simulator accepts new association and processing is resumed normally.</li> </ol>
<b>Expected Results</b>	Lost association is detected and a new association is established.

15.1.8 AMG.SOA.DOWN and AMG.LSMS.DOWN

<b>Purpose</b>	To verify that the SOA/LSMS detects and recovers from NPAC SMS Simulator going down.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service.
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• An association is established between the SOA/LSMS and NPAC SMS Simulator.</li> <li>• Systems clocks are synchronized.</li> </ul>
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The NPAC SMS Simulator is manually brought down.</li> <li>2. SOA/LSMS detects the NPAC SMS Simulator is down and proceeds to re-establish an association with the NPAC SMS Simulator.</li> <li>3. NPAC SMS Simulator accepts the association and normal processing is resumed.</li> </ol>
<b>Expected Results</b>	NPAC SMS Simulator failure is detected and an association is re-established.

## 16 App-to-App Test Cases

### 16.1 Audit Test Cases

<b>MO</b>	Audit Test Cases
<b>Purpose</b>	This section contains the test cases for the subscriptionAudit Managed Object Class pertaining to the Application processes of the SOA and LSMS to NPAC SMS Simulator Interface, as part of the Application to Application testing of the NPAC SMS Interoperability Test.
<b>Prerequisite</b>	All Managed ObjectConformance testing has been completed. Several subscriptionVersion (testing Service Provider and other Service Provider) have been created and network data has been downloaded into the testing LSMS.

#### 16.1.1 A2A.LSMS.VAL.MISSVER.subscriptionAudit

<b>Purpose</b>	To test the LSMS's ability to handle an NPAC SMS Simulator initiated subscription version audit for a missing version in the LSMS.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service.
<b>Prerequisites</b>	A subscription version exists on the NPAC SMS Simulator, but not the LSMS system.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator creates an audit for a subscription version that is currently missing on the LSMS.</li> <li>2. NPAC SMS issues the M-GET request to the LSMS for the subscriptionVersion object.</li> <li>3. LSMS returns an empty M-GET result indicating the object was not found.</li> <li>4. NPAC SMS performs the comparisons and issues the M-CREATE request to the LSMS for the missing subscriptionVersion.</li> <li>5. LSMS responds with a successful M-CREATE response.</li> </ol>
<b>Expected Results</b>	LSMS successfully handles the M-GET and M-CREATE requests.

#### 16.1.2 A2A.LSMS.VAL.OBSVER.subscriptionAudit

<b>Purpose</b>	To test the LSMS's ability to handle a NPAC SMS Simulator initiated subscription version audit, for an obsolete subscription version on the LSMS.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service.
<b>Prerequisites</b>	A subscription version exists on the LSMS, but not the NPAC SMS Simulator system.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator creates an audit for a subscription version that currently exists only on the LSMS.</li> <li>2. NPAC SMS issues the M-GET request to the LSMS for the subscriptionVersion object.</li> </ol>



	<ol style="list-style-type: none"> <li>3. LSMS returns the M-GET result containing all the attributes.</li> <li>4. NPAC SMS performs the comparisons and issues the M-DELETE request to the LSMS for the missing subscriptionVersion.</li> <li>5. LSMS responds with a successful M-DELETE response.</li> </ol>
<b>Expected Results</b>	LSMS successfully handle the NPAC SMS Simulator M-GET and M-DELETE.

### 16.1.3 A2A.LSMS.VAL.ERRVER.subscriptionAudit

<b>Purpose</b>	To test the LSMS's ability to handle a NPAC SMS Simulator initiated subscription version audit, for a version with a discrepancy in one attribute value on the LSMS.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service.
<b>Prerequisites</b>	A subscription version exists on the NPAC SMS that has a different attribute value than one on the LSMS.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator creates an audit for a subscription version that currently exists only on the LSMS.</li> <li>2. NPAC SMS issues the M-GET request to the LSMS for the subscriptionVersion object.</li> <li>3. LSMS returns the M-GET result containing all the attributes.</li> <li>4. NPAC SMS performs the comparisons and issues the M-SET request to the LSMS for the subscriptionVersion data.</li> <li>5. LSMS responds with a successful M-SET result.</li> </ol>
<b>Expected Results</b>	LSMS successfully handle the NPAC SMS Simulator initiated M-GET and M-SET.

### 16.1.4 A2A.SOA.VAL.NODIS.TN.subscriptionAudit

<b>Purpose</b>	To test the SOA's ability to initiate a subscription version audit on a single TN, for multiple attributes, and handle the subsequent NPAC SMS Simulator responses with no discrepancy found.
<b>Severity</b>	C
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. If Audits are implemented this test case must be run.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues a subscriptionAudit object creation request to the NPAC SMS Simulator, specifying a single TN and multiple attributes.</li> <li>2. NPAC SMS Simulator responds with a successful M-CREATE response.</li> <li>3. NPAC SMS Simulator issues the objectCreation notification to the SOA.</li> <li>4. SOA confirms the M-EVENT-REPORT.</li> <li>5. NPAC SMS Simulator performs the audit to the LSMSs and finds no discrepancies.</li> <li>6. NPAC SMS Simulator issues a subscriptionAuditResults</li> </ol>

	<p>notification.</p> <ol style="list-style-type: none"> <li>7. SOA confirms the M-EVENT-REPORT.</li> <li>8. NPAC SMS Simulator issues the objectDeletion notification for the subscriptionAudit object.</li> <li>9. SOA confirms the M-EVENT-REPORT.</li> <li>10. NPAC SMS Simulator issues an M-DELETE request and response to itself for the subscriptionAudit object. (This step may not happen immediately depending on the implementation. The production NPAC SMS issues this delete during a future housekeeping period.)</li> </ol>
<b>Expected Results</b>	SOA successfully initiates the audit and handles the subsequent interaction from NPAC SMS Simulator for this transaction.

16.1.5 A2A.SOA.VAL.NODIS.TNRNG.subscriptionAudit

<b>Purpose</b>	To test the SOA's ability to initiate a subscription version audit on a range of TNs, for a single attribute, and handle the subsequent NPAC SMS Simulator responses with no discrepancy found.
<b>Severity</b>	O
<b>Severity Explanation</b>	No impact on ability to provide service. If Audits are implemented, this functionality may be supported by auditing a single TN at a time.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues subscriptionAudit object creation request to NPAC SMS Simulator, specifying a range of TNs and a single attribute.</li> <li>2. NPAC SMS Simulator responds with a successful M-CREATE response.</li> <li>3. NPAC SMS Simulator issues the objectCreation notification to the SOA.</li> <li>4. SOA confirms the M-EVENT-REPORT.</li> <li>5. NPAC SMS Simulator performs the audit to the LSMSs and finds no discrepancies.</li> <li>6. NPAC SMS Simulator issues a subscriptionAuditResults notification.</li> <li>7. SOA confirms the M-EVENT-REPORT.</li> <li>8. NPAC SMS Simulator issues the objectDeletion notification for the subscriptionAudit object.</li> <li>9. SOA confirms the M-EVENT-REPORT.</li> <li>10. NPAC SMS Simulator issues an M-DELETE request and response to itself for the subscriptionAudit object. (This step may not happen immediately depending on the implementation. The production NPAC SMS issues this delete during a future housekeeping period.)</li> </ol>
<b>Expected Results</b>	SOA successfully initiate the audit and handle the subsequent interaction from NPAC SMS Simulator of this transaction.

16.1.6 A2A.SOA.VAL.NODIS.ACTRNG.subscriptionAudit

<b>Purpose</b>	To test the SOA's ability to initiate a subscription version audit for a range of TNs with a subscriptionAuditTN-ActivationRange, for a single attribute, and handle the subsequent NPAC SMS Simulator responses with no discrepancy found.
<b>Severity</b>	O
<b>Severity Explanation</b>	No impact on ability to provide service. If Audits are implemented, functionality may be supported by A2A.SOA.VAL.NODIS.TN.subscriptionAudit or A2A.SOA.VAL.NODIS.TNRNG.subscriptionAudit.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues subscriptionAudit object creation request to NPAC SMS Simulator, specifying a range of TNs, a single attribute and an activation range.</li> <li>2. NPAC SMS Simulator responds with a successful M-CREATE response.</li> <li>3. NPAC SMS Simulator issues the objectCreation notification to the SOA.</li> <li>4. SOA confirms the M-EVENT-REPORT.</li> <li>5. NPAC SMS Simulator performs the audit to the LSMSs and finds no discrepancies.</li> <li>6. NPAC SMS Simulator issues a subscriptionAuditResults notification.</li> <li>7. SOA confirms the M-EVENT-REPORT.</li> <li>8. NPAC SMS Simulator issues the objectDeletion notification for the subscriptionAudit object.</li> <li>9. SOA confirms the M-EVENT-REPORT.</li> <li>10. NPAC SMS Simulator issues an M-DELETE request and response to itself for the subscriptionAudit object. (This step may not happen immediately depending on the implementation. The production NPAC SMS issues this delete during a future housekeeping period.)</li> </ol>
<b>Expected Results</b>	SOA successfully initiates the audit and handles the subsequent interaction from the NPAC SMS Simulator for this transaction.

16.1.7 A2A.SOA.VAL.WITHDIS.TN.subscriptionAudit

<b>Purpose</b>	To test the SOA's ability to initiate a subscription version audit on a single TN, for all auditable attributes, and handle the subsequent discrepancy notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	No impact on ability to provide service. If Audits are implemented, functionality may be supported by auditing a single TN at a time.
<b>Prerequisites</b>	A2A.SOA.VAL.NODIS.TN.subscriptionAudit
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues subscriptionAudit object creation request to NPAC SMS Simulator, specifying a TN and all auditable attributes.</li> <li>2. NPAC SMS Simulator responds with a successful M-CREATE response.</li> <li>3. NPAC SMS Simulator issues the objectCreation notification to the SOA.</li> <li>4. SOA confirms the M-EVENT-REPORT.</li> </ol>

	<ol style="list-style-type: none"> <li>5. NPAC SMS Simulator performs the audit to the LSMSs and finds the discrepancies.</li> <li>6. For each discrepant subscriptionVersion, the NPAC SMS Simulator issues a subscriptionAuditResults notification.</li> <li>7. SOA confirms each subscriptionAuditResults notification received.</li> <li>8. NPAC SMS Simulator issues a subscriptionAuditResults notification.</li> <li>9. SOA confirms the M-EVENT-REPORT.</li> <li>10. NPAC SMS Simulator issues the objectDeletion notification for the subscriptionAudit object.</li> <li>11. SOA confirms the M-EVENT-REPORT.</li> <li>12. NPAC SMS Simulator issues an M-DELETE request and response to itself for the subscriptionAudit object. (This step may not happen immediately depending on the implementation. The production NPAC SMS issues this delete during a future housekeeping period.)</li> </ol>
<b>Expected Results</b>	SOA successfully initiate the audit and handles the subsequent interaction from NPAC SMS Simulator for this transaction.

#### 16.1.8 A2A.SOA.VAL.WITHDIS.TNRNG.subscriptionAudit

<b>Purpose</b>	To test the SOA's ability to initiate a subscription version audit on a TN range, for a single attribute, and handle the subsequent discrepancy notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	No impact on ability to provide service. If Audits are implemented, functionality may be supported by auditing a single TN at a time.
<b>Prerequisites</b>	A2A.SOA.VAL.NODIS.TNRNG.subscriptionAudit
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues subscriptionAudit object creation request to NPAC SMS Simulator, specifying a range of TNs and a single attribute.</li> <li>2. NPAC SMS Simulator responds with a successful M-CREATE response.</li> <li>3. NPAC SMS Simulator issues the objectCreation notification to the SOA.</li> <li>4. SOA confirms the M-EVENT-REPORT.</li> <li>5. NPAC SMS Simulator performs the audit to the LSMSs and finds the discrepant subscriptionVersions.</li> <li>6. For each discrepant subscriptionVersion, the NPAC SMS Simulator issues a subscriptionAuditResults notification.</li> <li>7. SOA confirms each subscriptionAuditResults notification received.</li> <li>8. NPAC SMS Simulator issues a subscriptionAuditResults notification.</li> <li>9. SOA confirms the M-EVENT-REPORT.</li> <li>10. NPAC SMS Simulator issues the objectDeletion notification for the subscriptionAudit object.</li> <li>11. SOA confirms the M-EVENT-REPORT.</li> <li>12. NPAC SMS Simulator issues an M-DELETE request and response to itself for the subscriptionAudit object. (This step may not happen immediately depending on the implementation. The production NPAC SMS issues this delete during a future</li> </ol>

	housekeeping period.)
<b>Expected Results</b>	SOA successfully initiates the audit and handles the subsequent interaction from NPAC SMS Simulator for this transaction.

16.1.9 A2A.SOA.VAL.WITHDIS.ACTRNG.subscriptionAudit

<b>Purpose</b>	To test the SOA's ability to initiate a subscription version audit for a range of TNs with an activation range, for multiple attributes, and handle the subsequent NPAC SMS Simulator discrepancy responses.
<b>Severity</b>	O
<b>Severity Explanation</b>	No impact on ability to provide service. If Audits are implemented, functionality may be implemented by A2A.SOA.VAL.WITHDIS.TN.subscriptionAudit or A2A.SOA.VAL.WITHDIS.TNRNG.subscriptionAudit.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues subscriptionAudit object creation request to NPAC SMS Simulator, specifying a range of TNs, multiple attributes and an activation range.</li> <li>2. NPAC SMS Simulator responds with a successful M-CREATE response.</li> <li>3. NPAC SMS Simulator issues the objectCreation notification to the SOA.</li> <li>4. SOA confirms the M-EVENT-REPORT.</li> <li>5. NPAC SMS Simulator performs the audit to the LSMSs and finds the discrepant subscriptionVersions.</li> <li>6. For each discrepant subscriptionVersion, the NPAC SMS Simulator issues a subscriptionAuditResults notification.</li> <li>7. SOA confirms each subscriptionAuditResults notification received.</li> <li>8. NPAC SMS Simulator issues a subscriptionAuditResults notification.</li> <li>9. SOA confirms the M-EVENT-REPORT.</li> <li>10. NPAC SMS Simulator issues the objectDeletion notification for the subscriptionAudit object.</li> <li>11. SOA confirms the M-EVENT-REPORT.</li> <li>12. NPAC SMS Simulator issues an M-DELETE request and response to itself for the subscriptionAudit object. (This step may not happen immediately depending on the implementation. The production NPAC SMS issues this delete during a future housekeeping period.)</li> </ol>
<b>Expected Results</b>	SOA successfully initiates the audit and handles the subsequent interaction from NPAC SMS Simulator for this transaction.

16.1.10 A2A.SOA.VAL.NPACCNCLD.subscriptionAudit

<b>Purpose</b>	To test the SOA's ability to initiate and handle a subscription version audit which is subsequently canceled by the NPAC SMS
----------------	--

	Simulator.
<b>Severity</b>	C
<b>Severity Explanation</b>	No impact on ability to provide service. Required if audit functionality is implemented.
<b>Prerequisites</b>	A2A.SOA.VAL.NODIS.TN.subscriptionAudit
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues subscriptionAudit object creation request to NPAC SMS Simulator.</li> <li>2. NPAC SMS Simulator responds with a successful M-CREATE response.</li> <li>3. NPAC SMS Simulator issues the objectCreation notification to the SOA.</li> <li>4. SOA confirms the M-EVENT-REPORT.</li> <li>5. NPAC SMS Simulator Personnel cancel the subscriptionAudit.</li> <li>6. NPAC SMS Simulator issues the objectDeletion notification for the subscriptionAudit object.</li> <li>7. SOA confirms the M-EVENT-REPORT.</li> <li>8. NPAC SMS Simulator issues an M-DELETE request and response to itself for the subscriptionAudit object. (This step may not happen immediately depending on the implementation. The production NPAC SMS issues this delete during a future housekeeping period.)</li> </ol>
<b>Expected Results</b>	SOA successfully initiates the audit and handles the subsequent interaction from the NPAC SMS Simulator.

16.1.11 A2A.SOA.INV.CRENOT.TIMOUT.subscriptionAudit

<b>Purpose</b>	To test the SOA's ability to initiate a subscriptionAudit and handle the condition that the subscriptionAudit object creation notification is not received.
<b>Severity</b>	O
<b>Severity Explanation</b>	No impact on ability to provide service. SOA may perform to verify error handling.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues subscriptionAudit object creation request to NPAC SMS Simulator.</li> <li>2. NPAC SMS Simulator responds with a successful M-CREATE response.</li> <li>3. NPAC SMS Simulator does NOT issue the objectCreation notification to the SOA.</li> <li>4. SOA re-issues subscriptionAudit object creation request to NPAC SMS Simulator.</li> <li>5. NPAC SMS Simulator responds with a duplicateManagedObjectInstance error response.</li> <li>6. SOA proceeds to successfully handle normal audit processing.</li> </ol>
<b>Expected Results</b>	The SOA detects the missing notification, re-issues the M-CREATE request, and handles the error response. Audit proceeds normally after second create.

16.1.12 A2A.SOA.VAL.WITHDIS.WSMSC.RANGE.subscriptionAudit

<b>Purpose</b>	To test the SOA's ability to initiate a subscription version audit for a range of TNs, and handles the subsequent NPAC SMS Simulator discrepancy responses for WSMSC data.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the SOA supports TN range audits and will be supporting WSMSC data.
<b>Prerequisites</b>	Active subscriptionVersionNPAC objects with WSMSC data discrepancies exist on the NPAC SMS simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues subscriptionAudit object creation request to NPAC SMS Simulator, specifying a range of TNs.</li> <li>2. NPAC SMS Simulator creates the subscriptionAudit object locally and responds to the M-CREATE.</li> <li>3. SOA confirms the objectCreation.</li> <li>4. NPAC SMS Simulator emulates performing the audit and issues a subscriptionAuditDiscrepancyRpt.</li> <li>5. SOA confirms the subscriptionAuditDiscrepancyRpt notification.</li> <li>6. NPAC SMS Simulator completes the audit and issues the subscriptionAuditResults notification.</li> <li>7. SOA handles the subscriptionAuditResults notification and responds with a confirmation.</li> <li>8. NPAC SMS Simulator issues the objectDeletion notification for the subscriptionAudit object.</li> <li>9. SOA handles the objectDeletion notification and responds with a confirmation.</li> </ol>
<b>Expected Results</b>	SOA successfully initiates the audit and handles the subsequent interaction from NPAC SMS Simulator of this transaction.

16.1.13 A2A.SOA.VAL.WITHDIS.WSMSC.SINGLE.subscriptionAudit

<b>Purpose</b>	To test the SOA's ability to initiate a subscription version audit by TN and handles the subsequent NPAC SMS Simulator discrepancy responses for WSMSC data discrepancies.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the SOA supports audits for a single TN and will be supporting WSMSC data.
<b>Prerequisites</b>	Active subscriptionVersionNPAC objects with WSMSC discrepancies exist on the NPAC SMS simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues a subscriptionAudit object creation request to the NPAC SMS Simulator, specifying a TN.</li> <li>2. NPAC SMS Simulator creates the object, responds to the request and issues the objectCreation notification.</li> <li>3. SOA handles the objectCreation notification and responds with confirmation</li> <li>4. NPAC SMS Simulator emulates performing the audit and issues a subscriptionAuditDiscrepancyRpt notification..</li> <li>5. SOA handles the subscriptionAuditDiscrepancyRpt notification, and responds with confirmation.</li> <li>6. NPAC SMS Simulator finishes the audit and sends the subscriptionAuditResults notification.</li> <li>7. SOA handles the subscriptionAuditResults notification, and responds with confirmation.</li> <li>8. NPAC SMS Simulator issues the objectDeletion notification for the subscriptionAudit object.</li> </ol>

	<p>9. SOA handles the objectDeletion notification and responds with confirmation.</p> <p>10. NPAC SMS deletes the subscriptionAudit object locally. (This step may not happen immediately depending on the implementation. The production NPAC SMS issues this delete during a future housekeeping period.)</p>
<b>Expected Results</b>	SOA successfully initiates the audit and handles the subsequent interaction from the NPAC SMS Simulator of this transaction.

16.1.14 A2A.SOA.VAL.WITHDIS.ASSOCSP.RANGE.subscriptionAudit

<b>Purpose</b>	To test the SOA's ability to initiate a subscription version audit for an associated service provider id in the SystemId, with an activation range, for multiple attributes, and handle the subsequent NPAC SMS Simulator discrepancy responses.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the SOA supports TN range audit processing for an associated service provider.
<b>Prerequisites</b>	An active subscriptionVersionNPAC object with discrepancies exists on the NPAC SMS simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues subscriptionAudit object creation request to NPAC SMS Simulator for an associated service provider, specifying activation range and multiple attributes, and handle the response message from the NPAC SMS Simulator.</li> <li>2. NPAC SMS Simulator creates the object, responds to the request and issues the objectCreation notification.</li> <li>3. SOA handles the objectCreation notification and responds with confirmation</li> <li>4. NPAC SMS Simulator emulates performing the audit and issues a subscriptionAuditDiscrepancyRpt notification..</li> <li>5. SOA handles the subscriptionAuditDiscrepancyRpt notification, and responds with confirmation.</li> <li>6. NPAC SMS Simulator finishes the audit and sends the subscriptionAuditResults notification.</li> <li>7. SOA handles the subscriptionAuditResults notification, and responds with confirmation.</li> <li>8. NPAC SMS Simulator issues the objectDeletion notification for the subscriptionAudit object.</li> <li>9. SOA handles the objectDeletion notification and responds with confirmation.</li> <li>10. NPAC SMS deletes the subscriptionAudit object locally. (This step may not happen immediately depending on the implementation. The production NPAC SMS issues this delete during a future housekeeping period.)</li> </ol>
<b>Expected Results</b>	SOA successfully initiates the audit and handles the subsequent transaction interaction from the NPAC SMS Simulator.

16.1.15 A2A.SOA.VAL.WITHDIS.ASSOCSP.SINGLE.subscriptionAudit

<b>Purpose</b>	To test the SOA's ability to initiate a subscription version audit for an associated service provider id in the SystemId, for a TN, for multiple attributes, and handle the subsequent NPAC SMS Simulator discrepancy responses.
----------------	--



<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the SOA supports audit processing for a single TN for an associated service provider.
<b>Prerequisites</b>	An active subscriptionVersionNPAC object with discrepancies exists on the NPAC SMS simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA issues subscriptionAudit object creation request to NPAC for an associated service provider, specifying a TN and multiple attributes, and handle the response message from the NPAC SMS Simulator.</li> <li>2. NPAC SMS Simulator creates the object, responds to the request and issues the objectCreation notification.</li> <li>3. SOA handles the objectCreation notification and responds with confirmation</li> <li>4. NPAC SMS Simulator emulates performing the audit and issues a subscriptionAuditDiscrepancyRpt notification..</li> <li>5. SOA handles the subscriptionAuditDiscrepancyRpt notification, and responds with confirmation.</li> <li>6. NPAC SMS Simulator finishes the audit and sends the subscriptionAuditResults notification.</li> <li>7. SOA handles the subscriptionAuditResults notification, and responds with confirmation.</li> <li>8. NPAC SMS Simulator issues the objectDeletion notification for the subscriptionAudit object.</li> <li>9. SOA handles the objectDeletion notification and responds with confirmation.</li> <li>10. NPAC SMS deletes the subscriptionAudit object locally. (This step may not happen immediately depending on the implementation. The production NPAC SMS issues this delete during a future housekeeping period.)</li> </ol>
<b>Expected Results</b>	SOA successfully initiates the audit and handles the subsequent transaction interaction from the NPAC SMS Simulator.

16.1.16 LSMS.VAL.MISSVER.subscriptionAudit.POOL

<b>Purpose</b>	To test the EDR LSMS's ability to handle an NPAC SMS Simulator initiated subscription version/number pool block audit for a missing number pool block in the EDR LSMS where the subscriptionLNPTType is equal to 'POOL'.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if the service provider is supporting number pool blocks.
<b>Prerequisites</b>	N/A
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator creates an audit for a subscription version/number pool block that is currently missing on the EDR LSMS.</li> <li>2. NPAC SMS issues the M-GET request to the LSMS for the numberPoolBlock object.</li> <li>3. LSMS returns an empty M-GET result indicating the object was not found.</li> <li>4. NPAC SMS issues the M-GET request to the LSMS for the subscriptionVersion object.</li> <li>5. LSMS returns an empty M-GET result indicating the object was not found.</li> <li>6. NPAC SMS performs the comparisons and issues the M-CREATE request to the LSMS for the missing</li> </ol>

	<p>numberPoolBlock (the missing subscriptionVersion object is correct behavior for EDR LSMS).</p> <p>7. LSMS responds with a successful M-CREATE response.</p>
<b>Expected Results</b>	LSMS successfully handles the M-GET and M-CREATE requests.

## 16.2 Service Provider and Network Data Test Cases

<b>MO</b>	Service Provider and Network Data Test Cases
<b>Purpose</b>	This section contains the test cases for Network Data, pertaining to the Application processes of the SOA and LSMS to NPAC SMS Simulator Interface, as part of the Application to Application testing of the NPAC SMS Interoperability Test.
<b>Prerequisite</b>	All Managed Object and stack to stack testing is completed. SOA, NPAC SMS Simulator and LSMS stacks and applications running.

### 16.2.1 A2A.LSMS.VAL.CREND.serviceProviderNPA-NXX

<b>Purpose</b>	Verify LSMS can create a new NPA-NXX for its own service provider network data.
<b>Severity</b>	C
<b>Severity Explanation</b>	No impact on ability to provide service. Must be performed if LSMS is managing network data.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS issues an M-CREATE request for a serviceProvNPA-NXX to the NPAC SMS Simulator to request that an NPA-NXX object be created.</li> <li>2. NPAC SMS Simulator issues a successful M-CREATE response.</li> <li>3. NPAC SMS issues an M-CREATE to the LSMS for the serviceProvNPA-NXX object.</li> <li>4. LSMS issues a successful M-CREATE response.</li> </ol>
<b>Expected Results</b>	serviceProvNPA-NXX object created on LSMS.

### 16.2.2 A2A.LSMS.VAL.DELND.serviceProviderNPA-NXX

<b>Purpose</b>	Verify LSMS can delete an NPA-NXX for its own service provider network data.
<b>Severity</b>	C
<b>Severity Explanation</b>	No impact on ability to provide service. Must be performed if LSMS is managing network data.
<b>Prerequisites</b>	serviceProvNPA-NXX already created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS issues a valid M-DELETE request for a serviceProvNPA-NXX to NPAC SMS Simulator to request that an NPA-NXX object be created for its network.</li> <li>2. NPAC SMS Simulator issues a successful M-DELETE response.</li> <li>3. LSMS receives M-DELETE request from NPAC SMS</li> </ol>

	<p>Simulator for the NPA-NXX object.</p> <p>4. LSMS responds with a successful M-DELETE response.</p>
<b>Expected Results</b>	serviceProvNPA-NXX object deleted on LSMS.

### 16.2.3 A2A.LSMS.VAL.CREND.serviceProviderLRN

<b>Purpose</b>	Verify LSMS can create a new LRN for its own service provider network data.
<b>Severity</b>	C
<b>Severity Explanation</b>	No impact on ability to provide service. Must be performed if LSMS is managing network data.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS issues a valid M-CREATE request for a serviceProvLRN to NPAC SMS Simulator to request that an LRN object be created for its network.</li> <li>2. NPAC SMS Simulator issues a successful M-CREATE response.</li> <li>3. LSMS receives M-CREATE request from NPAC SMS Simulator for the LRN object.</li> <li>4. LSMS responds with a successful M-CREATE response.</li> </ol>
<b>Expected Results</b>	serviceProvLRN object created on LSMS.

### 16.2.4 A2A.LSMS.VAL.DELND.serviceProviderLRN

<b>Purpose</b>	Verify LSMS can delete an LRN for its own service provider network data.
<b>Severity</b>	C
<b>Severity Explanation</b>	No impact on ability to provide service. Must be performed if LSMS is managing network data.
<b>Prerequisites</b>	serviceProvLRN already created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS issues a valid M-DELETE request for a serviceProvLRN to NPAC SMS Simulator to request that an LRN object be created for its network.</li> <li>2. NPAC SMS Simulator issues a successful M-DELETE response.</li> <li>3. LSMS receives M-DELETE request from NPAC SMS Simulator for the LRN object.</li> <li>4. LSMS responds with a successful M-DELETE response.</li> </ol>
<b>Expected Results</b>	serviceProvLRN object deleted on LSMS.

### 16.2.5 A2A.SOA.CAP.OP.SET.ASSOCSP.serviceProv

<b>Purpose</b>	To test the SOA 's ability to SET all of the mandatory attributes on a serviceProv MO instance for an associated service provider.
<b>Severity</b>	C

<b>Severity Explanation</b>	This test case must be executed if the service provider SOA is supporting service provider profile management for an associated service provider.
<b>Prerequisites</b>	A serviceProv MO instance has been created for an associated service provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues the M-SET serviceProv with an associated service provider is specified in the access control SystemId field and in the service provider id. The NPAC SMS Simulator handles the serviceProv M-SET and sends the M-SET response to the SOA.</li> <li>2. The SOA handles the M-SET response.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-SET request for the associated service provider and successfully handles the M-SET response from the NPAC SMS Simulator.

16.2.6 A2A.SOA.CAP.OP.GET.ASSOCSP.serviceProv

<b>Purpose</b>	To test the SOA 's ability to GET a serviceProv MO instance for an associated service provider.
<b>Severity</b>	O
<b>Severity Explanation</b>	This test case should be executed if the service provider SOA is supporting an M-Get of a service provider data for service provider profile management for an associated service provider. However, it does not impact the ability to perform LNP services.
<b>Prerequisites</b>	A serviceProv MO instance has been created for an associated service provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues the M-GET serviceProv with an associated service provider is specified in the access control SystemId field and in the service provider id. The NPAC SMS Simulator handles the serviceProv M-GET and sends the M-GET response to the SOA.</li> <li>2. The SOA handles the M-GET response.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-GET request and retrieves the attributes in the M-GET response from the NPAC SMS Simulator.

16.2.7 A2A.SOA.VAL.CREND.ASSOCSP.serviceProviderNPA-NXX

<b>Purpose</b>	Verify that the SOA, acting for an associated service provider, can perform a service provider NPA-NXX create.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if a SOA is to support associated service provider network data management.
<b>Prerequisites</b>	A serviceProvNetwork instance has been created for an associated service provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues the M-CREATE for the serviceProviderNPA-NXX with an associated service provider specified in the access control SystemId field and in the service provider id.</li> <li>2. The NPAC SMS Simulator handles the local serviceProviderNPA-NXX create for the associated service provider, and sends the M-CREATE response to the SOA.</li> </ol>
<b>Expected Results</b>	SOA issues a valid serviceProviderNPA-NXX M-CREATE and handles the M-CREATE response for an associated service provider.

16.2.8 A2A.SOA.VAL.DELND.ASSOCSP.serviceProviderNPA-NXX

<b>Purpose</b>	Verify that the SOA, acting for an associated service provider, can perform a delete of a service provider NPA-NXX.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if a SOA is to support associated service provider network data management.
<b>Prerequisites</b>	A serviceProvNetwork and serviceProviderNPA-NXX instance has been created for an associated service provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues the M-DELETE for the serviceProviderNPA-NXX with an associated service provider specified in the access control SystemId field and in the service provider id.</li> <li>2. The NPAC SMS Simulator handles the local serviceProviderNPA-NXX delete for the associated service provider, and sends the M-DELETE response to the SOA.</li> </ol>
<b>Expected Results</b>	SOA issues a valid serviceProviderNPA-NXX M-DELETE and handles the M-DELETE response for the associated service provider.

16.2.9 A2A.SOA.VAL.CREND.ASSOCSP.serviceProviderLRN

<b>Purpose</b>	Verify that the SOA, acting for an associated service provider, can perform a create for a service provider LRN.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if a SOA is to support associated service provider network data management.
<b>Prerequisites</b>	A serviceProvNetwork instance has been created for an associated service provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues the M-CREATE for the serviceProviderLRN with an associated service provider specified in the access control SystemId field and in the service provider id.</li> <li>2. The NPAC SMS Simulator handles the local serviceProviderLRN create for the associated service provider, and sends the M-CREATE response to the SOA.</li> </ol>
<b>Expected Results</b>	SOA issues a valid serviceProviderLRN M-CREATE and handles the M-CREATE response for the associated service provider.

16.2.10 A2A.SOA.VAL.DELND.ASSOCSP.serviceProviderLRN

<b>Purpose</b>	Verify that the SOA, acting for an associated service provider, can perform a delete of a service provider LRN.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if a SOA is to support associated service provider network data management.
<b>Prerequisites</b>	A serviceProvNetwork and serviceProviderLRN instance has been created for an associated service provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues the M-DELETE for the serviceProviderLRN with an associated service provider specified in the access control SystemId field and in the service provider id.</li> <li>2. The NPAC SMS Simulator handles the local serviceProviderLRN delete for the associated service provider, and sends the M-DELETE response to the SOA.</li> </ol>
<b>Expected Results</b>	SOA issues a valid serviceProviderLRN M-DELETE and handles the M-DELETE response for the associated service provider.

**16.3 Subscription Version Create Test Cases**

<b>MO</b>	Subscription Version Test Cases
<b>Purpose</b>	This section contains the A2A test cases for the Subscription Version flows listed in section 6.5 of IIS. The tests examine both the LSMS and the SOA applications. These test cases are part of the Application-to-Application testing of the NPAC SMS Interoperability Test.
<b>Prerequisite</b>	All prior testing phases, i.e. AMG, MOC, SEC, and S2S testing.

**16.3.1 A2A.NSOA.VAL.CREATE.TN-RANGE.SubscriptionVersion**

<b>Purpose</b>	Verify the New Service Provider SOA can perform a create for subscription versions on the NPAC SMS Simulator using a TN range. The Old Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if the SOA is supporting the creation of a range of subscription versions using the SubscriptionVersionNewSP-Create action.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA issues the M-ACTION SubscriptionVersionNewSP-Create to NPAC SMS Simulator using a TN-Range.</li> <li>2. The NPAC SMS Simulator creates the subscriptionVersionNPAC instances locally, and sends the M-ACTION response to the New Service Provider SOA.</li> <li>3. NPAC SMS Simulator issues an objectCreation or subscriptionVersionRangeObjectCreation notification for each subscription version created.</li> <li>4. The New Service Provider SOA confirms each objectCreation notification sent, which contains the ‘pending’ subscriptionVersionStatus for each of the newly created versions.</li> </ol>
<b>Expected Results</b>	The New Service Provider SOA successfully initiates the transaction and handles the subsequent interactions with the NPAC SMS Simulator. The created subscriptionVersionNPAC instances will have a subscriptionVersionStatus of ‘pending’.

**16.3.2 A2A.NSOA.VAL.CREATE.CONFLICT.SubscriptionVersion**

<b>Purpose</b>	Verify the New Service Provider SOA can handle a subscription version created in the ‘conflict’ state on the NPAC SMS Simulator. The Old Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The simulated Old Service Provider SOA locally creates a</li> </ol>

	<p>subscription version without providing authorization for the transfer of service.</p> <ol style="list-style-type: none"> <li>NPAC SMS Simulator sends the objectCreation or subscriptionVersionRangeObjectCreation notification to the New Service Provider SOA for the new subscriptionVersion with the subscriptionVersionStatus set to 'conflict'.</li> <li>The New Service Provider SOA confirms the objectCreation notification.</li> </ol>
<b>Expected Results</b>	The New Service Provider SOA successfully handles the interactions with the NPAC SMS Simulator. The created subscriptionVersionNPAC instance will have a status of 'conflict'.

### 16.3.3 A2A.OSOA.VAL.CREATE.TN-RANGE.SubscriptionVersion

<b>Purpose</b>	Verify the Old Service Provider SOA can perform a create for subscription versions on the NPAC SMS Simulator using a TN range. The New Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	O
<b>Severity Explanation</b>	Direct impact on ability to provide service. Requirements may be satisfied by MOC.SOA.CAP.ACT.subscriptionVersionOldSP-Create-Initial and MOC.SOA.CAP.ACT.subscriptionVersionOldSP-Create-Second.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>The Old Service Provider SOA issues the M-ACTION SubscriptionVersionOldSP-Create to the NPAC SMS Simulator using the TN-Range.</li> <li>The NPAC SMS Simulator creates the subscriptionVersionNPAC instances locally and sends the M-ACTION response to the Old Service Provider SOA.</li> <li>The NPAC SMS Simulator sends an objectCreation or subscriptionVersionRangeStatusAttributeValueChange notification for each subscription version created with the subscriptionVersionStatus set to 'pending'.</li> <li>The Old Service Provider SOA confirms each objectCreation notification sent.</li> </ol>
<b>Expected Results</b>	The Old Service Provider SOA successfully initiates the transaction and handles the subsequent interactions with the NPAC SMS Simulator. The created subscriptionVersionNPAC instances will have a subscriptionVersionStatus of 'pending'.

### 16.3.4 A2A.OSOA.VAL.NOCONC.ACTIVATE.SubscriptionVersion

<b>Purpose</b>	Verify the Old Service Provider SOA can handle the situation where a subscription version is activated by the new Service Provider SOA and the Old Service Provider SOA disregards both concurrence
----------------	---

	request notifications from the NPAC SMS Simulator. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service. Requirement exists to handle the initial and final concurrence requests.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The simulated New Service Provider SOA sends a subscriptionVersionNewSP-Create request for a subscription version to the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator issues the M-CREATE request and response and creates the subscriptionVersion locally.</li> <li>3. The NPAC SMS Simulator emits the object creation or subscriptionVersionRangeObjectCreation notification to the Old Service Provider SOA.</li> <li>4. The Old Service Provider SOA confirms the notification sent by the NPAC SMS Simulator.</li> <li>5. No response is received from Old Service Provider SOA regarding the newly created subscription version within “Initial Concurrence Window”.</li> <li>6. NPAC SMS Simulator sends M-EVENT-REPORT of subscriptionVersionOldSP-ConcurrenceRequest or subscriptionVersionRangeOldSP-ConcurrenceRequestnotification to the Old Service Provider SOA.</li> <li>7. The Old Service Provider SOA confirms the notification.</li> <li>8. Still no response from the Old Service Provider SOA regarding the newly created subscription version within the “Final Concurrence Window”.</li> <li>9. NPAC SMS Simulator sends M-EVENT-REPORT of subscriptionVersionOldSP-FinalConcurrenceWindowExpiration notification or subscriptionVersionRangeOldSP-FinalConcurrenceWindowExpiration to the Old Service Provider SOA.</li> <li>10. The Old Service Provider SOA confirms the notification.</li> <li>11. The simulated New Service Provider SOA activates the subscription version.</li> <li>12. NPAC SMS Simulator emulates sending M-CREATE request for the subscription version to all the Local LSMSSs, receives successful responses from each and updates the subscriptionVersionStatus to ‘active’.</li> <li>13. NPAC SMS Simulator sends a subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification with an ‘active’ status to the Old Service Provider SOA.</li> <li>14. Old Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The subscription version was successfully activated by the New Service Provider SOA in the absence of the Old Service Provider SOA’s concurrence.



16.3.5 A2A.OSOA.VAL.NOCONC.NOACTIVATE.SubscriptionVersion

<b>Purpose</b>	Verify the Old Service Provider SOA can handle the situation where a subscription version is to be activated by the new Service Provider SOA and the Old Service Provider SOA responds to the final concurrence notification by putting the subscriptionVersion into a status of 'conflict'. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The simulated New Service Provider SOA creates a subscription version on the NPAC SMS Simulator.</li> <li>2. NPAC SMS receives the create request and proceeds to issue an M-CREATE request and response to itself to create the subscriptionVersion locally.</li> <li>3. NPAC SMS Simulator sends the object creation or subscriptionVersionRangeObjectCreation notification to the Old Service Provider SOA.</li> <li>4. The Old Service Provider SOA confirms the object creation notification sent by the NPAC SMS Simulator.</li> <li>5. No response is received from Old Service Provider SOA regarding the newly created subscription version within the "Initial Concurrence Window".</li> <li>6. NPAC SMS Simulator sends M-EVENT-REPORT of subscriptionVersionOldSP-ConcurrenceRequest or subscriptionVersionRangeOldSP-ConcurrenceRequest notification to the Old Service Provider SOA.</li> <li>7. The Old Service Provider SOA confirms the notification.</li> <li>8. Still no response is received from the Old Service Provider SOA regarding the newly created subscription version in "Final Concurrence Window".</li> <li>9. NPAC SMS Simulator sends the subscriptionVersionOldSP-FinalConcurrenceWindowExpiration or subscriptionVersionRangeOldSP-FinalConcurrenceWindow notification to the Old Service Provider SOA.</li> <li>10. The Old Service Provider SOA confirms the notification.</li> <li>11. Before the subscription version is activated, the Old Service Provider SOA sends to the NPAC SMS Simulator a subscriptionVersionOldSP-Create request for the subscription version in which the subscriptionOldSP-Authorization is set to 'false' and the subscriptionStatusChangeCauseCode is provided.</li> <li>12. NPAC SMS Simulator sends the attributeChangeValue or subscriptionVersionRangeAttributeValueChange notification containing all the required attributes including the subscriptionOldSP-Authorization and subscriptionStatusChangeCauseCode to the Old Service Provider SOA</li> <li>13. Old Service Provider SOA confirms the notification.</li> <li>14. NPAC SMS Simulator sends the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification with the subscriptionVersionStatus set to 'conflict' to the Old Service Provider SOA.</li> </ol>

	15. Old Service Provider SOA confirms the notification.
<b>Expected Results</b>	The subscription version created by the New Service Provider SOA is put into conflict by the Old Service Provider SOA.

16.3.6 A2A.OSOA.VAL.CREATE.CONFLICT.SubscriptionVersion

<b>Purpose</b>	Verify the Old Service Provider SOA can create a subscription version with a status of ‘conflict’ on the NPAC SMS Simulator. The New Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The simulated New Service Provider SOA sends a create request for a subscription version on the NPAC SMS Simulator.</li> <li>2. NPAC SMS receives the create request and proceeds to issue an M-CREATE request and response to itself to create the subscriptionVersion locally.</li> <li>3. NPAC SMS Simulator sends the object creation or subscriptionVersionRangeObjectCreation notification to the Old Service Provider SOA.</li> <li>4. NPAC SMS Simulator emits the object creation notifications.</li> <li>5. The Old Service Provider SOA confirms the notification sent by the NPAC SMS Simulator, which contains the subscriptionStatus of ‘pending’ for the newly created version.</li> <li>6. The Old Service Provider SOA issues the subscriptionVersionOldSP-Create action with the subscriptionOldSP-Authorization set to False and a valid subscriptionStatusChangeCauseCode.</li> <li>7. NPAC SMS Simulator updates the subscriptionVersionStatus with the Old Service Provider’s data, sets the subscriptionVersionStatus to ‘conflict’ and responds to the M-ACTION.</li> <li>8. NPAC SMS Simulator issues the attributeValueChange or subscriptionVersionRangeAttributeValueChange notification with the subscriptionVersionStatus set to ‘conflict’.</li> <li>9. The Old Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The Old Service Provider SOA successfully handles the interactions with the NPAC SMS Simulator. The created subscriptionVersionNPAC will have a final status of ‘conflict’.

16.3.7 A2A.NSOA.VAL.CREATE.INTRA-SP-PORT.SubscriptionVersion

<b>Purpose</b>	Verify a SOA can create an Intra-Service Provider subscription version port. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if SOA is supporting intra-service provider ports.

<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The new service provider SOA issues the subscriptionVersionCreate action specifying an intra-service provider port by setting the subscriptionLnpType attribute to LISP.</li> <li>2. The NPAC simulator sends the M-ACTION response indicating the subscription version was successfully created.</li> <li>3. The NPAC simulator locally creates the 'pending' subscription version and emits the objectCreation or subscriptionVersionRangeObjectCreation notification.</li> <li>4. The SOA handles the notification sent by the NPAC simulator, and confirms it.</li> </ol>
<b>Expected Results</b>	The New SP SOA successfully initiates the transaction and handles the subsequent interactions with the NPAC simulator. The created subscriptionVersionNPAC instance will have a status of 'pending'.

### 16.3.8 A2A.DSOA.VAL.PORT-TO-ORIG.SubscriptionVersion

<b>Purpose</b>	To verify a New Service Provider SOA can perform a port-to-original. The Old Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct Impact on providing service.
<b>Prerequisites</b>	All subscription version create, activate and disconnect test cases.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA issues the subscriptionVersionCreate action specifying a port-to-original by setting the subscriptionPortingToOriginal-ServiceProviderSwitch attribute to 'true'.</li> <li>2. The NPAC SMS Simulator locally creates the 'pending' subscription version, responds to the M-ACTION and issues the objectCreation or subscriptionVersionRangeObjectCreation notification.</li> <li>3. The SOA confirms the objectCreation notification sent by the NPAC SMS Simulator.</li> <li>4. The SOA issues the M-ACTION subscriptionVersionActivate for the newly created and 'pending' subscriptionVersionNPAC instance.</li> <li>5. The NPAC SMS Simulator locally sets the subscriptionVersionStatus of the 'pending' instance to 'sending', and responds to the M-ACTION.</li> <li>6. The NPAC SMS Simulator emulates deleting the version from all the LSMSs, locally sets the subscriptionVersionStatus to 'old', and sends the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification to the SOA.</li> <li>7. 10. The New Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The SOA successfully initiates the transaction and handles the subsequent interactions with the NPAC SMS Simulator. The created subscriptionVersionNPAC instance will have a final status of 'old'.

16.3.9 A2A.NSOA.INV.MISS.INITIAL.CONC.SubscriptionVersion

<b>Purpose</b>	To verify the Old Service Provider SOA can detect and handle the situation where it receives the subscriptionVersionOldSP-FinalConcurrenceWindowExpiration notification without receiving the OldSP-ConcurrenceRequest notification first. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	O
<b>Severity Explanation</b>	Requirement does not exist. SOA may perform to validate error handling.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The simulated New Service Provider SOA sends a create request for a subscription version on the NPAC SMS Simulator.</li> <li>2. NPAC SMS Simulator receives the create request and proceeds to issue an M-CREATE request and response to itself to create the subscriptionVersion locally.</li> <li>3. The NPAC SMS Simulator issues the object creation or subscriptionVersionRangeObjectCreation notification to the Old Service Provider SOA.</li> <li>4. The Old Service Provider SOA confirms the notification sent by the NPAC SMS Simulator.</li> <li>5. No action is received from the Old Service Provider SOA regarding the new subscription version.</li> <li>6. Without sending subscriptionVersionOldSP-ConcurrenceRequest first, NPAC SMS Simulator sends M-EVENT-REPORT of subscriptionVersionOldSP-FinalConcurrenceWindowExpiration notification or subscriptionVersionRangeOldSP-FinalConcurrenceWindowExpiration to the Old Service Provider SOA.</li> <li>7. The Old Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The error is detected and logged.

16.3.10 A2A.NSOA.INV.STATE-TRANS.PEND-ACTIVE.SubscriptionVersion

<b>Purpose</b>	To verify the New Service Provider SOA can handle a state transition of a Subscription Version status from ‘pending’ to ‘active’ prior to the due date. The Old Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	O
<b>Severity Explanation</b>	Requirement does not exist. SOA may perform to validate error handling.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.subscriptionVersionNewSP-Create-Initial
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA issues the M-ACTION subscriptionVersionNewSP-Create to create a new subscription version on the NPAC SMS Simulator with a</li> </ol>

	<p>subscriptionVersionNewSP-DueDate set to several days into the future.</p> <ol style="list-style-type: none"> <li>2. NPAC SMS Simulator handles the local subscriptionVersionNPAC create, and sends the M-ACTION response to the New Service Provider SOA.</li> <li>3. NPAC SMS Simulator issues the objectCreation or subscriptionVersionRangeObjectCreation notification which contains the subscriptionVersionStatus set to 'pending' for the created version.</li> <li>4. The New Service Provider SOA confirms the notification.</li> <li>5. NPAC SMS Simulator issues a subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange M-EVENT-REPORT with a subscriptionVersionStatus set to 'active' prior to the due date.</li> <li>6. The New Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The New Service Provider SOA handles the notification.

16.3.11 A2A.NSOA.INV.STATE-TRANS.PEND-OLD.SubscriptionVersion

<b>Purpose</b>	To verify the New Service Provider SOA can handle an invalid state transition of a Subscription Version from Pending to Old. The Old Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	O
<b>Severity Explanation</b>	Requirement does not exist. SOA may run to verify error handling.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.subscriptionVersionNewSP-Create-Initial
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA issues the M-ACTION SubscriptionVersionNewSP-Create to the NPAC SMS Simulator.</li> <li>2. NPAC SMS Simulator handles the local subscriptionVersionNPAC create, and sends the M-ACTION response to the New Service Provider SOA.</li> <li>3. NPAC SMS Simulator issues the objectCreation or subscriptionVersionRangeObjectCreation notification which contains the subscriptionVersionStatus set to 'pending'.</li> <li>4. The New Service Provider SOA confirms the notification sent by the NPAC SMS Simulator.</li> <li>5. NPAC SMS Simulator issues a subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification with the subscriptionVersionStatus set to 'old'.</li> <li>6. The New Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The New Service Provider SOA handles the notification.

16.3.12 A2A.OSOA.INV.STATE-TRANS.PEND-OLD.SubscriptionVersion

<b>Purpose</b>	To verify that the Old Service Provider SOA can handle a state
----------------	--

	transition of a Subscription Version from 'pending' to 'old'. The New Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	O
<b>Severity Explanation</b>	Requirement does not exist. SOA may perform to verify error handling.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.subscriptionVersionOldSP-Create-Initial
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The Old Service Provider SOA issues the M-ACTION SubscriptionVersionOldSP-Create to the NPAC SMS Simulator.</li> <li>2. NPAC SMS Simulator handles the local subscriptionVersionNPAC create, and sends the M-ACTION response to the Old Service Provider SOA.</li> <li>3. NPAC SMS Simulator issues the objectCreation or subscriptionVersionRangeObjectCreation notification containing the subscriptionVersionStatus set to 'pending'.</li> <li>4. The Old Service Provider SOA confirms the notification.</li> <li>5. NPAC SMS Simulator issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification containing the subscriptionVersionStatus set to 'old'.</li> <li>6. The Old Service Provider SOA confirms the notification from the NPAC SMS Simulator.</li> </ol>
<b>Expected Results</b>	The Old Service Provider SOA handles the notification.

16.3.13 A2A.OSOA.INV.STATE-TRANS.PEND-FAILED.SubscriptionVersion

<b>Purpose</b>	To verify the Old Service Provider SOA can handle an invalid state transition of a Subscription Version from 'pending' to 'failed'. The New Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	O
<b>Severity Explanation</b>	Requirement does not exist. SOA may perform to verify error handling.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.subscriptionVersionOldSP-Create-Initial
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The Old Service Provider SOA issues the M-ACTION SubscriptionVersionOldSP-Create to the NPAC SMS Simulator.</li> <li>2. NPAC SMS Simulator handles the local subscriptionVersionNPAC create, and sends the M-ACTION response to the Old Service Provider SOA.</li> <li>3. NPAC SMS Simulator issues the objectCreation or subscriptionVersionRangeObjectCreation notification with the subscriptionVersionStatus set to 'pending'.</li> <li>4. The Old Service Provider SOA confirms the notification sent by the NPAC SMS Simulator.</li> <li>5. NPAC SMS Simulator issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange</li> </ol>

	notification with the subscriptionVersionStatus set to 'download-failed'. 6. The Old Service Provider SOA confirms the notification.
<b>Expected Results</b>	The Old Service Provider SOA handles the notification.

16.3.14 A2A.NSOA.INV.CREATE.ACTIVE.SubscriptionVersion

<b>Purpose</b>	To verify the New Service Provider SOA can handle the error condition where in response to a create request it receives an 'active' instead of a 'pending' subscription version status in the objectCreation notification. The Old SOA is simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	O
<b>Severity Explanation</b>	Requirement does not exist. May be performed to verify error handling.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.subscriptionVersionNewSP-Create-Initial
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA issues the M-ACTION SubscriptionVersionNewSP-Create.</li> <li>2. NPAC SMS Simulator handles the local subscriptionVersionNPAC create, and sends the M-ACTION response to the New Service Provider SOA.</li> <li>3. NPAC SMS Simulator issues the objectCreation or subscriptionVersionRangeObjectCreation notification which contains the subscriptionVersionStatus set to 'active' for the subscription version.</li> <li>4. The New Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The New Service Provider SOA handles the notification.

16.3.15 A2A.OSOA.INV.CREATE.SENDING.SubscriptionVersion

<b>Purpose</b>	To verify the New Service Provider SOA can handle the error condition where in response to a create request it receives a 'sending' instead of a 'pending' subscription version status in the objectCreation notification. The Old SOA is simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	O
<b>Severity Explanation</b>	Requirement does not exist. SOA may perform to validate error handling.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.subscriptionVersionOldSP-Create-Initial
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The Old Service Provider SOA issues the M-ACTION SubscriptionVersionOldSP-Create.</li> <li>2. NPAC SMS Simulator handles the local subscriptionVersionNPAC create, and sends the M-ACTION response to the Old Service Provider SOA.</li> <li>3. NPAC SMS Simulator issues the objectCreation or</li> </ol>

	<p>subscriptionVersionRangeObjectCreation notification with the subscriptionVersionStatus set to 'sending'.</p> <p>4. The Old Service Provider SOA confirms the objectCreation notification.</p>
<b>Expected Results</b>	The Old Service Provider SOA handles the notification.

16.3.16 A2A.NSOA.INV.OBJCRE.NOTMISS.SubscriptionVersion

<b>Purpose</b>	To verify the New Service Provider SOA can handle the error condition where it never receives the subscription version objectCreation notification in response to a create action request. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	O
<b>Severity Explanation</b>	Requirement does not exist. SOA may perform to verify error handling.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.subscriptionVersionNewSP-Create-Initial
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA issues the M-ACTION SubscriptionVersionNewSP-Create.</li> <li>2. NPAC SMS Simulator handles the local subscriptionVersionNPAC create and sends the M-ACTION response to the New Service Provider SOA.</li> <li>3. The NPAC SMS Simulator creates the version instance, but does not send the objectCreation or subscriptionVersionRangeObjectCreation notification to the New Service Provider SOA.</li> <li>4. The New Service Provider SOA issues an M-GET request for the subscriptionVersionNPAC instance which was supposed to be created.</li> <li>5. The NPAC SMS Simulator returns an M-GET response containing the newly created object.</li> <li>6. The New Service Provider SOA handles the M-GET result.</li> </ol>
<b>Expected Results</b>	The New Service Provider SOA detects the error and queries the NPAC SMS Simulator for the newly created version.

16.3.17 A2A.OSOA.INV.OBJCRE.NOTMISS.SubscriptionVersion

<b>Purpose</b>	To verify the Old Service Provider SOA can handle the error condition where it never receives the subscription version objectCreation notification in response to a create action request. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	O
<b>Severity Explanation</b>	Requirement does not exist. SOA may perform to verify error handling.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.subscriptionVersionOldSP-Create-Initial
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The Old Service Provider SOA issues the M-ACTION SubscriptionVersionOldSP-Create.</li> </ol>



	<ol style="list-style-type: none"> <li>2. NPAC SMS Simulator handles the local subscriptionVersionNPAC create and sends the M-ACTION response to the Old Service Provider SOA.</li> <li>3. The NPAC SMS Simulator creates the version instance, but does not send the objectCreation or subscriptionVersionRangeObjectCreation notification to the Old Service Provider SOA.</li> <li>4. The Old Service Provider SOA issues an M-GET request for the subscriptionVersionNPAC instance, which was supposed to be created.</li> <li>5. The NPAC SMS Simulator returns an M-GET response containing the newly created object.</li> <li>6. The Old Service Provider SOA handles the M-GET result.</li> </ol>
<b>Expected Results</b>	The Old Service Provider SOA detects the error and queries the NPAC SMS Simulator for the subscription version.

## 16.4 Subscription Version Activate Test Cases

### 16.4.1 A2A.NSOA.VAL.ACTIVATE.BYNPAC.SubscriptionVersion

<b>Purpose</b>	To verify the New Service Provider SOA can accept a subscription version activated by the NPAC SMS Simulator. The Old Service Provider SOA and LSMSs are simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service.
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• Successful completion of all CREATE test cases.</li> <li>• Subscription version exists in a state of ‘pending’.</li> </ul>
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The NPAC SMS Simulator activate the subscription version and sets subscriptionVersionStatus of the ‘pending’ instance to ‘sending’.</li> <li>2. The NPAC SMS Simulator emulates receiving positive M-CREATE response from all the LSMSs, locally sets the subscriptionVersionStatus of the ‘sending’ instance to ‘active’, and sends the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification to the New Service Provider SOA.</li> <li>3. The New Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The SOA successfully handles the interactions with the NPAC SMS Simulator resulting in an active version.

### 16.4.2 A2A.NSOA.VAL.ACTIVATE.SubscriptionVersion

<b>Purpose</b>	To verify the New Service Provider SOA can activate a subscription version in the pending state. The Old Service Provider SOA and the LSMSs are simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R

<b>Severity Explanation</b>	Direct impact on ability to provide service.
<b>Prerequisites</b>	Successful completion of all CREATE test cases.  'Pending' subscription version exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA issues the M-ACTION subscriptionVersionActivate for a 'pending' subscriptionVersionNPAC instance on the NPAC SMS Simulator.</li> <li>2. NPAC SMS Simulator sets the subscriptionVersionStatus to 'sending' and responds to the action.</li> <li>3. The NPAC SMS Simulator emulates receiving positive M-CREATE responses from all the LSMSs, locally sets the subscriptionVersionStatus of the 'sending' instance to 'active', and sends the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification to the New Service Provider SOA.</li> <li>4. The New Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	New Service Provider SOA successfully initiates the transaction and handles the subsequent interactions with the NPAC SMS Simulator.

#### 16.4.3 A2A.NSOA.VAL.ACTIVATE.FAIL.SubscriptionVersion

<b>Purpose</b>	To verify the New Service Provider SOA can handle an activation failure for a subscription version. The Old Service Provider SOA and the LSMSs are simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Impacts ability to provide service.
<b>Prerequisites</b>	A2A.NSOA.VAL.ACTIVATE.SubscriptionVersion  'Pending' subscription version exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA issues the M-ACTION subscriptionVersionActivate for a 'pending' subscriptionVersionNPAC instance on the NPAC SMS Simulator</li> <li>2. NPAC SMS Simulator updates the subscriptionVersionStatus to 'sending' and sends the action response.</li> <li>3. The NPAC SMS Simulator emulates receiving negative responses from all the LSMSs, locally sets the subscriptionVersionStatus to 'download-failed', and sends the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification to the New Service Provider SOA.</li> <li>4. The New Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	New Service Provider SOA successfully initiates the transaction and handles the subsequent interactions with the NPAC SMS Simulator.

16.4.4 A2A.NSOA.VAL.ACTIVATE.PARTFAIL.SubscriptionVersion

<b>Purpose</b>	To verify the New Service Provider SOA can handle a partial-failure activation for a subscription version. The Old Service Provider SOA and the LSMSs are simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Impacts ability to provide service
<b>Prerequisites</b>	A2A.NSOA.VAL.ACTIVATE.SubscriptionVersion  ‘Pending’ subscription version exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA issues the M-ACTION subscriptionVersionActivate for a pending subscriptionVersionNPAC instance on the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus to ‘sending’ and responds to the M-ACTION.</li> <li>3. The NPAC SMS Simulator emulates receiving a negative response from one LSMS, locally sets the subscriptionVersionStatus to ‘download-failed-partial’, updates the subscriptionFailedSP-List with the failed LSMS, and sends the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification to the New Service Provider SOA.</li> <li>4. The New Service Provider SOA confirms the subscriptionVersionStatusAttributeValueChange notification.</li> </ol>
<b>Expected Results</b>	New Service Provider SOA successfully initiates the transaction and handles the subsequent interactions with the NPAC SMS Simulator.

16.4.5 A2A.OSOA.VAL.ACTIVATE.SubscriptionVersion

<b>Purpose</b>	To verify the Old Service Provider SOA can handle the activation of a subscription version by the simulated New Service Provider SOA. The LSMSs are simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Impacts ability to provide service.
<b>Prerequisites</b>	Successful completion of all CREATE test cases.  ‘Pending’ subscription version exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The simulated New Service Provider SOA activates the pending subscription version.</li> <li>2. NPAC SMS Simulator locally sets the subscriptionVersionStatus to ‘sending’, and responds to the M-ACTION.</li> <li>3. The NPAC SMS Simulator emulates receiving positive responses from all the LSMSs for the subscriptionVersion M-CREATE, locally sets the subscriptionVersionStatus to ‘active’, and sends the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification to the Old Service Provider SOA.</li> </ol>

	4. The Old Service Provider SOA confirms the notification.
<b>Expected Results</b>	Old Service Provider SOA successfully handles the interactions with the NPAC SMS Simulator.

16.4.6 A2A.OSOA.VAL.ACTIVATE.FAIL.SubscriptionVersion

<b>Purpose</b>	To verify the Old Service Provider SOA can handle the activation failure of a subscription version by the simulated New Service Provider SOA. The LSMSs are simulated.
<b>Severity</b>	R
<b>Severity Explanation</b>	Impacts ability to provide service. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Prerequisites</b>	A2A.OSOA.VAL.ACTIVATE.SubscriptionVersion  ‘Pending’ subscription version exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The simulated New Service Provider SOA activates the pending subscription version.</li> <li>2. NPAC SMS Simulator locally sets the subscriptionVersionStatus to ‘sending’, and responds to the M-ACTION.</li> <li>3. The NPAC SMS Simulator emulates receiving positive responses from all the LSMSs for the subscriptionVersion M-CREATE, locally sets the subscriptionVersionStatus to ‘failed’ and sends the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification to the Old Service Provider SOA.</li> <li>4. The Old Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	Old Service Provider SOA successfully handles the interactions with the NPAC SMS Simulator.

16.4.7 A2A.OSOA.VAL.ACTIVATE.PARTFAIL.SubscriptionVersion

<b>Purpose</b>	To verify the Old Service Provider SOA can handle the partial-failure activation of a subscription version by the simulated New Service Provider SOA. The LSMSs are simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Impacts ability to provide service.
<b>Prerequisites</b>	A2A.OSOA.VAL.ACTIVATE.SubscriptionVersion  ‘Pending’ subscription version exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The simulated New Service Provider SOA activates the pending subscription version.</li> <li>2. NPAC SMS Simulator locally sets the subscriptionVersionStatus to ‘sending’, and responds to the M-ACTION.</li> <li>3. The NPAC SMS Simulator emulates receiving positive responses from all the LSMSs for the subscriptionVersion M-CREATE, locally sets the subscriptionVersionStatus to</li> </ol>

	<p>‘download-failed-partial’ and sends the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification to the Old Service Provider SOA.</p> <p>4. The Old Service Provider SOA confirms the notification.</p>
<b>Expected Results</b>	Old Service Provider SOA successfully handles the interactions with the NPAC SMS Simulator.

16.4.8 A2A.NSOA.ACTIVATE.ACTNOTMISS.SubscriptionVersion

<b>Purpose</b>	To verify the New Service Provider SOA can handle the error condition where the NPAC SMS Simulator never sends it the notification for a status change to active. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	O
<b>Severity Explanation</b>	SOA may perform to verify error handling.
<b>Prerequisites</b>	A2A.NSOA.VAL.ACTIVATE.SubscriptionVersion  Pending subscription version exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA issues the M-ACTION subscriptionVersionActivate for a ‘pending’ subscriptionVersionNPAC instance on the NPAC SMS Simulator.</li> <li>2. NPAC SMS Simulator update the subscriptionVersionStatus to ‘sending’ and sends the M-ACTION response.</li> <li>3. The NPAC SMS Simulator emulates receiving positive responses from all the LSMSSs, locally sets the subscriptionVersionStatus to ‘active’, but does not send the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification to the New Service Provider SOA.</li> <li>4. The New Service Provider SOA recognizes the error condition and issues an M-GET request for the status of the subscriptionVersionNPAC instance.</li> </ol>
<b>Expected Results</b>	The New Service Provider SOA detects the missing notification, and queries the NPAC SMS Simulator for the current status of the subscription version.

16.4.9 A2A.NSOA.INV.ACTIVATE.PARTFAIL.SubscriptionVersion

<b>Purpose</b>	To verify the New Service Provider SOA can handle the error condition where a partial-failure activation occurs and the NPAC SMS Simulator sends the notification for status change to ‘download-failed-partial’ with an empty Failed Service Provider List. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	O
<b>Severity Explanation</b>	Requirement does not exist. SOA may perform to validate error handling.
<b>Prerequisites</b>	A2A.NSOA.VAL.ACTIVATE.SubscriptionVersion

	'Pending' subscription version exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA issues the M-ACTION subscriptionVersionActivate for a 'pending' subscriptionVersionNPAC instance on the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus to 'sending' and responds to the M-ACTION request.</li> <li>3. The NPAC SMS Simulator emulates receiving a negative response from one LSMS locally sets the subscriptionVersionStatus to 'download-failed-partial', but does not update the failed Service Provider list with the failed LSMS, and sends the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification to the New Service Provider SOA.</li> <li>4. The New Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The New Service Provider SOA handles the error condition.

16.4.10 A2A.OSOA.INV.ACTIVATE.PARTFAIL.SubscriptionVersion

<b>Purpose</b>	To verify the Old Service Provider SOA can handle the error condition where a partial-failure activation occurs and the NPAC SMS Simulator sends it the notification for status change to partial-failed with an empty Failed Service Provider List. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	O
<b>Severity Explanation</b>	Requirement does not exist. SOA may perform to verify error handling.
<b>Prerequisites</b>	A2A.OSOA.VAL.ACTIVATE.SubscriptionVersion  'Pending' subscription version exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The simulated New Service Provider SOA sends an activation request to the NPAC SMS Simulator.</li> <li>2. NPAC SMS Simulator locally sets the subscriptionVersionStatus to 'sending' and responds to the activation request.</li> <li>3. The NPAC SMS Simulator emulates receiving a negative response from one LSMS, locally sets the subscriptionVersionStatus to 'download-failed-partial', but does not update the failed Service Provider list with the failed LSMS, and sends the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification to the Old Service Provider SOA.</li> <li>4. The Old Service Provider SOA confirms the notification and handles the error condition.</li> </ol>
<b>Expected Results</b>	The Old Service Provider SOA handles the notification.

16.4.11 A2A.NSOA.VAL.ACTIVATE.TN-RANGE.SubscriptionVersion

<b>Purpose</b>	To verify the New Service Provider SOA can activate a range of subscription versions in the pending state. The Old Service Provider SOA and the LSMSSs are simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	SOA must execute if supporting range activations using the subscriptionVersionActivate action.
<b>Prerequisites</b>	Successful completion of all CREATE test cases.  ‘Pending’ subscription versions exist.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA issues the M-ACTION subscriptionVersionActivate for a range of ‘pending’ subscriptionVersionNPAC instances on the NPAC SMS Simulator.</li> <li>2. NPAC SMS Simulator sets the subscriptionVersionStatuses to ‘sending’ and responds to the action.</li> <li>3. The NPAC SMS Simulator emulates receiving positive responses from all the LSMSSs, locally sets the subscriptionVersionStatus of the ‘sending’ instance to ‘active’, and sends the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification to the New Service Provider SOA.</li> <li>4. The New Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	New Service Provider SOA successfully initiates the transaction and handles the subsequent interactions with the NPAC SMS Simulator.

**16.5 Subscription Version Modify Test Cases**

16.5.1 A2A.NSOA.VAL.MODIFY.PEND.SubscriptionVersion

<b>Purpose</b>	To verify the New Service Provider SOA can modify a subscription version in the pending state using a Modify Action. The Old Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service.
<b>Prerequisites</b>	All version creation test cases.  ‘Pending’ subscription version exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA issues an M-ACTION subscriptionVersionModify to NPAC SMS Simulator.</li> <li>2. NPAC SMS Simulator locally sets the attribute values of the subscriptionVersionNPAC instance and responds to the M-ACTION request.</li> <li>3. NPAC SMS Simulator issues the attributeValueChange or subscriptionVersionRangeAttributeValueChange notification containing the updated attributes.</li> <li>4. The New Service Provider SOA confirms the notification.</li> </ol>

<b>Expected Results</b>	New Service Provider SOA successfully initiates the transaction and handles the subsequent interactions with the NPAC SMS Simulator. The subscriptionVersionNPAC attributes should be modified accordingly.
-------------------------	---

16.5.2 A2A.OSOA.VAL.MODIFY.PEND.SubscriptionVersion

<b>Purpose</b>	To verify the Old Service Provider SOA can modify a subscription version in the pending state using a Modify Action. The New Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service
<b>Prerequisites</b>	All version creation test cases.  ‘Pending’ subscription version exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The Old Service Provider SOA issues an M-ACTION subscriptionVersionModify to NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the attribute values of the subscriptionVersionNPAC instance and responds to the M-ACTION.</li> <li>3. NPAC SMS Simulator issues the attributeValueChange or subscriptionVersionRangeAttributeValueChange notification to the Old Service Provider SOA containing the updated attributes.</li> <li>4. The Old Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	Old Service Provider SOA successfully initiates the transaction and handles the subsequent interactions with the NPAC SMS Simulator. The subscriptionVersionNPAC attributes should be modified accordingly.

16.5.3 A2A.SOA.VAL.MODIFY.ACTIVE.SubscriptionVersion

<b>Purpose</b>	To test that a SOA can modify an active subscription version. The LSMSs are simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service.
<b>Prerequisites</b>	A subscription version had been created and activated.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues an M-ACTION subscriptionVersionModify to NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionNPAC attributes values for the instance to be modified, sets the instance’s subscriptionVersionStatus to ‘sending’ and responds to the M-ACTION.</li> <li>3. The NPAC SMS Simulator emulates receiving positive responses from all the LSMS’s, locally sets the instance’s subscriptionVersionStatus to ‘active’ and sends the subscriptionVersionStatusAttributeValueChange or</li> </ol>



	<p>subscriptionVersionRangeStatusAttributeValueChange notification to the SOA.</p> <p>4. The SOA confirms the notification.</p>
<b>Expected Results</b>	<p>The SOA successfully initiate the transaction and handles the subsequent interactions with the NPAC SMS Simulator. The subscriptionVersionNPAC attributes are modified.</p>

16.5.4 A2A.SOA.VAL.MODIFY.ACTIVE.TN-RANGE.SubscriptionVersion

<b>Purpose</b>	<p>Verify a SOA can modify active subscription versions on the NPAC SMS Simulator using a TN range. The LSMSs are simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.</p>
<b>Severity</b>	C
<b>Severity Explanation</b>	<p>SOA must execute if supporting range modifications using the subscriptionVersionModify action.</p>
<b>Prerequisites</b>	<p>A2A.SOA.VAL.MODIFY.ACTIVE.SubscriptionVersion</p> <p>A range of active subscription versions exists.</p>
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues an M-ACTION subscriptionVersionModify to NPAC SMS Simulator, using a TN-Range.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionNPAC attributes values for the instances to be modified, sets the instances’ subscriptionVersionStatus attributes to ‘sending’ and responds to the M-ACTION.</li> <li>3. The NPAC SMS Simulator emulates receiving positive responses from all the LSMS’s, locally sets the instances’ subscriptionVersionStatus attributes to ‘active’ and sends the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>4. The SOA confirms the notifications.</li> </ol>
<b>Expected Results</b>	<p>The SOA successfully initiate the transaction and handles the subsequent interactions with the NPAC SMS Simulator. The subscriptionVersionNPAC attributes for all the instances in the TN range are modified.</p>

16.5.5 A2A.SOA.VAL.MODIFY.BYNPAC.ACTIVE.SubscriptionVersion

<b>Purpose</b>	<p>To verify a SOA can handle subscription version modifications initiated by the NPAC SMS Simulator. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.</p>
<b>Severity</b>	R
<b>Severity Explanation</b>	<p>Direct impact on ability to provide service.</p>
<b>Prerequisites</b>	<p>A subscription version had been created and activated.</p>
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The NPAC SMS Simulator locally M-SETs the subscriptionVersionNPAC attributes values for the instance to be modified and the subscriptionVersionStatus to ‘sending’.</li> <li>2. The NPAC SMS Simulator emulates receiving positive responses from all the LSMSs and locally sets the instance’s</li> </ol>

	<p>subscriptionVersionStatus to 'active'.</p> <ol style="list-style-type: none"> <li>The NPAC SMS Simulator issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification with the subscriptionVersionStatus set to 'active'.</li> <li>The SOA confirms the subscriptionVersionStatusAttributeValueChange notification.</li> </ol>
<b>Expected Results</b>	New Service Provider SOA successfully handles the interactions with the NPAC SMS Simulator resulting from modifying the instance. The subscriptionVersionNPAC attributes are modified.

#### 16.5.6 A2A.SOA.VAL.MODIFY.PARTFAIL.SubscriptionVersion

<b>Purpose</b>	To verify a SOA can handle a modify action for an active subscription version where one LSMS fails the broadcast. The subscription version status will be set to 'active' and the Failed Service Provider List updated. The LSMSs are simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service.
<b>Prerequisites</b>	A subscription version had been created and activated.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>The SOA issues an M-ACTION subscriptionVersionModify to the NPAC SMS Simulator.</li> <li>The NPAC SMS Simulator locally sets the subscriptionVersionNPAC attributes values for the instance to be modified, updates the subscriptionVersionStatus to 'sending' and sends the M-ACTION response.</li> <li>The NPAC SMS Simulator emulates receiving positive responses from all but one LSMS, updates the Failed Service Provider List with the failed LSMS, locally sets the instance's subscriptionVersionStatus to 'active' and sends a subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>The SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The SOA successfully initiate the transaction and handles the subsequent interactions with the NPAC SMS Simulator. The subscriptionVersionNPAC attributes are modified.

#### 16.5.7 A2A.SOA.VAL.MODIFY.FAIL.SubscriptionVersion

<b>Purpose</b>	To verify a SOA can handle the condition where the final status of the version to be modified is set to 'active', and the subscriptionFailedSP-List is updated with all LSMSs in response to a failed modify active scenario. The LSMSs are simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service.
<b>Prerequisites</b>	A subscription version had been created and activated.

<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues an M-ACTION subscriptionVersionModify to the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionNPAC attributes values for the instance to be modified, updates the instance's subscriptionVersionStatus to 'sending' and sends the M-ACTION response.</li> <li>3. The NPAC SMS Simulator emulates receiving negative responses from all the LSMSs and locally sets the instance's subscriptionVersionStatus to 'active', updates the Failed Service Provider List attribute and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>4. The SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The SOA successfully initiates the transaction and handles the subsequent interactions with the NPAC SMS Simulator. The subscriptionVersionNPAC attributes are modified.

#### 16.5.8 A2A.SOA.INV.MODIFY.PARTFAIL.NOSPLIST.SubscriptionVersion

<b>Purpose</b>	To verify a SOA can handle the error condition where the final status of the subscription version to be modified is set to 'download-failed-partial', and the subscriptionFailedSP-List is not updated in response to a modify active version scenario. The LSMSs are simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	O
<b>Severity Explanation</b>	Requirement does not exist. SOA may perform to verify error handling.
<b>Prerequisites</b>	A subscription version had been created and activated.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues an M-ACTION subscriptionVersionModify to NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionNPAC attributes values for the instance to be modified, updates the instance's subscriptionVersionStatus to 'sending' and sends the M-ACTION response.</li> <li>3. The NPAC SMS Simulator emulates receiving a negative response from one LSMS, locally sets the instance's subscriptionVersionStatus to 'download-failed-partial', does not update the Failed Service Provider List attribute and sends the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>4. The SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The SOA handles the subscriptionVersionStatusAttributeValueChange notification.

#### 16.5.9 A2A.SOA.INV.MODIFY.ACTIVE.NOTMISS.SubscriptionVersion

<b>Purpose</b>	To verify a SOA can handle the error condition where the last
----------------	---

	notification of status change to active is not sent by the NPAC SMS Simulator in a modify active subscription version scenario. The LSMSs are simulated.
<b>Severity</b>	O
<b>Severity Explanation</b>	Requirement does not exist. SOA may perform to verify error handling.
<b>Prerequisites</b>	A subscription version had been created and activated.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues an M-ACTION subscriptionVersionModify to the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionNPAC attributes values for the instance to be modified, updates the instance's subscriptionVersionStatus to 'sending' and sends the M-ACTION response.</li> <li>3. The NPAC SMS Simulator emulates receiving positive responses from all the LSMS's and locally sets the instance's subscriptionVersionStatus to 'active', but does not send the corresponding notification.</li> <li>4. The SOA issues an M-GET request for the status of the subscription version and handles the response.</li> </ol>
<b>Expected Results</b>	The SOA detects the missing notification and queries for the status of the subscription version.

16.5.10 A2A.SOA.INV.MODIFY.ATTRCHNG.NOTMISS.SubscriptionVersion

<b>Purpose</b>	To verify a SOA can handle the error condition where in a modify of a pending subscription version scenario, the attributeValueChange notification is never emitted but the attributes of the pending subscription version instance are modified. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	O
<b>Severity Explanation</b>	Requirement does not exist. SOA may perform to validate error handling.
<b>Prerequisites</b>	A pending subscription version had been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues an M-ACTION subscriptionVersionModify to the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionNPAC attributes values for the instance to be modified and responds to the M-ACTION. No attributeValueChange or subscriptionVersionRangeAttributeValueChange notification is sent.</li> <li>3. The SOA issues an M-GET request for the attributes of the subscription version to be modified and handles the response.</li> </ol>
<b>Expected Results</b>	The SOA detects the missing notification and queries for the status of the version.

16.5.11 A2A.SOA.INV.MODIFY.ATTRSAME.NOTMISS.SubscriptionVersion

<b>Purpose</b>	To verify a SOA can handle the error condition where in a modify of a pending subscription version scenario, the
----------------	--

	attributeValueChange notification is never emitted and the attributes of the pending subscription version instance are not modified. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	O
<b>Severity Explanation</b>	Requirement does not exist. SOA may perform to validate error handling.
<b>Prerequisites</b>	A pending subscription version had been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues an M-ACTION subscriptionVersionModify to NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator responds to the M-ACTION, but does not set the subscriptionVersionNPAC attributes values for the instance to be modified, nor send the corresponding attributeValueChange or subscriptionVersionRangeAttributeValueChange notification to the SOA.</li> <li>3. The SOA issues an M-GET request for the attributes of the subscription version to be modified and handles the response which shows that the attributes are still the same.</li> <li>4. The SOA re-issues the M-ACTION subscriptionVersionModify to NPAC SMS Simulator.</li> <li>5. The NPAC SMS Simulator updates the attribute values locally, responds to the M-ACTION and sends the attributeValueChange notification.</li> <li>6. The SOA confirms the attributeValueChange notification.</li> </ol>
<b>Expected Results</b>	The SOA detects the missing notification and queries for the subscription version. Upon figuring out that the attribute did not change, the action is re-issued and processing completes normally.

16.5.12 A2A.SOA.VAL.MODIFY.PEND.TN-RANGE.SubscriptionVersion

<b>Purpose</b>	Verify a SOA can modify pending subscription versions on the NPAC SMS Simulator using a TN range. The LSMSs are simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	SOA must execute if supporting range modifications using the subscriptionVersionModify action.
<b>Prerequisites</b>	A2A.SOA.VAL.MODIFY.PEND.SubscriptionVersion A range of pending active subscription versions exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues an M-ACTION subscriptionVersionModify to NPAC SMS Simulator, using a TN-Range.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionNPAC attributes values for the instances to be modified and responds to the M-ACTION.</li> <li>3. NPAC SMS Simulator issues the attributeValueChange or subscriptionVersionRangeAttributeValueChange notification containing the updated attributes.</li> <li>4. The SOA confirms the notification(s).</li> </ol>
<b>Expected Results</b>	The SOA successfully initiate the transaction and handles the subsequent interactions with the NPAC SMS Simulator. The subscriptionVersionNPAC attributes for all the instances in the TN range are modified.

**16.6 Subscription Version Cancel Test Cases**

**16.6.1 A2A.SOA.VAL.CANCEL.SubscriptionVersion**

<b>Purpose</b>	To verify the SOA can initiate a cancel request of a pending subscription version. The Old Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Impacts ability to provide service.
<b>Prerequisites</b>	A non-concurrent pending subscription version has been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA issues a subscriptionVersionCancel M-ACTION request to the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus of the instance to be canceled to ‘canceled’, responds to the M-ACTION and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>3. The New Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The New Service Provider SOA successfully initiates the transaction and handles the subsequent interactions with the NPAC SMS Simulator. The created subscriptionVersionNPAC instance will have a status of canceled.

**16.6.2 A2A.NSOA.VAL.CANCEL.BYOSOASubscriptionVersion**

<b>Purpose</b>	To verify the New Service Provider SOA can handle the cancellation of a pending subscription version by the simulated Old Service Provider SOA. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Impacts ability to provide service.
<b>Prerequisites</b>	A concurrent, pending subscription version has been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The simulated Old Service Provider SOA issues a request to cancel a pending subscription version to the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus of the instance to be canceled to ‘cancel-pending’, responds to the Old Service Provider and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>3. The New Service Provider SOA confirms the notification.</li> <li>4. The New Service Provider SOA issues the subscriptionVersionNewSP-CancellationAcknowledge M-ACTION request.</li> </ol>

	<p>5. The NPAC SMS Simulator responds to the M-ACTION request from the New Service Provider SOA, emulates receiving the Old Service Provider SOA's cancellation acknowledge request, locally sets the subscriptionVersionStatus to 'canceled', and sends the subscriptionVersionStatusAttributeValueChanged or subscriptionVersionRangeStatusAttributeValueChanged notification.</p> <p>6. The New Service Provider SOA confirms the notification.</p>
<b>Expected Results</b>	The New Service Provider SOA successfully handles the interactions with the NPAC SMS Simulator. The pending subscriptionVersionNPAC instance will have a final status of canceled.

16.6.3 A2A.NSOA.VAL.CANCEL.TN-RANGE.SubscriptionVersion

<b>Purpose</b>	To verify the New Service Provider SOA can initiate a cancel request of pending subscription versions using a TN Range. The Old Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	The SOA must perform if ranges are part of their implementation.
<b>Prerequisites</b>	A2A.NSOA.VAL.CANCEL.SubscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA issues a subscriptionVersionCancel M-ACTION request to the NPAC SMS Simulator with a TN Range.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus of the instances to be canceled to 'cancel-pending', responds to the M-ACTION and issues the subscriptionVersionStatusAttributeValueChanged notifications.</li> <li>3. The New Service Provider SOA confirms the subscriptionVersionStatusAttributeValueChanged or subscriptionVersionRangeStatusAttributeValueChanged notifications.</li> <li>4. The New Service Provider SOA issues the subscriptionVersionNewSP-CancellationAcknowledge M-ACTION for a TN-Range or issues an action for each TN.</li> <li>5. The NPAC SMS Simulator responds to the New Service Provider SOA's action(s), emulates receiving the Old Service Provider SOA's cancellation acknowledge requests, locally sets the subscriptionVersionStatus attributes to 'canceled', and issues the subscriptionVersionStatusAttributeValueChanged or subscriptionVersionRangeStatusAttributeValueChanged notifications.</li> <li>6. The New Service Provider SOA confirms each notification received.</li> </ol>
<b>Expected Results</b>	The New Service Provider SOA successfully initiates the transaction and handles the subsequent interactions with the NPAC SMS Simulator. The subscriptionVersionNPAC instances will have a final status of 'canceled'.

16.6.4 A2A.OSOA.VAL.CANCEL.SubscriptionVersion

<b>Purpose</b>	To verify the Old Service Provider SOA can initiate a cancel request of a pending subscription version. The New Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Impacts ability to provide service.
<b>Prerequisites</b>	A pending subscription version has been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The Old Service Provider SOA issues a subscriptionVersionCancel M-ACTION request to the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus of the instance to be canceled to ‘cancel-pending’, responds to the M-ACTION and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>3. The Old Service Provider SOA confirms the notification.</li> <li>4. The Old Service Provider SOA issues the subscriptionVersionOldSP-CancellationAcknowledge M-ACTION request.</li> <li>5. The NPAC SMS Simulator responds to the action, emulates receiving the New Service Provider SOA’s cancellation acknowledge request, locally sets the subscriptionVersionStatus to ‘canceled’, and issues the subscriptionVersionStatusAttributeValueChange notification.</li> <li>6. The Old Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The Old Service Provider SOA successfully initiates the transaction and handles the subsequent interactions with the NPAC SMS Simulator. The created subscriptionVersionNPAC instance will have a final status of ‘canceled’.

16.6.5 A2A.OSOA.VAL.CANCEL.BYNSOA.SubscriptionVersion

<b>Purpose</b>	To verify the Old Service Provider SOA can handle the cancellation of a pending subscription version by the simulated New Service Provider SOA. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Impacts ability to provide service.
<b>Prerequisites</b>	A pending subscription version has been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The simulated New Service Provider SOA issues a request to cancel a pending subscription version to the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus of the instance to be canceled to ‘cancel-pending’, responds to the New Service Provider SOA request and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> </ol>



	<ol style="list-style-type: none"> <li>3. The Old Service Provider SOA confirms the notification.</li> <li>4. The Old Service Provider SOA issues the subscriptionVersionOldSP-CancellationAcknowledge M-ACTION request.</li> <li>5. The NPAC SMS Simulator responds to the Old Service Provider SOA's action, emulates receiving the New Service Provider SOA's cancellation acknowledge request, locally sets the subscriptionVersionStatus to 'canceled', and issues the subscriptionVersionStatusAttributeValueChanged or subscriptionVersionRangeStatusAttributeValueChanged notification.</li> <li>6. The Old Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The Old Service Provider SOA successfully handles the interactions with the NPAC SMS Simulator. The pending subscriptionVersionNPAC instance will have a status of 'canceled'.

#### 16.6.6 A2A.OSOA.VAL.CANCEL.TN-RANGE.SubscriptionVersion

<b>Purpose</b>	To verify the Old Service Provider SOA can initiate a cancel request of a set of pending subscription versions using a TN Range. The New Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if the SOA implementation supports TN Ranges.
<b>Prerequisites</b>	A2A.OSOA.VAL.CANCEL.SubscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The Old Service Provider SOA issues a subscriptionVersionCancel M-ACTION request to the NPAC SMS Simulator with a TN Range.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus of the instances to be canceled to 'cancel-pending', responds to the M-ACTION and issues the subscriptionVersionStatusAttributeValueChanged or subscriptionVersionRangeStatusAttributeValueChanged notifications.</li> <li>3. The Old Service Provider SOA confirms each notification received.</li> <li>4. The Old Service Provider SOA issues the subscriptionVersionOldSP-CancellationAcknowledge M-ACTION requests for each version or by TN-Range.</li> <li>5. The NPAC SMS Simulator responds to the M-ACTION, emulates receiving the New Service Provider SOA's cancellation acknowledge requests, locally sets the subscriptionVersionStatus attributes to 'canceled', and issues the subscriptionVersionStatusAttributeValueChanged or subscriptionVersionRangeStatusAttributeValueChanged notification(s).</li> <li>6. The Old Service Provider SOA confirms each notification received.</li> </ol>
<b>Expected Results</b>	The Old Service Provider SOA successfully initiates the transaction and handles the subsequent interactions with the NPAC SMS Simulator. The created subscriptionVersionNPAC instances will have the final status of 'canceled'.

16.6.7 A2A.OSOA.VAL.CANCEL.NOCONC.SubscriptionVersion

<b>Purpose</b>	To verify the Old Service Provider SOA can handle a subscription version going to the canceled state because the simulated New Service Provider SOA did not issue the second create within the concurrence window. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Impacts ability to provide service.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.subscriptionVersionOldSP-Create-Initial
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The Old Service Provider SOA issues the M-ACTION SubscriptionVersionOldSP-Create.</li> <li>2. the NPAC SMS Simulator creates the local subscriptionVersionNPAC object, sends the M-ACTION response to the Old Service Provider SOA and issues the objectCreatoror subscriptionVersionRangeObjectCreation notification.</li> <li>3. The Old Service Provider SOA confirms the notification sent by the NPAC SMS Simulator.</li> <li>4. The NPAC SMS Simulator emulates never receiving the second create from the New Service Provider SOA, locally sets the subscriptionVersionStatus to ‘canceled’, and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>5. The Old Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The Old Service Provider SOA successfully handles the interactions with the NPAC SMS Simulator. The created subscriptionVersionNPAC instance will have a final status of ‘canceled’.

16.6.8 A2A.NSOA.VAL.CANCEL.BYNPAC.SubscriptionVersion

<b>Purpose</b>	To verify the New Service Provider SOA can handle a subscription version being canceled by the NPAC SMS Simulator. The Old Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service.
<b>Prerequisites</b>	All subscription version create test cases are complete.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The NPAC SMS Simulator locally sets the subscriptionVersionStatus of an existing ‘pending’ version to ‘cancel-pending’, and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>2. The New Service Provider SOA confirms the notification.</li> <li>3. The New Service Provider SOA issues the</li> </ol>

	<p>subscriptionVersionNewSP-CancellationAcknowledge M-ACTION request.</p> <p>4. The NPAC SMS Simulator responds to the M-ACTION, emulates receiving the Old Service Provider SOA's cancellation acknowledge request, locally sets the subscriptionVersionStatus to 'canceled', and issues the subscriptionVersionStatusAttributeValueChanged or subscriptionVersionRangeStatusAttributeValueChanged notification.</p> <p>5. The New Service Provider SOA confirms the notification.</p>
<b>Expected Results</b>	The New Service Provider SOA successfully handles the interactions with the NPAC SMS Simulator. The created subscriptionVersionNPAC instance will have a final status of 'canceled'.

16.6.9 A2A.OSOA.VAL.CANCEL.BYNPAC.SubscriptionVersion

<b>Purpose</b>	To verify the Old Service Provider SOA can handle a subscription version being canceled by the NPAC SMS Simulator. The New Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service.
<b>Prerequisites</b>	All subscription version create test cases are complete.
<b>Procedure</b>	<p>1. The NPAC SMS Simulator locally sets the subscriptionVersionStatus of an existing subscription version from 'pending' to 'cancel-pending' and issues the subscriptionVersionStatusAttributeValueChanged or subscriptionVersionRangeStatusAttributeValueChanged notification.</p> <p>2. The Old Service Provider SOA confirms the notification.</p> <p>3. The Old Service Provider SOA issues the subscriptionVersionOldSP-CancellationAcknowledge M-ACTION request.</p> <p>4. The NPAC SMS Simulator responds to the M-ACTION, emulates receiving the New Service Provider SOA's cancellation acknowledge request, locally sets the subscriptionVersionStatus to 'canceled' and issues the subscriptionVersionStatusAttributeValueChanged or subscriptionVersionRangeStatusAttributeValueChanged notification.</p> <p>5. The Old Service Provider SOA confirms the notification.</p>
<b>Expected Results</b>	The Old Service Provider SOA successfully handles the interactions with the NPAC SMS Simulator. The created subscriptionVersionNPAC instance will have a final status of 'canceled'.

16.6.10 A2A.NSOA.VAL.CANCEL.ACKREQ.SubscriptionVersion

<b>Purpose</b>	Verify the New Service Provider SOA will respond to the
----------------	---

	notification subscriptionVersionCancellationAcknowledgeRequest with the action subscriptionVersionNewSP-CancellationAcknowledge. The Old Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Impacts ability to provide service
<b>Prerequisites</b>	A2A.NSOA.VAL.CANCEL.SubscriptionVersion A cancel-pending subscription version exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The NPAC SMS Simulator will request the New Service Provider SOA acknowledge an existing ‘cancel-pending’ subscription version by sending the notification for subscriptionVersionCancellationAcknowledgeRequest or subscriptionVersionRangeCancellationAcknowledgeRequest to the SOA.</li> <li>2. The New Service Provider SOA confirms the notification.</li> <li>3. The New Service Provider SOA sends the subscriptionVersionNewSP-CancellationAcknowledge action in response to the notification.</li> <li>4. The NPAC SMS Simulator responds to the M-ACTION, locally sets the version status to ‘canceled’ and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>5. The New Service Provider SOA confirms the subscriptionVersionStatusAttributeValueChange notification.</li> </ol>
<b>Expected Results</b>	The New Service Provider SOA successfully handles the interactions with the NPAC SMS Simulator. The created subscriptionVersionNPAC instance will have a status of ‘canceled’.

16.6.11 A2A.OSOA.VAL.CANCEL.ACKREQ.SubscriptionVersion

<b>Purpose</b>	Verify the Old Service Provider SOA will respond to the notification subscriptionVersionCancellationAcknowledgeRequest with the action subscriptionVersionOldSP-CancellationAcknowledge. The New Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Impacts ability to provide service.
<b>Prerequisites</b>	A2A.OSOA.VAL.CANCEL.SubscriptionVersion. A cancel-pending subscription version exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The NPAC SMS Simulator will request that the Old Service Provider SOA acknowledge an existing ‘cancel-pending’ subscription version by sending the notification for subscriptionVersionCancellationAcknowledgeRequest or subscriptionVersionRangeCancellationAcknowledgeRequest to the SOA.</li> <li>2. The Old Service Provider SOA confirms the notification.</li> <li>3. The Old Service Provider SOA issues the</li> </ol>

	<p>subscriptionVersionOldSP-CancellationAcknowledge action in response to the notification.</p> <p>4. The NPAC SMS Simulator responds to the action, locally sets the version status to 'canceled' and sends the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</p> <p>5. The Old Service Provider SOA confirms the notification.</p>
<b>Expected Results</b>	The Old Service Provider SOA successfully handles the interactions with the NPAC SMS Simulator. The created subscriptionVersionNPAC instance will have a status of 'canceled'.

16.6.12 A2A.NSOA.INV.CANCEL.CONFLICT.SubscriptionVersion

<b>Purpose</b>	Verify the New Service Provider SOA can handle the error situation where a subscription version in the cancel-pending state goes to conflict because the simulated Old Service Provider SOA does not acknowledge the cancellation request. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service.
<b>Prerequisites</b>	A pending subscription version has been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA issues a subscriptionVersionCancel M-ACTION request to the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus of the instance to be canceled to 'cancel-pending', responds to the M-ACTION and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>3. The New Service Provider SOA confirms the notification.</li> <li>4. The New Service Provider SOA issues the subscriptionVersionNewSP-CancellationAcknowledge M-ACTION request.</li> <li>5. The NPAC SMS Simulator responds to the M-ACTION, emulates not receiving the Old Service Provider SOA's cancellation acknowledge request, locally sets the subscriptionVersionStatus to 'conflict', and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>6. The New Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The New Service Provider SOA successfully initiates the transaction and handles the subsequent interactions with the NPAC SMS Simulator. The subscriptionVersionNPAC instance will have a final status of 'conflict'.

16.6.13 A2A.NSOA.VAL.CANCEL.CANCELED.SubscriptionVersion

<b>Purpose</b>	Verify the New Service Provider SOA can handle the situation
----------------	--

	where it cancels a subscription version and the simulated Old Service Provider SOA does not acknowledge the cancellation request. The NPAC SMS Simulator cancels the subscription version. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service.
<b>Prerequisites</b>	A pending subscription version has been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA issues a subscriptionVersionCancel M-ACTION request to the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus of the instance to be canceled to ‘cancel-pending’, responds to the M-ACTION, and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>3. The New Service Provider SOA confirms the notification.</li> <li>4. The New Service Provider SOA issues the subscriptionVersionNewSP-CancellationAcknowledge M-ACTION request.</li> <li>5. The NPAC SMS Simulator responds to the M-ACTION, emulates not receiving the Old Service Provider SOA’s cancellation acknowledge request, locally sets the subscriptionVersionStatus to ‘canceled’, and the subscriptionVersionStatusChangeCauseCode to “NPAC SMS Simulator automatic cancellation”, and issues the corresponding subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>6. The New Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The New Service Provider SOA successfully initiates the transaction and handles the subsequent interactions with the NPAC SMS Simulator. The subscriptionVersionNPAC instance will have a final status of ‘canceled’.

16.6.14 A2A.OSOA.VAL.CANCEL.CONFLICT.SubscriptionVersion

<b>Purpose</b>	Verify the Old Service Provider SOA can handle the situation where a subscription version in the cancel-pending state goes to conflict because the simulated New Service Provider SOA does not acknowledge the cancellation request. The attribute subscriptionStatusChangeCauseCode will be set accordingly. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct impact on ability to provide service.
<b>Prerequisites</b>	A pending subscription version has been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The Old Service Provider SOA issues a subscriptionVersionCancel M-ACTION request to the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus of the instance to be canceled to</li> </ol>

	<p>‘cancel-pending’, and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</p> <ol style="list-style-type: none"> <li>3. The Old Service Provider SOA confirms the notification.</li> <li>4. The Old Service Provider SOA issues the subscriptionVersionOldSP-CancellationAcknowledge M-ACTION request.</li> <li>5. The NPAC SMS Simulator responds to the M-ACTION, emulates not receiving the New Service Provider SOA’s cancellation acknowledge request, locally sets the subscriptionVersionStatus to ‘conflict’, the subscriptionStatusChangeCauseCode to ‘General Conflict’ and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>6. The Old Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The Old Service Provider SOA successfully initiates the transaction and handles the subsequent interactions with the NPAC SMS Simulator. The subscriptionVersionNPAC instance will have a final status of ‘conflict’.

16.6.15 A2A.NSOA.INV.CANCEL.PEND.SubscriptionVersion

<b>Purpose</b>	To verify the New Service Provider SOA can handle the error condition where in response to the subscriptionVersionCancel action for a pending version, the NPAC SMS Simulator sets the version status to ‘canceled’ immediately. The Old Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	O
<b>Severity Explanation</b>	Requirement does not exist. SOA may perform to verify error handling.
<b>Prerequisites</b>	MOC.SOA.CAP.ACT.subscriptionVersionNewSP-Create-Initial
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA issues the subscriptionVersionNewSP-Create M-ACTION request.</li> <li>2. The NPAC SMS Simulator creates the subscription version locally, sends the M-ACTION response to the New Service Provider SOA and issues the objectCreation or subscriptionVersionRangeObjectCreation notification.</li> <li>3. The New Service Provider SOA confirms the notification sent by the NPAC SMS Simulator.</li> <li>4. The simulated Old Service Provider SOA issues a local M-ACTION request to the NPAC SMS Simulator for subscriptionVersionOldSP-Create.</li> <li>5. The NPAC SMS Simulator subscriptionVersionNPAC instance, responds to the M-ACTION and issues the attributeValueChange or subscriptionVersionRangeAttributeValueChange notification.</li> <li>6. The New Service Provider SOA confirms the notification sent by the NPAC SMS Simulator.</li> <li>7. The New Service Provider SOA issues a subscriptionVersionCancel M-ACTION request to the NPAC</li> </ol>

	<p>SMS Simulator for the pending version.</p> <p>8. The NPAC SMS Simulator responds to the M-ACTION, skips setting the 'cancel-pending' status and acknowledgment requests, locally sets the subscriptionVersionStatus of the instance to 'canceled', and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</p> <p>9. The New Service Provider SOA confirms the notification.</p>
<b>Expected Results</b>	The New Service Provider SOA detects an error.

16.6.16 A2A.OSOA.INV.CANCEL.CONFLICT.SubscriptionVersion

<b>Purpose</b>	To verify the Old Service Provider SOA can handle the error condition where in response to the subscriptionVersionCancel action for a version in conflict, the NPAC SMS Simulator sets the version status to 'canceled' immediately. The New Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	O
<b>Severity Explanation</b>	Requirement does not exist. SOA may perform to verify error handling.
<b>Prerequisites</b>	A2A.OSOA.VAL.CREATE.CONFLICT.SubscriptionVersion  A subscription version exists in a state of 'conflict' for which both the old and new service providers have concurred.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The Old Service Provider SOA issues a subscriptionVersionCancel M-ACTION request for to NPAC SMS Simulator for a subscription version in conflict.</li> <li>2. The NPAC SMS Simulator responds to the M-ACTION, locally sets the subscriptionVersionStatus of the instance to be canceled to 'canceled', and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>3. The Old Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The Old Service Provider SOA detects the error.

16.6.17 A2A.NSOA.INV.CANCEL.ACTIVE.SubscriptionVersion

<b>Purpose</b>	Verify the New Service Provider SOA can handle the error condition where a version to be canceled becomes active. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	O
<b>Severity Explanation</b>	Requirement does not exist. SOA may perform to verify error handling.
<b>Prerequisites</b>	A pending subscription version has been created.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA issues a subscriptionVersionCancel M-ACTION request for to NPAC</li> </ol>



	<p>SMS Simulator.</p> <ol style="list-style-type: none"> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus of the instance to be canceled to 'cancel-pending', responds to the M-ACTION and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>3. The New Service Provider SOA confirms the notification.</li> <li>4. The New Service Provider SOA issues the subscriptionVersionNewSP-CancellationAcknowledge M-ACTION request.</li> <li>5. The NPAC SMS Simulator locally sets the subscriptionVersionStatus to 'active', responds to the M-ACTION request and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>6. The New Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The New Service Provider SOA detects the error.

## 16.7 Subscription Version Disconnect Test Cases

### 16.7.1 A2A.SOA.VAL.IMMDISC.SubscriptionVersion

<b>Purpose</b>	To test that a SOA can perform an immediate disconnect of an active subscription version.
<b>Severity</b>	C
<b>Severity Explanation</b>	Direct impact on providing service. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Prerequisites</b>	A subscription version had been created and activated.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues a subscriptionVersionDisconnect M-ACTION to the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus to 'sending' and responds to the M-ACTION.</li> <li>3. The NPAC SMS Simulator emulates broadcasting the M-DELETEs and receiving successful responses from all LSMSs, sets the subscriptionVersionStatus to 'old' and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>4. The SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The SOA successfully initiates the transaction and handles the subsequent interactions with the NPAC SMS Simulator. The subscriptionVersionNPAC instance has a final status of 'old'.

16.7.2 A2A.SOA.VAL.DEFDISC.SubscriptionVersion

<b>Purpose</b>	To verify a SOA can perform a deferred disconnect on an active subscription version. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	SOA must perform if implementing deferred disconnects.
<b>Prerequisites</b>	A subscription version had been created and activated.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues a subscriptionVersionDisconnect M-ACTION with the subscriptionEffectiveReleaseDate attribute set.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus to ‘disconnect-pending’, responds to the M-ACTION and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>3. The SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The SOA successfully initiates the transaction and handles the subsequent interactions with the NPAC SMS Simulator.

16.7.3 A2A.SOA.VAL.IMMDISC.BYNPAC.SubscriptionVersion

<b>Purpose</b>	To verify the SOA can handle an immediate disconnect of an active subscription version initiated by the NPAC SMS Simulator. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct Impact on providing service.
<b>Prerequisites</b>	A subscription version had been created and activated.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The NPAC SMS Simulator locally initiates a disconnect and sets the subscriptionVersionStatus to ‘sending’.</li> <li>2. The NPAC SMS Simulator emulates the broadcast and receiving successful responses from all LSMSs, locally sets the subscriptionVersionStatus to ‘old’ and issues a subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>3. The SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The SOA successfully handles interactions with the NPAC SMS Simulator. The subscriptionVersionNPAC instance has a final status of ‘old’.

16.7.4 A2A.SOA.VAL.IMMDISC.FAIL.SubscriptionVersion

<b>Purpose</b>	To verify a SOA can handle an immediate disconnect where all of the LSMSs fail the delete requests. The subscription version status will be set to ‘active’ and the Failed Service Provider List will contain all the LSMSs. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct Impact on providing service.

<b>Prerequisites</b>	A subscription version had been created and activated.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues a subscriptionVersionDisconnect M-ACTION to the NPAC.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus to 'sending' and responds to the M-ACTION.</li> <li>3. The NPAC SMS Simulator emulates the broadcast to the LSMSs which results in the subscriptionVersionStatus being set back to 'active' because of a full failure and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>4. The SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The SOA successfully initiates the transaction and handles the subsequent interactions with the NPAC SMS Simulator. The subscriptionVersionNPAC instance has a final status of 'active'.

16.7.5 A2A.SOA.VAL.IMMDISC.PARTFAIL.SubscriptionVersion

<b>Purpose</b>	To verify a SOA can handle the scenario where an immediate disconnect results in a partially-failed subscription version because one LSMS fails the broadcast. The LSMSs are simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct Impact on providing service.
<b>Prerequisites</b>	A subscription version had been created and activated.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues a subscriptionVersionDisconnect M-ACTION to the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus to 'sending' and responds to the M-ACTION.</li> <li>3. The NPAC SMS Simulator emulates receiving a negative response from one LSMS and updates the subscriptionFailedSP-List with its ID, locally sets the subscriptionVersionStatus to 'partial-download-failed', and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>4. The SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The SOA successfully initiates the transaction and handles the subsequent interactions with the NPAC SMS Simulator. The subscriptionVersionNPAC instance has a final status of 'partial-download-failed'.

16.7.6 A2A.SOA.VAL.IMMDISC.TN-RANGE.SubscriptionVersion

<b>Purpose</b>	Verify a SOA can perform an immediate disconnect of a range of subscription versions. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
----------------	--

<b>Severity</b>	C
<b>Severity Explanation</b>	Execute test case if SOA supports range disconnects.
<b>Prerequisites</b>	A range of subscription versions exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues an M-ACTION subscriptionVersionDisconnect to the NPAC.</li> <li>2. The NPAC SMS Simulator responds to the M-ACTION, emulates receiving the responses from all the LSMSs, locally sets the subscriptionVersionStatus to 'old', and emits the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>3. The SOA handles the notification and confirms it.</li> </ol>
<b>Expected Results</b>	The SOA successfully initiates the transaction and handles the subsequent interactions with the NPAC simulator. The subscriptionVersionNPAC instances have a status of 'old'.

16.7.7 A2A.SOA.INV.IMMDISC.ACT.OLD.SubscriptionVersion

<b>Purpose</b>	To verify a SOA can handle the scenario where an immediate disconnect results in the NPAC SMS Simulator setting the subscription version status to old immediately. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	O
<b>Severity Explanation</b>	Requirement does not exist. SOA may perform to verify error handling.
<b>Prerequisites</b>	A subscription version had been created and activated.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues a subscriptionVersionDisconnect M-ACTION to the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus to 'old', responds to the M-ACTION and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>3. The SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The SOA detects the error.

16.7.8 A2A.SOA.INV.IMMDISC.OLD.SubscriptionVersion

<b>Purpose</b>	To verify a SOA can handle the error condition where for an immediate disconnect, the final status is set to 'old' but the Failed Service Provider List contains the names of all the LSMSs. The LSMSs are simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	O
<b>Severity Explanation</b>	Requirement does not exist. SOA may perform to verify error handling.
<b>Prerequisites</b>	A subscription version had been created and activated.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues a subscriptionVersionDisconnect M-ACTION to the NPAC SMS Simulator.</li> </ol>

	<ol style="list-style-type: none"> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus to 'sending' and responds to the M-ACTION response.</li> <li>3. The NPAC SMS Simulator emulates receiving negative responses from all the LSMSs, locally sets the subscriptionVersionStatus to 'old' instead of 'failed', updates the Failed Service Provider List with all LSMSs, and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>4. The SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The SOA detects the error.

### 16.7.9 A2A.SOA.INV.IMMDISC.FAILED.SubscriptionVersion

<b>Purpose</b>	To verify a SOA can handle the error condition where on an immediate disconnect, the final status is set to 'failed' but the Failed Service Provider List contains the name of only one LSMS. The LSMSs are simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	O
<b>Severity Explanation</b>	Requirement does not exist. SOA may perform to verify error handling.
<b>Prerequisites</b>	A subscription version had been created and activated.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues a subscriptionVersionDisconnect M-ACTION to the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus to 'sending' and responds to the M-ACTION.</li> <li>3. The NPAC SMS Simulator emulates receiving a negative response from one LSMS locally sets the subscriptionVersionStatus to 'failed' instead of 'download-failed-partial', updates the Failed Service Provider List with the failed LSMS, and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>4. The SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The SOA detects the error.

### 16.7.10 A2A.SOA.INV.IMMDISC.OLD.FAILService Provider.SubscriptionVersion

<b>Purpose</b>	To verify a SOA can handle the error condition where on an immediate disconnect, the final status is set to 'old' and the Failed Service Provider List contains the name of one LSMS. The LSMSs are simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	O
<b>Severity Explanation</b>	Requirement does not exist.
<b>Prerequisites</b>	A subscription version had been created and activated.

<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues a subscriptionVersionDisconnect M-ACTION to the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus to 'sending', and responds to the M-ACTION.</li> <li>3. The NPAC SMS Simulator emulates receiving a negative response from one LSMS, locally sets the subscriptionVersionStatus to 'old' instead of 'download-failed-partial', updates the Failed Service Provider List with the failed LSMS, and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>4. The SOA handles the notification.</li> </ol>
<b>Expected Results</b>	The SOA detects the error.

16.7.11 A2A.SOA.VAL.CANCEL.DISCPEND.SubscriptionVersion

<b>Purpose</b>	To verify a New Service Provider SOA can handle the subscription version cancellation scenario, where a disconnect-pending version is canceled. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Direct Impact on providing service. SOA must perform if deferred disconnect is supported.
<b>Prerequisites</b>	A disconnect-pending subscription version exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA issues the subscriptionVersionCancel M-ACTION request for the disconnect-pending version.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus of the instance to 'active', responds to the M-ACTION and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>3. The New Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The New Service Provider SOA initiates the transaction, and handles the interactions with the simulator successfully. The final status of the subscription version is 'active'.

**16.8 Subscription Version Conflict Test Cases**

16.8.1 A2A.NSOA.VAL.CONFLICT.RESOLV.SubscriptionVersion

<b>Purpose</b>	To verify the New Service Provider SOA can handle the resolution of a subscription version conflict. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct Impact on providing service.

<b>Prerequisites</b>	A subscription version with the subscription version status equal to 'conflict' exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The NPAC SMS Simulator initiates the resolution of the conflict by locally setting the subscriptionVersionStatus to 'pending' and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>2. The New Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The New Service Provider SOA handles the interactions with the NPAC SMS Simulator successfully. The subscription version status is now 'pending'.

16.8.2 A2A.NSOA.VAL.CONFLICT.RESOLV.BYNSOA.SubscriptionVersion

<b>Purpose</b>	To verify the New Service Provider SOA can resolve a subscription version in conflict. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct Impact on providing service.
<b>Prerequisites</b>	A subscription version with subscription version status equal to 'conflict' exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA initiates the resolution of the conflict state by sending a subscriptionVersionRemoveFromConflict M-ACTION request to the simulator.</li> <li>2. The NPAC SMS Simulator verifies the six hours tunable is not being violated, removes the conflict by locally setting the subscriptionVersionStatus to 'pending', responds to the M-ACTION and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>3. The New Service Provider SOA confirms the notification.</li> </ol>
<b>Expected Results</b>	The New Service Provider SOA handles the interactions with the NPAC SMS Simulator successfully. The subscription version status is now 'pending'.

16.8.3 A2A.OSOA.VAL.CONFLICT.RESOLV.SubscriptionVersion

<b>Purpose</b>	To verify the Old Service Provider SOA can handle the resolution of a subscription version conflict. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct Impact on providing service.
<b>Prerequisites</b>	A subscription version with the subscription version status equal to 'conflict' exists.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The NPAC SMS Simulator initiates the resolution of the conflict state by locally setting the subscriptionVersionStatus to</li> </ol>

	<p>‘pending’, and issues the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</p> <p>2. The Old Service Provider SOA confirms the notification sent by the simulator.</p>
<b>Expected Results</b>	The Old Service Provider SOA handles the interactions with the NPAC SMS Simulator successfully. The final subscription version status is ‘pending’.

#### 16.8.4 A2A.OSOA.VAL.CONFLICT.RESOLV.BYOSOA.SubscriptionVersion

<b>Purpose</b>	To verify the Old Service Provider SOA can handle the resolution of a subscription version conflict. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Direct Impact on providing service. SOA must perform if using an M-SET.
<b>Prerequisites</b>	A2A.OSOA.VAL.CREATE.CONFLICT.SubscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The Old Service Provider SOA issues the subscriptionVersionOldSP-Create action with the subscriptionOldSP-Authorization set to ‘false’ and provides a subscriptionStatusChangeCauseCode value.</li> <li>2. The NPAC SMS Simulator creates the subscription version with a subscriptionVersionStatus of ‘conflict’, responds to the M-ACTION and issues the object creation or subscriptionVersionRangeObjectCreation notification.</li> <li>3. The Old Service Provider SOA confirms the notification.</li> <li>4. The Old Service Provider SOA initiates the resolution of the conflict state by sending an M-SET request with the subscriptionOldSP-Authorization set to ‘true’.</li> <li>5. The NPAC SMS Simulator executes the M-SET request, sets the subscriptionVersionStatus to ‘pending’, sends the M-SET response, and issues the attributeValueChange or subscriptionVersionRangeAttributeValueChange notification and the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification to the Old Service Provider SOA.</li> <li>6. The Old Service Provider SOA confirms both the notifications sent by the simulator.</li> </ol>
<b>Expected Results</b>	The Old Service Provider SOA handles the interactions with the NPAC SMS Simulator successfully. The subscription version status is now ‘pending’.

#### 16.8.5 A2A.NSOA.VAL.CONFLICT.RESOLV.TN-RANGE.BYNSOA.SubscriptionVersion

<b>Purpose</b>	To verify the New Service Provider SOA can resolve a range of subscription versions in conflict. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	C



<b>Severity Explanation</b>	SOA must execute if supporting range conflict resolution using the subscriptionVersionRemoveFromConflict action.
<b>Prerequisites</b>	2 or more subscription versions with subscription version status equal to 'conflict' exist.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA initiates the resolution of the conflict state by sending a subscriptionVersionRemoveFromConflict M-ACTION request to the simulator for a range of subscription versions.</li> <li>2. The NPAC SMS Simulator verifies the six hours tunable is not being violated, removes the conflict by locally setting the subscriptionVersionStatus to 'pending', responds to the M-ACTION and issues either the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>3. The New Service Provider SOA confirms the notification(s).</li> </ol>
<b>Expected Results</b>	The New Service Provider SOA handles the interactions with the NPAC SMS Simulator successfully. The subscription version status is now 'pending'.

## 16.9 LSMS Test Cases

### 16.9.1 A2A.LSMS.VAL.ACTIVATE.BYNPAC.SubscriptionVersion

<b>Purpose</b>	To verify the LSMS can accept a subscription version for a new NPA-NXX which is activated by the NPAC SMS Simulator. The New Service Provider SOA and Old Service Provider SOA are simulated.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct Impact on providing service.
<b>Prerequisites</b>	All MOC test cases.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The NPAC SMS Simulator locally creates a subscription version for a new NPA-NXX and issues the subscriptionVersionNewNPA-NXX notification to the LSMS.</li> <li>2. The LSMS confirms the subscriptionVersionNewNPA-NXX notification.</li> <li>3. The NPAC SMS Simulator proceeds to activate the newly created version, sets its subscription version status to 'sending', and issues the M-CREATE request to the LSMS.</li> <li>4. The LSMS handles the M-CREATE request for the subscription version instance and provides a successful response.</li> <li>5. The NPAC SMS Simulator locally sets the subscription version status to 'active'.</li> </ol>
<b>Expected Results</b>	The LSMS handles the notification, creates the subscriptionVersion locally and responds successfully to the NPAC SMS Simulator.

### 16.9.2 A2A.LSMS.VAL.MODIFY.BYNPAC.ACTIVE.SubscriptionVersion

<b>Purpose</b>	To verify a LSMS can handle subscription version modifications initiated by the NPAC SMS Simulator.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct Impact on providing service.

<b>Prerequisites</b>	A subscriptionVersion had been created and activated.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The NPAC SMS Simulator locally initiates the modification of an active subscription version. The status of that version is set to 'sending' by the NPAC SMS Simulator. The NPAC SMS Simulator then issues the M-SET request for the subscription version to the LSMS.</li> <li>2. The LSMS handles the M-SET request for the subscriptionVersion attributes successfully and responds to the NPAC SMS Simulator.</li> <li>3. The NPAC SMS Simulator locally sets the subscription version status to 'active'.</li> </ol>
<b>Expected Results</b>	The LSMS updates the subscriptionVersion attributes locally and responds successfully to the NPAC SMS Simulator.

### 16.9.3 A2A.LSMS.VAL.IMMDISC.BYNPAC.SubscriptionVersion

<b>Purpose</b>	To verify the LSMS can handle an immediate disconnect of an active subscription version initiated by NPAC SMS Simulator.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct Impact on providing service.
<b>Prerequisites</b>	A subscriptionVersion had been created and activated.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The NPAC SMS Simulator locally initiates the immediate disconnect of an active subscription version. The status of that subscription version is set to 'sending' by the NPAC SMS Simulator. The NPAC SMS Simulator then issues the M-DELETE request for the subscription version to the LSMS.</li> <li>2. The LSMS handles the M-DELETE request for the subscriptionVersion successfully and responds to the NPAC SMS Simulator.</li> <li>3. The NPAC SMS Simulator locally sets the subscription version status to 'old'.</li> </ol>
<b>Expected Results</b>	The LSMS deletes the subscriptionVersion locally and responds successfully to the NPAC SMS Simulator.

### 16.9.4 A2A.LSMS.VAL.CREATE.MULT.SubscriptionVersion

<b>Purpose</b>	To verify the LSMS can handle a create action for a group of subscription versions with the same routing information.
<b>Severity</b>	R
<b>Severity Explanation</b>	Direct Impact on providing service.
<b>Prerequisites</b>	A2A.LSMS.VAL.ACTIVATE.BYNPAC.SubscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The NPAC SMS Simulator locally initiates the creation and activation of five subscription versions with the same routing information. Their subscription version statuses are set to 'sending'. The NPAC SMS Simulator then issues the M-ACTION request subscriptionVersionLocalSMS-Create to the LSMS.</li> <li>2. The LSMS validates the M-ACTION request, and returns the response to the NPAC SMS Simulator.</li> <li>3. The LSMS locally creates the new versions specified by the action and issues the subscriptionVersionLocalSMS-</li> </ol>

	CreateResults notification upon completion to the NPAC SMS Simulator. 4. The NPAC SMS Simulator confirms the notification and verifies the results.
<b>Expected Results</b>	The LSMS creates the subscription versions locally and responds successfully to the NPAC SMS Simulator.

#### 16.9.5 A2A.LSMS.INV.CREATE.MULT.SubscriptionVersion

<b>Purpose</b>	To verify the LSMS can handle a create action for a group of subscription versions with the same routing information where one of the versions has an invalid TN.
<b>Severity</b>	O
<b>Severity Explanation</b>	Direct Impact on providing service. LSMS may perform to verify error handling.
<b>Prerequisites</b>	A2A.LSMS.VAL.CREATE.MULT.SubscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The NPAC SMS Simulator locally initiates the creation and activation of five subscription versions with the same routing information. Their statuses are set to 'sending'. The NPAC SMS Simulator then issues the M-ACTION request subscriptionVersionLocalSMS-Create, to create these versions on the LSMS. The action argument will contain a subscription version object with an invalid TN (7 digits instead of 10).</li> <li>2. The LSMS validates the M-ACTION request and returns the response to the NPAC SMS Simulator.</li> <li>3. The LSMS locally creates the new versions which have valid TNs as specified by the action, fails to create the invalid version, and issues the subscriptionVersionLocalSMS-CreateResults notification with a Failed TN List which includes the erroneous version's TN.</li> <li>4. The NPAC SMS Simulator confirms the notification and verifies the results.</li> </ol>
<b>Expected Results</b>	The LSMS creates the valid subscription versions locally and identifies the erroneous version in the response to the NPAC SMS Simulator.

#### 16.9.6 A2A.LSMS.INV.CREATE.UNKNOWN.NPA-NXX.SubscriptionVersion

<b>Purpose</b>	To verify the LSMS can handle the condition where it receives a create request for a subscription version with an NPA-NXX for which the subscriptionVersionNewNPA-NXX notification was never sent to that LSMS.
<b>Severity</b>	O
<b>Severity Explanation</b>	No requirements exist. LSMS may perform to verify the situation.
<b>Prerequisites</b>	A2A.LSMS.VAL.ACTIVATE.BYNPAC.SubscriptionVersion
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The NPAC SMS Simulator locally creates a subscription version for a new NPA-NXX, but does not issue the subscriptionVersionNewNPA-NXX notification.</li> <li>2. The NPAC SMS Simulator activates the new subscription version and sends an M-CREATE request for the subscription version to the LSMS.</li> </ol>

	3. The LSMS handles the request, creates the version instance and responds successfully to the M-CREATE request.
<b>Expected Results</b>	The missing notification does not adversely affect the LSMS.

**16.10 SOA WSMSC Data Test Cases (NANC 203)**

**16.10.1 A2A.NSOA.VAL.CREATE.WSMSC.SubscriptionVersion**

<b>Purpose</b>	Verify the New Service Provider SOA can perform a create for a subscription version on the NPAC SMS Simulator with WSMSC DPC and SSN specified. The Old Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if the SOA will be supporting WSMSC data and the product supports new service provider create messages.
<b>Prerequisites</b>	None.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA issues the M-ACTION subscriptionVersionNewSP-Create with WSMSC data.</li> <li>2. The NPAC SMS Simulator handles the local subscriptionVersionNPAC instance create, sends the M-ACTION response to the New Service Provider SOA and issues the objectCreation or subscriptionVersionRangeObjectCreation notification.</li> <li>3. The New Service Provider SOA handles the notification(s) sent by the NPAC SMS Simulator that contains the ‘pending’ state for the created versions and responds with the notification confirmation(s).</li> </ol>
<b>Expected Results</b>	The New Service Provider SOA successfully initiates the subscriptionVersionNewSP-Create M-ACTION and handles the subsequent interactions with the NPAC SMS Simulator.

**16.10.2 A2A.NSOA.VAL.MODIFY.WSMSC.SubscriptionVersion**

<b>Purpose</b>	To verify a SOA can modify an active subscription’s version WSMSC DPC and SSN. The LSMSs are simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if the SOA will be supporting WSMSC data and the product supports subscription version modification.
<b>Prerequisites</b>	An active subscription version exists for the service provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues an M-ACTION subscriptionVersionModify to NPAC SMS Simulator that modifies the WSMSC DPC and SSN, and handles the action response message from the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionNPAC attributes values for the instance to be modified, and then it sets the instance’s subscriptionVersionStatus to ‘sending’.</li> <li>3. The NPAC SMS Simulator emulates receiving positive responses from all the SMSs and locally sets the instance’s</li> </ol>

	<p>subscriptionVersionStatus to 'active'.</p> <p>4. The SOA handles the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification for the 'active' status sent by the NPAC SMS Simulator, and responds with confirmation.</p>
<b>Expected Results</b>	The SOA successfully initiates the subscriptionVersionModify M-ACTION and handles the subsequent interactions with the NPAC SMS Simulator.

16.10.3 A2A.SOA.VAL.QUERY.WSMSC.SubscriptionVersion

<b>Purpose</b>	To verify a SOA can query a subscription version with WSMSC DPC and SSN set.
<b>Severity</b>	O
<b>Severity Explanation</b>	Test case should be executed if the SOA will be supporting WSMSC data and subscription version query.
<b>Prerequisites</b>	SubscriptionVersionNPAC exists on the NPAC SMS Simulator with WSMSC data.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues the M-GET for the specified subscription version with WSMSC data.</li> <li>2. The NPAC SMS Simulator responds with the M-GET result.</li> </ol>
<b>Expected Results</b>	The SOA successfully initiates the M-GET and successfully handles the M-GET result.

**16.11 LSMS WSMSC Data Test Cases (NANC 203)**

16.11.1 A2A.LSMS.VAL.CREATE.WSMSC.SubscriptionVersion

<b>Purpose</b>	To verify the LSMS can successfully process a subscription version create with WSMSC DPC and SSN data present.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if the LSMS is supporting WSMSC data.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends M-CREATE request for the subscription version with WSMSC data.</li> <li>2. LSMS responds to the M-CREATE.</li> </ol>
<b>Expected Results</b>	The LSMS accepts the M-CREATE and returns an M-CREATE response.

16.11.2 A2A.LSMS.VAL.CREATE.MULT.WSMSC.SubscriptionVersion

<b>Purpose</b>	To verify the LSMS can successfully process the M-ACTION subscriptionVersionLocalSMS-Create with WSMSC DPC and SSN data present and respond with the M-EVENT-REPORT, subscriptionVersionLocalSMS-ActionResults.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if the LSMS is supporting WSMSC data.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends M-ACTION</li> </ol>

	<p>subscriptionVersionLocalSMS-Create request with WSMSC data.</p> <ol style="list-style-type: none"> <li>LSMS responds with an M-ACTION response.</li> <li>LSMS sends M-EVENT-REPORT subscriptionVersionLocalSMS-ActionResults.</li> <li>NPAC SMS Simulator confirms M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The Local SMS receives the M-ACTION and the M-EVENT-REPORT and replies to both to the NPAC SMS Simulator.

16.11.3 A2A.LSMS.VAL.QUERY.SCOPE.D.WSMSC.SubscriptionVersion

<b>Purpose</b>	To verify the LSMS can successfully process a scoped and filtered M-GET request with a TN range for a subscription version with WSMSC DPC and SSN data present.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if the LSMS is supporting WSMSC data.
<b>Prerequisites</b>	One or more subscriptionVersions exist on the Local SMS.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>NPAC SMS Simulator sends a scoped and filtered M-GET request specifying a TN range that has WSMSC data.</li> <li>LSMS responds with linked getResult.</li> </ol>
<b>Expected Results</b>	The LSMS processes the M-GET report and replies to the request with all the attributes for the instances.

16.11.4 A2A.LSMS.VAL.MODIFY.WSMSC.SubscriptionVersion

<b>Purpose</b>	To verify the LSMS can successfully process an M-SET request for a subscription version with WSMSC DPC and SSN data present.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if the LSMS is supporting WSMSC data.
<b>Prerequisites</b>	A subscriptionVersion exists on the Local SMS.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>NPAC SMS Simulator sends M-SET request for WSMSC data.</li> <li>LSMS responds to the M-SET.</li> </ol>
<b>Expected Results</b>	The Local SMS receives the M-SET request and responds to the NPAC SMS Simulator.

**16.12 Subscription Timer and Business Types (NANC 201 and 202)**

16.12.1 A2A.SOA.VAL.QUERY.SUBTIMER.SubscriptionVersion

<b>Purpose</b>	To test that a SOA can query a subscription version with subscriptionTimerType value set.
<b>Severity</b>	O
<b>Severity Explanation</b>	Test case should be executed if the SOA will be supporting subscription timer data and subscription version query.
<b>Prerequisites</b>	SubscriptionVersionNPAC exists on the NPAC SMS Simulator with subscriptionTimerType for the service provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>The SOA issues the M-GET for the specified subscription version with subscriptionTimerType.</li> <li>The NPAC SMS Simulator responds with the M-GET result.</li> </ol>
<b>Expected Results</b>	The SOA successfully initiates the M-GET and successfully handles

	the M-GET result.
--	-------------------

16.12.2 A2A.SOA.VAL.QUERY.BUSTYPE.SubscriptionVersion

<b>Purpose</b>	To test that a SOA can query a subscription version with the subscriptionBusinessType value set.
<b>Severity</b>	O
<b>Severity Explanation</b>	Test case should be executed if the SOA will be supporting business type data and subscription version query.
<b>Prerequisites</b>	SubscriptionVersionNPAC exists on the NPAC SMS Simulator with subscriptionBusinessType.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues the M-GET for the specified subscription version with subscriptionBusinessType.</li> <li>2. The NPAC SMS Simulator responds with the M-GET result.</li> </ol>
<b>Expected Results</b>	The SOA successfully initiates the M-GET and successfully handles the M-GET result.

16.12.3 A2A.OSOA.VAL.NOT.subscriptionVersionOldSP-ConcurrenceRequest

<b>Purpose</b>	To test that a SOA can successfully handle the subscriptionVersionOldSP-ConcurrenceRequest with the subscriptionTimerType and subscriptionBusinessType values included. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if the SOA will be supporting business type and/or subscription timer data.
<b>Prerequisites</b>	A subscriptionVersionNPAC instance has been created on the NPAC SMS Simulator with the service provider as the old service provider that has not concurred.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends the subscriptionVersionOldSP-ConcurrenceRequest or subscriptionVersionRangeOldSP-ConcurrenceRequest M-EVENT-REPORT with subscriptionTimerType and subscriptionBusinessType.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA responds with the M-EVENT-REPORT confirmation.

16.12.4 A2A.OSOA.VAL.NOT.subscriptionVersionOldSPFinalConcurrenceWindowExpiration

<b>Purpose</b>	To test that a SOA can successfully handle the subscriptionVersionOldSPFinalConcurrenceWindow with the subscriptionTimerType and subscriptionBusinessType values included. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be run if the SOA will be supporting business type and/or subscription timer data.
<b>Prerequisites</b>	A subscriptionVersionNPAC instance has been created on the NPAC SMS Simulator where the service provider is the old service provider.

<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends the subscriptionVersionOldSP-ConcurrenceRequest or subscriptionVersionRangeOldSP-ConcurrenceRequestM-EVENT-REPORT with subscriptionTimerType and subscriptionBusinessType.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA responds with the M-EVENT-REPORT confirmation.

16.12.5 A2A.NSOA.VAL.NOT.subscriptionVersionNewSP-CreateRequest

<b>Purpose</b>	To test that a SOA can successfully handle the subscriptionVersionNewSP-CreateRequest with the subscriptionTimerType and subscriptionBusinessType values included. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if the SOA will be supporting business type and/or subscription timer data.
<b>Prerequisites</b>	A subscriptionVersionNPAC instance has been created on the NPAC SMS Simulator with the service provider as the new service provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator sends the subscriptionVersionNewSP-CreateRequest or subscriptionVersionRangeNewSP-CreateRequestM-EVENT-REPORT with subscriptionTimerType and subscriptionBusinessType.</li> <li>2. SOA confirms the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	The SOA responds with the M-EVENT-REPORT confirmation.

**16.13 Missing Sending Notification Test Cases (NANC 207)**

16.13.1 A2A.NSOA.VAL.ACTIVATE.NOTMISS.SubscriptionVersion

<b>Purpose</b>	To test that the New Service Provider SOA can handle the condition where the ‘sending’ status change notification is never sent by the NPAC SMS Simulator in an activate subscription version scenario. The LSMSs are simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Test case must be run to ensure the new service provider SOA can correctly process the NPAC SMS Simulator notifications for a subscription version activate.
<b>Prerequisites</b>	Pending subscriptionVersionNPAC exists with the service provider as the new service provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues the M-ACTION subscriptionVersionActivate to NPAC SMS Simulator, and handles the action response message from the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionNPAC instance’s subscriptionVersionStatus to ‘sending’.</li> <li>3. The NPAC SMS Simulator does not send the notification for the ‘sending’ status subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange.</li> <li>4. The NPAC SMS Simulator emulates receiving positive</li> </ol>



	<p>responses from all the SMSs and locally sets the instance's subscriptionVersionStatus to 'active'.</p> <p>5. The SOA handles the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification for the 'active' status sent by the NPAC SMS Simulator, and responds with confirmation.</p>
<b>Expected Results</b>	The new service provider SOA successfully issues the subscriptionVersionActivate M-ACTION and handles the subsequent interactions with the NPAC SMS Simulator.

16.13.2 A2A.OSOA.VAL.ACTIVATE.NOTMISS.SubscriptionVersion

<b>Purpose</b>	To test that the Old Service Provider SOA can handle the condition where the 'sending' status change notification is never sent by the NPAC SMS Simulator in an activation of a subscription version scenario. The LSMSs and New SOA are simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Test case must be run to ensure the old service provider SOA can correctly process the NPAC SMS Simulator notifications for a subscription version activate.
<b>Prerequisites</b>	Pending subscription version exists with the service provider as the old service provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The simulated new service provider SOA issues the M-ACTION subscriptionVersionActivate to NPAC SMS Simulator, and handles the action response message from the NPAC SMS simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionNPAC instance's subscriptionVersionStatus to 'sending'.</li> <li>3. The NPAC SMS Simulator does not send the notification for the 'sending' status subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange.</li> <li>4. The NPAC SMS Simulator emulates receiving positive responses from all the SMSs and locally sets the instance's subscriptionVersionStatus to 'active'.</li> <li>5. The SOA handles the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification for the 'active' status sent by the NPAC SMS Simulator, and responds with confirmation.</li> </ol>
<b>Expected Results</b>	The old service provider SOA can successfully handles the interactions for subscription version activation with the NPAC SMS Simulator.

16.13.3 A2A.SOA.VAL.MODIFY.ACTIVE.NOTMISS.SubscriptionVersion

<b>Purpose</b>	To test a SOA can handle the condition where the 'sending' status change notification is never sent by the NPAC SMS Simulator in a modify active subscription version scenario. The LSMSs are simulated. This test case must be executed twice if a SOA is
----------------	--

	supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Test case must be run to ensure the service provider SOA can correctly process the NPAC SMS Simulator notifications for a modify active if the service provider supports modify active functionality.
<b>Prerequisites</b>	Active subscription version exists for the service provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues the M-ACTION subscriptionVersionModify to NPAC SMS Simulator to modify routing data and handles the action response message from the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionNPAC attributes values for the instance to be modified, and then it sets the instance’s subscriptionVersionStatus to ‘sending’.</li> <li>3. The NPAC SMS Simulator does not send the notification for the ‘sending’ status subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange.</li> <li>4. The NPAC SMS Simulator emulates receiving positive responses from all the SMSs and locally sets the instance’s subscriptionVersionStatus to ‘active’.</li> <li>5. The SOA handles the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification for the ‘active’ status sent by the NPAC SMS Simulator, and responds with confirmation.</li> </ol>
<b>Expected Results</b>	The SOA successfully initiates the subscriptionVersionModify M-ACTION and handles the subsequent interactions with the NPAC SMS Simulator.

16.13.4 A2A.SOA.VAL.IMMDISC.NOTMISS.SubscriptionVersion

<b>Purpose</b>	To test that a SOA can handle the condition where the ‘sending’ status change notification is never sent by the NPAC SMS Simulator in a disconnect subscription version scenario. The LSMSs are simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	R
<b>Severity Explanation</b>	Test case must be run to ensure the service provider SOA can correctly process the NPAC SMS Simulator notifications for an immediate disconnect of a subscription version if the functionality is supported by the service provider.
<b>Prerequisites</b>	Active subscription version exists for the service provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues the M-ACTION subscriptionVersionDisconnect to NPAC SMS Simulator, and handles the action response message from the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionNPAC instance’s subscriptionVersionStatus to ‘sending’.</li> <li>3. The NPAC SMS Simulator does not send the notification for the ‘sending’ status subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange.</li> </ol>

	<ol style="list-style-type: none"> <li>4. The NPAC SMS Simulator emulates receiving positive responses from all the SMSs and locally sets the instance's subscriptionVersionStatus to 'old'.</li> <li>5. The SOA handles the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification for the 'old' status sent by the NPAC SMS Simulator, and responds with confirmation.</li> </ol>
<b>Expected Results</b>	The SOA successfully initiates the subscriptionVersionDisconnect M-ACTION and handles the subsequent interactions with the NPAC SMS Simulator.

**16.14 Associated Service Provider Test Cases (NANC 48)**

16.14.1 A2A.NSOA.VAL.CREATE.FIRST.ASSOCSP.SubscriptionVersion

<b>Purpose</b>	Verify that the New SOA, acting for an associated service provider, can perform a first create for a subscription version on the NPAC SMS Simulator. The Old Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if a SOA is to support associated service provider subscription version processing.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA issues the M-ACTION subscriptionVersionNewSP-Create with an associated service provider is specified in the access control SystemId field and in the new service provider id in the action. The NPAC SMS Simulator handles the local subscriptionVersionNPAC create for the associated service provider, and sends the M-ACTION response to the New Service Provider SOA.</li> <li>2. The New Service Provider SOA acting for an associated service provider handles the objectCreation or subscriptionVersionRangeObjectCreation notification sent by the NPAC SMS Simulator, which contains the 'pending' state for the newly created version, and responds with the notification confirmation.</li> </ol>
<b>Expected Results</b>	The New Service Provider SOA successfully initiates the subscriptionVersionNewSP-Create M-ACTION for an associated service provider and handles the subsequent interactions with the NPAC SMS Simulator.

16.14.2 A2A.NSOA.VAL.CREATE.SECOND.ASSOCSP.SubscriptionVersion

<b>Purpose</b>	Verify that the New Service Provider SOA for an associated service provider can perform a second create for a subscription version after the simulated Old Service Provider SOA create. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if a SOA is to support associated service

	provider subscription version processing.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The simulated Old Service Provider SOA issues a local request to create a new subscription version with the new service provider set to the service provider id of the associated service provider.</li> <li>2. The NPAC SMS Simulator handles the local subscriptionVersionNPAC create and sends the M-ACTION response for the associated service provider id to the Old Service Provider SOA.</li> <li>3. The New Service Provider SOA handles the objectCreation or subscriptionVersionRangeObjectCreation notification from the NPAC SMS Simulator, and responds with confirmation.</li> <li>4. The New Service Provider SOA issues an M-ACTION request to the NPAC SMS simulator for subscriptionVersionNewSP-Create with an associated service provider is specified in the access control SystemId field and in the new service provider id in the action. The simulator handles the local M-SET for the existing subscriptionVersionNPAC instance, and sends the action response to the New Service Provider SOA.</li> <li>5. The New Service Provider SOA handles the attributeValueChange or subscriptionVersionRangeAttributeValueChange notification sent by the NPAC SMS Simulator for the associated service provider, and responds with confirmation.</li> </ol>
<b>Expected Results</b>	The New Service Provider SOA successfully initiates the subscriptionVersionCreate M-ACTION after the new service provider create for an associated service provider and handles the interactions with the NPAC SMS Simulator.

16.14.3 A2A.OSOA.VAL.CREATE.FIRST.ASSOCSP.SubscriptionVersion

<b>Purpose</b>	Verify that the Old Service Provider SOA for an associated service provider can perform a first create for a subscription version on the NPAC SMS Simulator. The New Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if a SOA is to support associated service provider subscription version processing were the old service provider supports issuing the first create if the functionality supported by the product.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The Old Service Provider SOA issues the M-ACTION subscriptionVersionOldSP-Create with an associated service provider is specified in the access control SystemId field and in the old service provider id in the action. The NPAC SMS Simulator handles the local subscriptionVersionNPAC create, and sends the M-ACTION response to the Old Service Provider SOA.</li> <li>2. The Old Service Provider SOA for the associated service provider handles the objectCreation or subscriptionVersionRangeObjectCreation notification sent by the NPAC SMS Simulator that contains the 'pending' state for</li> </ol>

	the newly created version, and responds with the notification confirmation.
<b>Expected Results</b>	The Old Service Provider SOA successfully initiates the subscriptionVersionOldSP-Create M-ACTION and handles the subsequent interactions with the NPAC SMS Simulator for the associated service provider.

16.14.4 A2A.OSOA.VAL.CREATE.SECOND.ASSOCSP.SubscriptionVersion

<b>Purpose</b>	Verify that the Old Service Provider SOA can perform a second create for an associated service provider for a subscription version after the simulated New Service Provider SOA create. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if a SOA is to support associated service provider subscription version processing were the old service provider supports issuing the second create if the functionality is supported by the product.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The simulated New Service Provider SOA issues a local request to create a new subscription version with the old service provider id set to the service provider id of the associated service provider.</li> <li>2. The Old Service Provider SOA handles for the associated service provider the objectCreation or subscriptionVersionRangeObjectCreationnotification from the NPAC SMS Simulator, and responds with confirmation.</li> <li>3. The Old Service Provider SOA issues an M-ACTION request to the NPAC SMS Simulator for subscriptionVersionOldSP-Create with an associated service provider is specified in the access control SystemId field and in the old service provider id in the action. The simulator handles the local M-SET for the existing subscriptionVersionNPAC instance, and sends the action response to the Old Service Provider SOA.</li> <li>4. The Old Service Provider SOA handles for the associated service provider the attributeValueChange or subscriptionVersionRangeAttributeValueChange notification sent by the NPAC SMS Simulator, and responds with confirmation.</li> </ol>
<b>Expected Results</b>	The Old Service Provider SOA successfully initiates the subscriptionVersionOldSP-Create ACTION and handles the interactions with the NPAC SMS Simulator for the associated service provider.

16.14.5 A2A.OSOA.VAL.NOCONC.ACTIVATE.ASSOCSP.SubscriptionVersion

<b>Purpose</b>	Verify that Old Service Provider SOA can handle the situation where a subscription version is activated by the new Service Provider SOA when no concurrence is issued by the old Service Provider SOA. This test case must be executed twice if a SOA is
----------------	--

	supporting both “individual” and “range/list” notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if a SOA is to support associated service provider subscription version processing.
<b>Prerequisites</b>	A pending subscription version exists that can be activated by the new service provider for which the old service provider has not concurred.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The simulated New Service Provider SOA creates first a subscription version on the NPAC SMS Simulator. The NPAC SMS Simulator emits the object creation notification to the Old Service Provider SOA acting for an associated service provider.</li> <li>2. The Old Service Provider SOA responds to the object creation or subscriptionVersionRangeObjectCreation notification sent by the NPAC SMS Simulator.</li> <li>3. No response is received from Old Service Provider SOA regarding the newly created subscription version in “Initial Concurrence Window”. NPAC SMS Simulator sends M-EVENT-REPORT of subscriptionVersionOldSP-ConcurrenceRequest or subscriptionVersionRangeOldSP-ConcurrenceRequest notification to the Old Service Provider SOA.</li> <li>4. The Old Service Provider SOA responds to the notification.</li> <li>5. Still no response from the Old Service Provider SOA regarding the newly created subscription version in “Final Concurrence Window”. NPAC SMS Simulator sends M-EVENT-REPORT of subscriptionVersionOldSP-FinalConcurrenceWindowExpiration or subscriptionVersionRangeOldSP-FinalConcurrenceWindowExpiration notification to the Old Service Provider SOA.</li> <li>6. The Old Service Provider SOA responds to the notification.</li> <li>7. NPAC SMS Simulator simulates sending M-CREATE request on the subscription version to all the Local LSMSs, and getting positive responses from each of the Local SMS. NPAC SMS Simulator sends a subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification with a ‘active’ status to the Old Service Provider SOA acting for an associated service provider.</li> <li>8. Old Service Provider SOA handles and responds to the notification for the ‘active’ status.</li> </ol>
<b>Expected Results</b>	The Old Service Provider SOA successfully handles notifications of subscription version activation from the NPAC SMS Simulator when an old service provider that is an associated service provider has not provided concurrence.

16.14.6 A2A.NSOA.VAL.ACTIVATE.ASSOCSP.SubscriptionVersion

<b>Purpose</b>	To verify the New Service Provider SOA can activate a subscription version in the pending state. The Old Service Provider SOA and the LSMSs are simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if a SOA is to support associated service provider subscription version processing.

<b>Prerequisites</b>	A subscription version exists in a pending state that can be activated for the new service provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA issues, for an associated service provider, the M-ACTION subscriptionVersionActivate for a 'pending' subscriptionVersionNPAC instance on the NPAC SMS Simulator with an associated service provider is specified in the access control and handles the action response message sent by the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus of the 'pending' instance to 'sending'.</li> <li>3. The NPAC SMS Simulator emulates receiving positive responses from all the LSMSs, locally sets the subscriptionVersionStatus of the 'sending' instance to 'active', and sends the corresponding notification to the New Service Provider SOA acting for an associated service provider.</li> <li>4. The New Service Provider SOA handles the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification for the 'active' status, and responds with confirmation.</li> </ol>
<b>Expected Results</b>	New Service Provider SOA, for an associated service provider successfully initiate the subscriptionVersionActivate M-ACTION and handles the subsequent interactions with the NPAC SMS Simulator.

16.14.7 A2A.NSOA.VAL.MODIFY.PEND.ASSOCSP.SubscriptionVersion

<b>Purpose</b>	Test that the New Service Provider SOA, acting for an associated service provider, can modify a subscription version in the pending state using a Modify Action. The Old Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if a SOA is to support associated service provider subscription version processing for new service provider modify pending.
<b>Prerequisites</b>	A pending subscription version exists for the new service provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA, acting for an associated service provider, issues an M-ACTION subscriptionVersionModify to NPAC SMS Simulator to modify the due date with an associated service provider is specified in the access control SystemId and handles the response message from the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the attribute values of the subscriptionVersionNPAC instance, and sends the corresponding attributeValueChange or subscriptionVersionRangeAttributeValueChange notification to the New Service Provider SOA acting for an associated service provider.</li> <li>3. The New Service Provider SOA handles the notification, and responds with confirmation.</li> </ol>
<b>Expected Results</b>	New Service Provider SOA, acting for an associated service provider, successfully initiates the subscriptionVersionModify M-

	ACTION and handles the subsequent interactions with the NPAC SMS Simulator.
--	---

16.14.8 A2A.OSOA.VAL.MODIFY.PEND.ASSOCSP.SubscriptionVersion

<b>Purpose</b>	Test that the Old Service Provider SOA, acting for an associated service provider, can modify a subscription version in the pending state using a Modify Action. The New Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if a SOA is to support associated service provider subscription version processing for old service provider modify pending.
<b>Prerequisites</b>	A pending subscription version exists for the old service provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The Old Service Provider SOA issues, for an associated service provider, an M-ACTION subscriptionVersionModify to NPAC SMS Simulator for the due date with an associated service provider is specified in the access control SystemId and handles the response message from the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the attribute values of the subscriptionVersionNPAC instance, and sends the corresponding attributeValueChange or subscriptionVersionRangeAttributeValueChange notification to the Old Service Provider SOA acting for an associated service provider.</li> <li>3. The Old Service Provider SOA handles the notification, and responds with confirmation.</li> </ol>
<b>Expected Results</b>	Old Service Provider SOA, acting for an associated service provider successfully initiates the subscriptionVersionModify M-ACTION and handles the subsequent interactions with the NPAC SMS Simulator.

16.14.9 A2A.SOA.VAL.MODIFY.ACTIVE.ASSOCSP.SubscriptionVersion

<b>Purpose</b>	To test that a SOA, for an associated service provider, can modify an active subscription version. The LSMSs are simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if a SOA is to support associated service provider subscription version processing for modify active.
<b>Prerequisites</b>	An active subscription version exists for the service provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA, for an associated service provider, issues an M-ACTION subscriptionVersionModify to NPAC SMS Simulator to modify routing data with an associated service provider is specified in the access control SystemId and handles the action response message from the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionNPAC attributes values for the instance to be modified, and then it sets the instance’s subscriptionVersionStatus to ‘sending’.</li> <li>3. The NPAC SMS Simulator emulates receiving positive</li> </ol>



	<p>responses from all the SMSs, locally sets the instance's subscriptionVersionStatus to 'active' and sends a subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange.</p> <p>4. The SOA for an associated service provider, handles the notification sent by the NPAC SMS Simulator for the 'active' status, and responds with confirmation.</p>
<b>Expected Results</b>	The SOA, acting for an associated service provider, successfully initiate the subscriptionVersionModify M-ACTION and handles the subsequent interactions with the NPAC SMS Simulator.

16.14.10 A2A.NSOA.VAL.CANCEL.ASSOCSP.SubscriptionVersion

<b>Purpose</b>	Test that the SOA, acting for an associated service provider, can initiate a cancel request of a pending subscription version. The Old Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if a SOA is to support associated service provider subscription version processing for cancellation.
<b>Prerequisites</b>	A pending subscription version exists for the new service provider were the old service provider has not concurred.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The New Service Provider SOA, acting for an associated service provider, issues an M-ACTION request for subscriptionVersionCancel to NPAC SMS Simulator with an associated service provider is specified in the access control SystemId and handles the response message from the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus of the instance to be canceled to 'canceled', and emits the corresponding notification.</li> <li>3. The New Service Provider SOA, acting for an associated service provider, handles the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification sent by the NPAC SMS Simulator for the 'canceled' status, and confirms it.</li> </ol>
<b>Expected Results</b>	The New Service Provider SOA, acting for an associated service provider, successfully initiates the subscriptionVersionCancel M-ACTION and handles the subsequent interactions with the NPAC SMS Simulator.

16.14.11 A2A.OSOA.VAL.CANCEL.ASSOCSP.SubscriptionVersion

<b>Purpose</b>	Test that the Old Service Provider SOA for an associated service provider can initiate a cancel request of a pending subscription version. The New Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if a SOA is to support associated service provider subscription version processing for cancellation.

<b>Prerequisites</b>	A pending subscription version exists for the old service provider where the new service provider has concurred.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The Old Service Provider SOA for the associated service provider issues an M-ACTION request for subscriptionVersionCancel to NPAC SMS Simulator with an associated service provider is specified in the access control SystemId and handles the response message from the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus of the instance to be canceled to 'cancel-pending', and emits the corresponding notification.</li> <li>3. The Old Service Provider SOA for the associated service provider handles the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification sent by the NPAC SMS Simulator for the 'cancel-pending' status, and confirms it.</li> <li>4. The Old Service Provider SOA for the associated service provider issues the subscriptionVersionOldSP-CancellationAcknowledge M-ACTION request, and handles the response message from the simulator.</li> <li>5. The NPAC SMS Simulator emulates receiving the New Service Provider SOA's cancellation acknowledge request, locally sets the subscriptionVersionStatus to 'canceled', and emits the corresponding notification.</li> <li>6. The Old Service Provider SOA for the associated service provider handles the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification sent by the NPAC SMS Simulator for the 'canceled' status, and confirms it.</li> </ol>
<b>Expected Results</b>	The Old Service Provider SOA for an associated service provider successfully initiates the subscriptionVersionCancel M-ACTION and handles the subsequent interactions with the NPAC SMS Simulator.

16.14.12 A2A.NSOA.VAL.CANCEL.ACKREQ.ASSOCSP.SubscriptionVersion

<b>Purpose</b>	Verify the SOA will respond to the subscriptionVersionCancellationAcknowledgeRequest for an associated service provider as the new service provider with the action subscriptionVersionNewSP-CancellationAcknowledge. The Old Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if a SOA is to support associated service provider subscription version processing for cancellation.
<b>Prerequisites</b>	A subscription version with a cancel-pending status exists for the associated service provider as the new service provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The NPAC SMS Simulator will request that the SOA, for an associated service provider, acknowledge an existing 'cancel-pending' subscription version. The NPAC SMS Simulator sends</li> </ol>

	<p>the notification for subscriptionVersionCancellationAcknowledgeRequest or subscriptionVersionRangeCancellationAcknowledgeRequest to that SOA.</p> <ol style="list-style-type: none"> <li>2. The SOA handles the notification and confirms it.</li> <li>3. The SOA issues the subscriptionVersionNewSP-CancellationAcknowledge action for the associated service provider in response to the notification and handles the NPAC SMS Simulator response. The action sent will have the associated service provider is specified in the access control SystemId</li> <li>4. The NPAC SMS Simulator locally sets the version status to ‘canceled’ and emit the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</li> <li>5. The New Service Provider SOA handles the notification for the ‘canceled’ status and confirms it.</li> </ol>
<b>Expected Results</b>	The SOA successfully handles the cancellation acknowledgement request interactions for the associated service provider as the new service provider with the NPAC SMS Simulator.

16.14.13

A2A.OSOA.VAL.CANCEL.ACKREQ.ASSOCSP.SubscriptionVersion

<b>Purpose</b>	Verify the SOA will respond to the subscriptionVersionCancellationAcknowledgeRequest for an associated service provider as the old service provider with the action subscriptionVersionOldSP-CancellationAcknowledge. The New Service Provider SOA is simulated. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if a SOA is to support associated service provider subscription version processing for cancellation.
<b>Prerequisites</b>	A subscription version with a cancel pending status exists for the associated service provider as the new service provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The NPAC SMS Simulator will request that the SOA, for an associated service provider, acknowledge an existing ‘cancel-pending’ subscription version by sending the notification for subscriptionVersionCancellationAcknowledgeRequest or subscriptionVersionRangeCancellationAcknowledgeRequest to that SOA.</li> <li>2. SOA handles the notification and confirms it.</li> <li>3. The SOA, for the associated service provider, issues the subscriptionVersionOldSP-CancellationAcknowledge action in response to the notification and handles the NPAC SMS Simulator response. The action sent will have the associated service provider is specified in the access control SystemId</li> <li>4. The NPAC SMS Simulator locally sets the version status to ‘canceled’ and emits the</li> </ol>

	<p>subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification.</p> <p>5. The Old service provider SOA handles the notification for the 'canceled' status and confirms it.</p>
<b>Expected Results</b>	The SOA successfully handles the cancellation acknowledgement request interactions for the associated service provider as the old service provider with the NPAC SMS Simulator.

16.14.14 A2A.SOA.VAL.IMMDISC.ASSOCSP.SubscriptionVersion

<b>Purpose</b>	To test that a SOA for an associated service provider can perform an immediate disconnect on an active subscription version. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if a SOA is to support associated service provider subscription version processing for an immediate disconnect.
<b>Prerequisites</b>	A subscription version had been created and activated by the associated service provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA for an associated service provider issues an M-ACTION subscriptionVersionDisconnect to the NPAC SMS Simulator with an associated service provider is specified in the access control SystemId, and handles the response message from the simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus to 'disconnect-pending', and emits the corresponding notification.</li> <li>3. The SOA for an associated service provider handles the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification for the 'disconnect-pending' status and confirms it.</li> <li>4. The NPAC SMS Simulator locally sets the subscriptionVersionStatus to 'sending'.</li> <li>5. The NPAC SMS Simulator emulates receiving the responses from all the LSMSSs, locally sets the subscriptionVersionStatus to 'old', and emits the corresponding notification.</li> <li>6. The SOA for an associated service provider handles the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification for the 'old' status and confirms it.</li> </ol>
<b>Expected Results</b>	The SOA for an associated service provider successfully initiates the subscriptionVersionDisconnect M-ACTION and handles the subsequent interactions with the NPAC SMS Simulator.

16.14.15 A2A.SOA.VAL.DEFDISC.ASSOCSP.SubscriptionVersion

<b>Purpose</b>	To test that a SOA for an associated service provider can perform a deferred disconnect on an active subscription version. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if a SOA is to support associated service

	provider subscription version processing for a deferred disconnect.
<b>Prerequisites</b>	A subscription version had been created and activated by the associated service provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA, for an associated service provider, issues an M-ACTION subscriptionVersionDisconnect with the subscriptionEffectiveReleaseDate attribute set and an associated service provider is specified in the access control SystemId. It then handles the response message from the NPAC SMS Simulator.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionStatus to 'disconnect-pending', and emits the corresponding notification.</li> <li>3. The SOA for an associated service provider handles the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification for the 'disconnect-pending' status and confirms it.</li> </ol>
<b>Expected Results</b>	The SOA for an associated service provider successfully initiates the subscriptionVersionDisconnect for a deferred disconnect and handles the subsequent interactions with the NPAC SMS Simulator.

16.14.16 A2A.NSOA.VAL.CONFLICT.RESOLV.ASSOCSP.SubscriptionVersion

<b>Purpose</b>	To verify that the SOA, for an associated service provider as the new service provider, can handle the resolution of a subscription version conflict. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if a SOA is to support associated service provider subscription version processing for cancellation acknowledgement.
<b>Prerequisites</b>	A subscription version with a status of conflict exists where the new service provider is an associated service provider for the SOA,
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The NPAC SMS Simulator initiates the resolution of the conflict state by locally setting the subscriptionVersionStatus to 'pending', and emits the corresponding notification.</li> <li>2. The New service provider SOA handles the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification for the 'pending' status sent by the NPAC SMS Simulator, and confirms it.</li> </ol>
<b>Expected Results</b>	The SOA, for an associated service provider, has successfully processed the subscriptionVersionStatusAttributeValueChange notification.

16.14.17 A2A.OSOA.VAL.CONFLICT.RESOLV.ASSOCSP.SubscriptionVersion

<b>Purpose</b>	To verify the SOA, for an associated service provider who is the old service provider, can handle the resolution of a subscription version conflict. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	C

<b>Severity Explanation</b>	Test case must be executed if a SOA is to support associated service provider subscription version processing for cancellation acknowledgement.
<b>Prerequisites</b>	A subscription version with a status of conflict exists where the old service provider is an associated service provider for the SOA,
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The NPAC SMS Simulator initiates the resolution of the conflict state by locally setting the subscriptionVersionStatus to 'pending', and emits the corresponding notification.</li> <li>2. The SOA handles the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange notification for the 'pending' status sent by the NPAC SMS Simulator, and confirms it.</li> </ol>
<b>Expected Results</b>	The SOA, for an associated service provider, has successfully processed the subscriptionVersionStatusAttributeValueChange notification.

16.14.18 A2A.SOA.VAL.PORT-TO-ORIG.ASSOCSP.SubscriptionVersion

<b>Purpose</b>	To verify that the SOA can, for an associated service provider, create a port-to-original port. This test case must be executed twice if a SOA is supporting both "individual" and "range/list" notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if a SOA is to support associated service provider subscription version processing for port-to-original porting.
<b>Prerequisites</b>	
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues the subscriptionVersionCreate action for an associated service provider specifying a port-to-original by setting the subscriptionPortingToOriginal-SPSwitch attribute to TRUE. The associated service provider is specified in the access control SystemId and in the new service provider id. The SOA handles the M-ACTION response.</li> <li>2. The NPAC SMS Simulator locally creates the 'pending' subscription version and emits the objectCreation or subscriptionVersionRangeObjectCreation notification.</li> <li>3. The SOA handles the notification sent by the NPAC SMS Simulator, and confirms it.</li> <li>4. The SOA issues, for the associated service provider, the M-ACTION subscriptionVersionActivate for the newly created and 'pending' subscriptionVersionNPAC instance. The associated service provider is specified in the access control SystemId and in the new service provider id. The SOA handles the action response message sent by the NPAC simulator.</li> <li>5. The NPAC SMS Simulator locally sets the subscriptionVersionStatus of the 'pending' instance to 'sending'.</li> <li>6. The NPAC SMS Simulator emulates deleting that version from all the LSMSs, locally sets the subscriptionVersionStatus to 'old', and sends the subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange</li> </ol>

	notification to the SOA for the 'old' status. 7. The SOA handles the notification for the 'old' Status, and responds with confirmation.
<b>Expected Results</b>	The SOA successfully initiates the port to original for the associated service provider and handles the subsequent interactions with the NPAC SMS simulator.

16.14.19 A2A.SOA.CAP.ACT.ASSOCSP.numberPoolBlockCreateAction

<b>Purpose</b>	To verify that the SOA can, for an associated service provider, create number pool block.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if a SOA is to support associated service provider number pool blocks.
<b>Prerequisites</b>	N/A
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid numberPoolBlockCreate M-ACTION request for an associated service provider. The associated service provider is specified in the access control SystemId and in the new service provider id.</li> <li>NPAC SMS Simulator responds with a successful M-ACTION response.</li> </ol>
<b>Expected Results</b>	The SOA successfully initiates the port to original for the associated service provider and handles the subsequent interactions with the NPAC SMS simulator.

16.14.20 A2A.SOA.CAP.OP.SET.ASSOCSP.numberPoolBlock

<b>Purpose</b>	Verify the SOA's ability to SET all the modifiable attributes of the numberPoolBlockNPAC managed object instance for an associated service provider.
<b>Severity</b>	C
<b>Severity Explanation</b>	Test case must be executed if a SOA is to support associated service provider number pool blocks.
<b>Prerequisites</b>	A numberPoolBlockNPAC object exists on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-SET request for all modifiable attributes of the numberPoolBlockNPAC object.</li> <li>NPAC SMS Simulator responds with a successful M-SET result containing all modifiable attributes.</li> </ol>
<b>Expected Results</b>	SOA issues a valid M-SET request for the associated service provider and updates the attributes successfully on the NPAC SMS Simulator.

**16.15 Miscellaneous Scenarios Test Cases**

<b>MO</b>	Miscellaneous Scenarios Test Cases
<b>Purpose</b>	This section contains the test cases for Miscellaneous Scenarios, pertaining to the Application processes of the SOA and LSMS to NPAC SMS Simulator Interface, as part of the Application to Application testing of the NPAC SMS Interoperability Test.
<b>Prerequisite</b>	All Managed Object and stack to stack testing completed. SOA, NPAC SMS Simulator, and LSMS stacks and applications running.

16.15.1 A2A.SOA.VAL.MISC.ACTION.resync

<b>Purpose</b>	Verify SOA can process resynchronization updates from NPAC SMS Simulator. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if a SOA is to support network and/or notification data recovery.
<b>Prerequisites</b>	Network and notification data exist to recover.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA established association with NPAC SMS Simulator, with resynchronization flag on.</li> <li>2. SOA, if supporting network data recovery, sends the InpDownload action request to NPAC SMS Simulator to start network data download for a specified period of time.</li> <li>3. NPAC SMS Simulator responds with network data updates.</li> <li>4. SOA, if supporting notification data recovery, sends the InpNotificationRecovery action request to NPAC SMS Simulator to start notification data download for a specified period of time.</li> <li>5. NPAC SMS Simulator responds with notification updates.</li> <li>6. SOA sends action request to NPAC SMS Simulator to set the resynchronization flag off.</li> <li>7. NPAC SMS Simulator sends the action response.</li> </ol>
<b>Expected Results</b>	SOA associates in recovery mode, issues data download and/or notification recovery actions, and receives action responses containing network and/or notification data updates. Test case must be executed twice for each type of recovery if both recovery requests can not be issued sequentially.

16.15.2 A2A.SOA.INV.MISC.ACTION.resync

<b>Purpose</b>	<p>Verify SOA can process resynchronization update errors from NPAC SMS Simulator.</p> <p>The purpose of this scenario is to test the behavior of the SOA under the following conditions with the SOA having established association with NPAC SMS Simulator, with resynchronization flag on:</p> <ol style="list-style-type: none"> <li>1. SOA, if supporting network data, sends the InpDownload action request to start network data download for a specified period of time and the NPAC SMS Simulator responds with criteria-too-large error.</li> <li>2. SOA, if supporting notification data recovery, sends the InpNotificationRecovery action request to NPAC SMS Simulator to start notification data download for a specified period of time and the NPAC SMS Simulator responds with criteria-too-large error.</li> </ol> <p>The procedure and behavior responses for this scenario should be agreed upon between the test engineer and vendor.</p>
<b>Severity</b>	O
<b>Severity Explanation</b>	Required if a SOA is to support network and/or notification data recovery.



<b>Prerequisites</b>	Network and notification data exists to be recovered.
<b>Procedure</b>	To be determined by vendor and test engineer.  Note: This test case may need to be executed more than once to complete the testing requirements.
<b>Expected Results</b>	SOA associates in recovery mode, issues data download and/or notification recovery actions, and receives action and handles error conditions.

16.15.3 A2A.SOA.VAL.MISC.ACTION.ASSOCSP.resync

<b>Purpose</b>	Verify SOA can process resynchronization updates from NPAC SMS Simulator for an associated service provider. This test case must be executed twice if a SOA is supporting both “individual” and “range/list” notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if a SOA is to support network and/or notification data recovery for an associated service provider.
<b>Prerequisites</b>	Network and notification data exist to recover for the associated service provider.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA established association with NPAC SMS Simulator, with resynchronization flag on.</li> <li>2. SOA, if supporting network data recovery, sends the InpDownload action request to NPAC SMS Simulator to start network data download at a specified period of time for the associated service provider.</li> <li>3. NPAC SMS Simulator responds with network data updates.</li> <li>4. SOA, if supporting notification data recovery, sends the InpNotificationRecovery action request to NPAC SMS Simulator to start notification data download for a specified period of time for the associated service provider.</li> <li>5. NPAC SMS Simulator responds with notification updates.</li> <li>6. SOA sends the InpRecoveryComplete action request to the NPAC SMS Simulator to set the resynchronization flag off.</li> <li>7. NPAC SMS Simulator responds to the action.</li> </ol>
<b>Expected Results</b>	SOA associates in recovery mode, issues data download and notification recovery actions, and receives action responses containing network and notification data updates time for the associated service. Test case must be executed twice for each type of recovery if both recovery requests can not be issued sequentially.

16.15.4 A2A.LSMS.VAL.MISC.ACTION.resync

<b>Purpose</b>	Verify LSMS can process resynchronization updates from NPAC SMS Simulator.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if a LSMS is to support notification data recovery.
<b>Prerequisites</b>	Network, subscription, and notification data exist to recover.  Test case must be executed independently for each type of recovery supported if the recovery requests can not be issued sequentially.

<p><b>Procedure</b></p>	<ol style="list-style-type: none"> <li>1. LSMS established association with NPAC SMS Simulator, with resynchronization flag on.</li> <li>2. LSMS, if supported, sends the InpDownload action request to NPAC SMS Simulator to start network data download at a specified period of time.</li> <li>3. NPAC SMS Simulator responds with network data updates.</li> <li>4. LSMS, if supported, sends the InpDownload action request to NPAC SMS Simulator to start subscription data download at a specified period of time.</li> <li>5. NPAC SMS Simulator responds with subscription data updates.</li> <li>6. LSMS, if supported, sends the InpDownload action request to NPAC SMS Simulator to start number pool block data download at a specified period of time.</li> <li>7. NPAC SMS Simulator responds with number pool block updates.</li> <li>8. LSMS, if supported, sends the InpNotificationRecovery action request to NPAC SMS Simulator to start notification data download for a specified period of time.</li> <li>9. NPAC SMS Simulator responds with notification data updates.</li> <li>10. LSMS sends the InpRecoveryComplete action request to NPAC to set resynchronization flag off.</li> <li>11. NPAC SMS Simulator responds to the action.</li> </ol>
<p><b>Expected Results</b></p>	<p>LSMS associates in recovery mode, issues data download and notification recovery actions, and receives action responses containing network, subscription and notification data updates and takes appropriate action to update its databases.</p>

16.15.5 A2A.LSMS.INV.MISC.ACTION.resync

<p><b>Purpose</b></p>	<p>Verify LSMS can process resynchronization update errors from NPAC SMS Simulator.</p> <p>The purpose of this scenario is to test the behavior of the LSMS under the following conditions with the LSMS having established association with NPAC SMS Simulator, with resynchronization flag on:</p> <ol style="list-style-type: none"> <li>1. LSMS sends the InpDownload action request to start network data download for a specified period of time and the NPAC SMS Simulator responds with criteria-too-large error.</li> <li>2. LSMS sends the InpDownload action request to start subscription data download for a specified period of time and the NPAC SMS Simulator responds with criteria-too-large error.</li> <li>3. LSMS sends the InpDownload action request to start number pool block dat download for a specified period of time and the NPAC SMS Simulator responds with criteria-too-large error.</li> <li>4. LSMS, if supporting notification data recovery, sends the InpNotificationRecovery action request to NPAC SMS Simulator to start notification data download for a specified period of time and the NPAC SMS Simulator responds with criteria-too-large error.</li> </ol>
-----------------------	---

	The procedure and behavior responses for this scenario should be agreed upon between the test engineer and vendor.
<b>Severity</b>	O
<b>Severity Explanation</b>	Required if a LSMS is to support notification data recovery.
<b>Prerequisites</b>	Network, subscription, and notification data exist to recover.
<b>Procedure</b>	To be determined by vendor and test engineer.  Note: This test case may need to be executed more than once to complete the testing requirements.
<b>Expected Results</b>	LSMS associates in recovery mode, issues data download and notification recovery actions, and receives action responses and handles error conditions. Test case must be executed for each type of recovery supported if the recovery requests can not be issued sequentially.

16.15.6 A2A.SOA.VAL.MISC.ACTION.resync\_3\_1

<b>Purpose</b>	Verify SOA can process resynchronization updates from NPAC SMS Simulator at such time the SOA changes from supporting “individual” subscription version notifications to “range/list” notifications.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if a SOA is supporting “range/list” notifications.
<b>Prerequisites</b>	Network and notification data exist to recover. The notification data should contain at least one of all types of “range/list” notifications each with the RANGE set to 1 TN. See section 5.5.1-2 for a complete list of the notifications.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA established association with NPAC SMS Simulator, with resynchronization flag on.</li> <li>2. SOA, if supporting network data recovery, sends the InpDownload action request to NPAC SMS Simulator to start network data download for a specified period of time.</li> <li>3. NPAC SMS Simulator responds with network data updates.</li> <li>4. SOA, if supporting notification data recovery, sends the InpNotificationRecovery action request to NPAC SMS Simulator to start notification data download for a specified period of time.</li> <li>5. NPAC SMS Simulator responds with notification updates.</li> <li>6. SOA sends action request to NPAC SMS Simulator to set the resynchronization flag off.</li> <li>7. NPAC SMS Simulator sends the action response.</li> </ol>
<b>Expected Results</b>	SOA associates in recovery mode, issues data download and/or notification recovery actions, and receives action responses containing network and/or notification data updates. Test case must be executed twice for each type of recovery if both recovery requests cannot be issued sequentially.

**16.16 A2A Number Pooling – SOA to NPAC SMS**

16.16.1 A2A.SOA.VAL.GET.SCOPE.subsriptionVersion.TN-LNPTYPE

<b>Purpose</b>	Verify the SOA's ability to correctly issue a scope and filtered M-GET request for all the attributes of a subsriptionVersionNPAC managed object instance by TN and subsriptionLNPTYPE.
<b>Severity</b>	O
<b>Severity Explanation</b>	Required if SOA will be querying the NPAC SMS.
<b>Prerequisites</b>	N/A
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid M-GET request for all attributes of a subsriptionVersionNPAC object with a filter set for TN and subsriptionLNPTYPE.</li> <li>NPAC SMS Simulator responds with a successful M-GET result containing all attributes.</li> </ol>
<b>Expected Results</b>	SOA issues a valid scope and filtered M-GET request for the subsription version and successfully handles the reply.

**16.17 A2A Number Pooling – LSMS to NPAC SMS**

16.17.1 A2A.LSMS.VAL.GET.SCOPE.subsriptionVersion.TN-LNPTYPE

<b>Purpose</b>	Verify the LSMS's ability to issue a scope and filtered M-GET request for all the attributes of a subsriptionVersion managed object instance by TN and subsriptionLNPTYPE.
<b>Severity</b>	O
<b>Severity Explanation</b>	Required if LSMS will be querying the NPAC SMS.
<b>Prerequisites</b>	N/A
<b>Procedure</b>	<ol style="list-style-type: none"> <li>LSMS issues a valid M-GET request for all attributes of a subsriptionVersion object with a filter set for TN and subsriptionLNPTYPE.</li> <li>NPAC SMS Simulator responds with a successful M-GET result containing all attributes.</li> </ol>
<b>Expected Results</b>	LSMS issues a valid scope and filtered M-GET request for the subsription version and successfully handles the reply.

**16.18 A2A Number Pooling NPAC SMS to LSMS**

16.18.1 A2A.LSMS.VAL.CREATE.BYNPAC.subsriptionVersion.POOL

<b>Purpose</b>	Verify the non-EDR LSMS's ability to respond correctly to an M-CREATE request for a single subsription version with subsriptionLNPTYPE equal to 'POOL'.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if the non-EDR LSMS is supporting subsriptionVersions with subsriptionLNPTYPE equal to 'POOL'.
<b>Prerequisites</b>	N/A
<b>Procedure</b>	<ol style="list-style-type: none"> <li>NPAC SMS Simulator issues an M-CREATE request for a subsriptionVersion object. The subsriptionLNPTYPE is set to 'POOL'.</li> <li>LSMS responds with a successful M-CREATE response.</li> </ol>
<b>Expected Results</b>	NPAC SMS Simulator receives the successful response.

16.18.2 A2A.LSMS.VAL.CREATE.RANGE.BYNPAC.subscriptionVersion.POOL

<b>Purpose</b>	Verify the non-EDR LSMS's ability to respond correctly to a create request for a multiple subscription versions with subscriptionLNPTType equal to 'POOL'.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if the non-EDR LSMS is supporting subscriptionVersions with subscriptionLNPTType equal to 'POOL'.
<b>Prerequisites</b>	N/A
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues a valid subscriptionVersionLocalSMS-CreateAction.</li> <li>2. LSMS responds with a successful subscriptionVersionLocalSMSCreateReply M-ACTION reply.</li> </ol>
<b>Expected Results</b>	NPAC SMS Simulator receives the successful response.

16.18.3 A2A.LSMS.VAL.GET.SCOPE.D.BYNPAC.subscriptionVersion.TN-LNPTYPE

<b>Purpose</b>	Verify the LSMS's ability to respond correctly to a scoped and filtered M-GET request for the subscriptionVersion managed object instances where the subscriptionLNPTType is set to 'POOL'. The filter contains a TN Range.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required for all non-EDR LSMSs.
<b>Prerequisites</b>	N/A
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC SMS Simulator issues a valid scoped and filtered M-GET request for all attributes of the subscriptionVersion object.</li> <li>2. LSMS responds with the linked M-GET results containing all attributes.</li> </ol>
<b>Expected Results</b>	NPAC SMS Simulator receives the valid responses.

# Appendix A Testing Registration Form

## FOR NANC 2.0.0 to 3.1.0

09-17-01

**Service Provider Name:**

**Service Provider Agent/Supplier:**

**Point of Contact (for testing):**

Name:

Email:

Phone:

Fax:

Address:

**Testing Period:** (start must be filled)

**Date**

**Time**

Start:

End:

**Extended Testing Hours Expected:**

YES

NO

**Test Access Type:**

Dedicated

**Interface Under Test:**

SOA

LSMS

**Test Category:**

Note In order to conduct NANC 3.0 or higher NANC 2.0 must have been complete successfully.

**NANC**

**2.0**

**3.0**

**3.1.0**

--	--	--

Stack-to-Stack

--	--	--

Security Group A

--	--	--

Security Group B

--	--	--

Managed Object

--	--	--

Association Management

--	--	--

Application-to-Application

**System Under Test:**

System Name:  
System Version:  
Location  
Hardware Platform Identification:  
Operating System Identification:  
Protocol Stack Identification:  
Test Driver Identification:

**(If two systems are being tested)**

**System Under Test:**

System Name:  
System Version:  
Location  
Hardware Platform Identification:  
Operating System Identification:  
Protocol Stack Identification:  
Test Driver Identification:

---

<b>Protocol Stack Profile:</b>	<b>Reference</b>	<b>Version</b>	<b>PICS reference</b>
--------------------------------	------------------	----------------	-----------------------

**Information Object Reference:**

Interface Specification  
Functional Specification  
MOCS Reference

**Other Conformance References:**

SCS Reference:  
ICS Reference(s)  
IXIT Reference(s)

**Prerequisite Met for Testing:**            **Yes**        **No**





## Appendix B Test Case Nomenclature

STACK-TO-STACK TEST-ID SYMBOLS	
ABBREVIATION	DESCRIPTION
S2S	Stack-to-Stack Interoperability Testing
VAL	Valid Test
INV	Invalid Test
SOA	Initiating system is SOA
LSMS	Initiating system is LSMS
ASSOC	Association (A-ASSOCIATE) Request
RELES	Release (A-RELEASE) Request
ABORT	Abort (A-ABORT) Request
INVK	Invalid KEY
INVT	Invalid Time
ISMFU	Invalid Systems Management Functional Unit Identifier
ISEQ	Invalid Sequence Number

SECURITY TEST-ID SYMBOLS	
ABBREVIATION	DESCRIPTION
SEC	Security Interoperability Testing
VAL	Valid Test
INV	Invalid Test
SOA	Initiating system is SOA
LSMS	Initiating system is LSMS
ASSOC	Association (A-ASSOCIATE) Request
RELES	Release (A-RELEASE) Request
ABORT	Abort (A-ABORT) Request
INVK	Invalid KEY
INVT	Invalid Time
ISMFU	Invalid Systems Management Functional Unit Identifier
ISEQ	Invalid Sequence Number

MOC TEST-ID SYMBOLS	
ABBREVIATION	DESCRIPTION
MOC	Managed Object Conformance Interoperability Testing
NPAC	Initiating System is NPAC
SOA	Initiating system is SOA
LSMS	Initiating system is LSMS
CAP	MO Capability Test
OP	Operation Test
NOT	Notification Test
ACT	Action Test

MOC TEST-ID SYMBOLS	
ABBREVIATION	DESCRIPTION
VAL	Valid behaviour Test
INV	Invalid behaviour Test
CRE	MO Instance Create Test
DEL	MO Instance Delete Test
SET	Attribute Set Test
GET	Attribute Get Test
SING	Operation on Single Attribute Test
MULT	Operation on Multiple Attribute Test
COND	Operation on Conditional Attribute Test
AUTO	Automatic Object Naming
RO	Read Only
CO	Contained Objects
SCOP	Scoped Test
FILT	Filter Test
BND	Boundary Test
MIN	Lower Bound Test
MAX	Upper Bound Test
MAXQ	Maximum number of allowed queries
MAXB	Maximum number of allowed Bytes
RANGE	Tests the “range” structure of a “range/list” notification
LIST	Tests the “list” structure of a “range/list” notification

RECOVERY TEST-ID SYMBOLS	
ABBREVIATION	DESCRIPTION
AMG	Association Management Interoperability Testing
SOA	Initiating system is SOA
LSMS	Initiating system is LSMS
ASSOC	Association (A-ASSOCIATE) Request
REASSOC	re-establish Association
REQTMOT	Request Timeout Test
RETRY	Retry a Request
SWOV	Switch Over
BKUP	Backup NPAC
CMIP	CMIP requests
SECVIOL	Security Violation Test
LOSS	Association Loss Test
DOWN	NPAC Down test
SAME	Retry Same Host
OTHER	Retry Other Host

A2A TEST-ID SYMBOLS	
ABBREVIATION	DESCRIPTION
A2A	Application to Application Test
LSMS	System Under Test is an LSMS
SOA	System Under Test is a SOA
NSOA	System Under Test is a New SOA
OSOA	System Under Test is an Old SOA
DSOA	System Under Test is a Donor SOA
VAL	Valid Transaction Test
INV	Invalid Transaction / Inopportune Behavior Test
Audit Test Cases	
MISSVER	Missing Subscription Version Test
OBSVER	Old Subscription Version Test
ERRVER	Erroneous Subscription Version Test
NODIS	No Discrepancy Found Test
TN	Single Telephone Number Test
TNRNG	Telephone Number Range Test
ACTRNG	Activation Range Test
WITHDIS	Audit Discrepancy Found Test
NPACCNCLD	Canceled by NPAC
CRENOT	Object Creation Notification
TIMOUT	Operation/Transaction Timeout Test
COMP	Audit Complete Test
NORES	Missing Audit Results Test
NUMTN	Audit Number of TNs Test
COMPTN	Completed Number of TNs Test
NUMDISERR	Number of Discrepancies Error Test
Service Provider and Network Data Test Cases	
SETSP	Set Service Provider Test
CREND	Create a Network Data Instance Test
DELND	Delete a Network Data Instance Test
Subscription Version Test Cases	
CREATE	Subscription Version Creation Test
ACTIVATE	Subscription Version Activation Test
MODIFY	Subscription Version Modification Test
CANCEL	Subscription Version Cancellation Test
IMMDISC	Subscription Version Immediate Disconnect Test
DEFDISC	Subscription Version Deferred Disconnect Test
STATE-TRANS	State Transition Test
FIRST	First Create Transaction Test
SECOND	Second Create Transaction Test

A2A TEST-ID SYMBOLS	
ABBREVIATION	DESCRIPTION
TN-RANGE	Telephone Number Range Transaction Test
PEND	Pending Subscription Version Test
CONFLICT	Conflict Subscription Version Test
ACT, ACTIVE	Active Subscription Version Test
OLD	Old Subscription Version Test
PARTFAIL	Partially Failed Subscription Version Test
FAIL, FAILED	Failed Subscription Version Test
SENDING	Sending Subscription Version Test
CANCEL-PEND	Cancel-Pending Subscription Version Test
DISCPEND	Disconnect-Pending Subscription Version Test
OBJCRE	Object Creation Notification Test
NOTMISS	Missing Notification Test
ACTNOTMISS	Active Status Missing Notification Test
BYNPAC	Operation Performed by NPAC Test
BYOSOA	Operation Performed by Old SOA Test
BYNSOA	Operation Performed by New SOA Test
ATTRCHNG	Attribute is Changed Test
STATCHNG	Status Attribute is Changed Test
ATTRSAME	Attribute in Unchanged Test
NONONC	No Concurrence by Other SOA Test
ACKREQ	Acknowledge Request Test
RESOLV	Conflict Resolution Test
PORT-TO-ORIG	Port To Original SP Test
MULT	Multiple Versions Test
UNKNOWN	Unknown Instance Test
Miscellaneous Test Cases	
MISC	Miscellaneous Test
ACTION	Action Request Test
EVENT	Event Report Test
SET	Set Request Test

## Appendix C Complete ITP Test Case Checklist

Test Case Number and Name			Sev	Date	Result
<b>Stack to Stack Test Cases</b>					
1	9.1.1	S2S.SOA.PING and S2S.LSMS.PING	O		
2	9.1.2	S2S.SOA.FTP and S2S.LSMS.FTP	R		
3	9.1.3	S2S.SOA.VAL.ASSOC and S2S.LSMS.VAL.ASSOC	R		
4	9.1.4	S2S.SOA.VAL.RELES and S2S.LSMS.VAL.RELES	R		
5	9.1.5	S2S.SOA.VAL.RELES.BYNPAC and S2S.LSMS.VAL.RELES.BYNPAC	O		
6	9.1.6	S2S.SOA.VAL.ABORT and S2S.LSMS.VAL.ABORT	R		
7	9.1.7	S2S.SOA.VAL.ABORT.BYNPAC and S2S.LSMS.VAL.ABORT.BYNPAC	R		
<b>Security Test Cases</b>					
1	10.1.1	SEC.SOA.VAL.ASSOC.NOSIG and SEC.LSMS.VAL.ASSOC.NOSIG	O		
2	10.1.2	SEC.SOA.INV.ASSOC.INVSYS and SEC.LSMS.INV.ASSOC.INVSYS	R		
3	10.1.3	SEC.SOA.INV.ASSOC.INVT and SEC.LSMS.INV.ASSOC.INVT	R		
4	10.1.4	SEC.SOA.ASSOC.SEQ and SEC.LSMS.INV.ASSOC.SEQ	R		
5	10.2.1	SEC.SOA.VAL.ASSOC and SEC.LSMS.VAL.ASSOC	R		
6	10.2.2	SEC.SOA.INV.ASSOC.INVK and SEC.LSMS.INV.ASSOC.INVK	R		
7	10.2.3	SEC.SOA.INV.ASSOC.INVSIG and SEC.LSMS.INV.ASSOC.INVSIG	R		
8	10.2.4	SEC.SOA.INV.NOT.INVSIG and SEC.LSMS.INV.NOT.INVSIG	R		
9	10.2.5	SEC.LSMS.INV.CREATE.INVSEQ	R		
10	10.2.6	SEC.LSMS.INV.SET.INVSIG	R		
11	10.2.7	SEC.LSMS.INV.ACTION.INVSYS	R		
12	10.2.8	<a href="#">SEC.SOA.INV.GET.INVT</a> and SEC.LSMS.INV.GET.INVT	R		
13	10.2.9	<a href="#">SEC.SOA.INV.DELETE.INVSIG</a> and SEC.LSMS.INV.DELETE.INVSIG	R		
14	10.2.10	SEC.SOA.INV.ASSOC.ASSOCSP.INVSYS	C		
<b>MOC InpNPAC-SMS (SOA to NPAC SMS)</b>					
1	11.1.1	MOC.SOA.CAP.OP.GET.InpNPAC-SMS	O		
2	11.1.2	MOC.SOA.CAP.NOT.InpNPAC-SMS-Operational-Information	R		
3	11.1.3	MOC.SOA.INV.NOT.InpNPAC-SMS-Operational-Information	O		
4	11.1.4	MOC.SOA.CAP.NOT.subscriptionVersionNewNPA-NXX	C		
5	11.1.5	MOC.SOA.INV.GET.InpNPAC-SMS	C		
6	11.1.6	MOC.SOA.INV.NOT.subscriptionVersionNewNPA-	O		

Test Case Number and Name			Sev	Date	Result
		NXX			
7	11.1.7*	MOC.SOA.CAP.ACT.InpNotificationRecovery	C		
8	11.1.8	MOC.SOA.INV.ACT.InpNotificationRecovery	C		
9	11.1.9	MOC.SOA.CAP.OP.ACT.InpRecoveryComplete	C		
10	11.1.10	MOC.SOA.INV.ACT.InpRecoveryComplete	C		
<b>MOC InpServiceProvs (SOA to NPAC SMS)</b>					
1	11.2.1	MOC.SOA.CAP.OP.GET.InpServiceProvs	O		
2	11.2.2	MOC.SOA.INV.GET.InpServiceProvs	O		
<b>MOC InpAudits (SOA to NPAC SMS)</b>					
1	11.3.1	MOC.SOA.CAP.OP.GET.InpAudits	O		
2	11.3.2	MOC.SOA.INV.GET.InpAudits	O		
<b>MOC InpSubscriptions (SOA to NPAC SMS)</b>					
1	11.4.1	MOC.SOA.CAP.OP.GET.InpSubscriptions	O		
2	11.4.2*	MOC.SOA.CAP.ACT.subscriptionVersionNewSP-Create-Initial	R		
3	11.4.3*	MOC.SOA.CAP.ACT.subscriptionVersionOldSP-Create-Initial	R		
4	11.4.4*	MOC.SOA.CAP.ACT.subscriptionVersionNewSP-Create-Second	R		
5	11.4.5	MOC.SOA.CAP.ACT.subscriptionVersionOldSP-Create-Second	R		
6	11.4.6	MOC.SOA.CAP.ACT.subscriptionVersionActivate-VersionId	C		
7	11.4.7	MOC.SOA.CAP.ACT.subscriptionVersionActivate-TN	C		
8	11.4.8	MOC.SOA.CAP.ACT.subscriptionVersionActivate-TNRange	C		
9	11.4.9	MOC.SOA.CAP.ACT.subscriptionVersionModify	R		
10	11.4.10	MOC.SOA.CAP.ACT.subscriptionVersionCancel	R		
11	11.4.11	MOC.SOA.CAP.ACT.subscriptionVersionOldSP-CancellationAcknowledge	R		
12	11.4.12	MOC.SOA.CAP.ACT.subscriptionVersionNewSP-CancellationAcknowledge	R		
13	11.4.13	MOC.SOA.CAP.ACT.subscriptionVersionDisconnect	R		
14	11.4.14	MOC.SOA.CAP.ACT.subscriptionVersionRemoveFromConflict	R		
15	11.4.15	MOC.SOA.INV.GET.InpSubscriptions	C		
16	11.4.16	MOC.SOA.INV.ACT.subscriptionVersionNewSP-Create	R		
17	11.4.17	MOC.SOA.INV.ACT.subscriptionVersionOldSP-Create	R		
18	11.4.18	MOC.SOA.INV.ACT.subscriptionVersionActivate	R		
19	11.4.19	MOC.SOA.INV.ACT.subscriptionVersionModify	R		
20	11.4.20	MOC.SOA.INV.ACT.subscriptionVersionCancel	R		
21	11.4.21	MOC.SOA.INV.ACT.subscriptionVersionOldSP-CancellationAcknowledge	R		
22	11.4.22	MOC.SOA.INV.ACT.subscriptionVersionNewSP-CancellationAcknowledge	R		
23	11.4.23	MOC.SOA.INV.ACT.subscriptionVersionDisconnect	R		

Test Case Number and Name			Sev	Date	Result
24	11.4.24	MOC.SOA.INV.ACT.subscriptionVersionRemoveFromConflict	R		
25	11.4.25	MOC.SOA.CAP.ACT.numberPoolBlockCreateAction	C		
26	11.4.26	MOC.SOA.INV.ACT.numberPoolBlockCreateAction	C		
27	11.4.27	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeStatusAttributeValueChange	C		
28	11.4.28	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeAttributeValueChange	C		
29	11.4.29	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeObjectCreation	C		
30	11.4.30	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeDonorSP-CustomerDisconnectDate	C		
31	11.4.31	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeCancellationAcknowledgeRequest	C		
32	11.4.32	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeNewSP-CreateRequest	C		
33	11.4.33	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeOldSP-ConcurrenceRequest	C		
34	11.4.34	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeOldSPFinalConcurrenceWindowExpiration	C		
35	11.4.35	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRangeNewSP-FinalCreateWindowExpiration	C		
36	11.4.36	MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeStatusAttributeValueChange	C		
<del>37</del>	<del>11.4.37</del>	<del>MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeAttributeValueChange</del>	<del>C</del>		
<del>37</del> <del>38</del>	<del>11.4.373</del> <del>8</del>	<del>MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeObjectCreation</del>	<del>C</del>		
<del>38</del> <del>39</del>	<del>11.4.383</del> <del>8</del>	<del>MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeDonorSP-CustomerDisconnectDate</del>	<del>C</del>		
<del>39</del> <del>40</del>	<del>11.4.394</del> <del>0</del>	<del>MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeCancellationAcknowledgeRequest</del>	<del>C</del>		
<del>40</del> <del>41</del>	<del>11.4.404</del> <del>1</del>	<del>MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeNewSP-CreateRequest</del>	<del>C</del>		
<del>41</del> <del>42</del>	<del>11.4.414</del> <del>2</del>	<del>MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeOldSP-ConcurrenceRequest</del>	<del>C</del>		
<del>42</del> <del>43</del>	<del>11.4.424</del> <del>3</del>	<del>MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeOldSPFinalConcurrenceWindowExpiration</del>	<del>C</del>		
<del>43</del>	<del>11.4.43</del>	<del>MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeNewSP-FinalCreateWindowExpiration</del>	<del>C</del>		
44	11.4.44	MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeNewSP-FinalCreateWindowExpiration	C		
45	11.4.45	MOC.SOA.INV.NOT.RANGE.subscriptionVersionRangeStatusAttributeValueChange	O		
46	11.4.46	MOC.SOA.INV.NOT.RANGE.subscriptionVersionRangeAttributeValueChange	O		
47	11.4.47	MOC.SOA.INV.NOT.RANGE.subscriptionVersionRangeObjectCreation	O		
48	11.4.48	MOC.SOA.INV.NOT.RANGE.subscriptionVersionRangeDonorSP-CustomerDisconnectDate	O		
49	11.4.49	MOC.SOA.INV.NOT.RANGE.subscriptionVersionRangeCancellationAcknowledgeRequest	O		
50	11.4.50	MOC.SOA.INV.NOT.RANGE.subscriptionVersionRangeNewSP-CreateRequest	O		

Test Case Number and Name			Sev	Date	Result
51	11.4.51	MOC.SOA.INV.NOT.RANGE.subscriptionVersionRange OldSP-ConcurrenceRequest	O		
52	11.4.52	MOC.SOA.INV.NOT.RANGE.subscriptionVersionRange OldSPFinalConcurrenceWindowExpiration	O		
53	11.4.53	MOC.SOA.INV.NOT.RANGE.subscriptionVersionRange NewSP-FinalCreateWindowExpiration	O		
<b>MOC InpNetwork (SOA to NPAC SMS)</b>					
1	11.5.1	MOC.SOA.CAP.OP.GET.InpNetwork	O		
2	11.5.2	MOC.SOA.INV.GET.InpNetwork	C		
3	11.5.3	MOC.SOA.CAP.ACT.InpNetwork.InpDownload	C		
4	11.5.4	MOC.SOA.INV.ACT.InpNetwork.InpDownload	C		
5	11.5.5	MOC.SOA.VAL.InpDownload-NPA-NXX-X	C		
<b>MOC serviceProv (SOA to NPAC SMS)</b>					
1	11.6.1	MOC.SOA.CAP.OP.SET.serviceProv	C		
2	11.6.2	MOC.SOA.CAP.OP.GET.serviceProv	O		
3	11.6.3	MOC.SOA.VAL.SET.SING.serviceProv	C		
4	11.6.4	MOC.SOA.VAL.SET.SING.COND.serviceProv	C		
5	11.6.5	MOC.SOA.VAL.SET.MULT.serviceProv	C		
6	11.6.6	MOC.SOA.INV.SET.serviceProv	C		
7	11.6.7	MOC.SOA.INV.GET.serviceProv	C		
8	11.6.8	MOC.SOA.BND.MIN.SET.serviceProv	C		
9	11.6.9	MOC.SOA.BND.MAX.SET.serviceProv	C		
<b>MOC subscriptionAudit (SOA to NPAC SMS)</b>					
1	11.7.1	MOC.SOA.CAP.OP.CRE.subscriptionAudit	C		
2	11.7.2	MOC.SOA.CAP.OP.GET.subscriptionAudit	O		
3	11.7.3	MOC.SOA.CAP.OP.DEL.subscriptionAudit	O		
4	11.7.4	MOC.SOA.CAP.NOT.subscriptionAuditResults	C		
5	11.7.5	MOC.SOA.CAP.NOT.subscriptionAudit- DiscrepancyReport	C		
6	11.7.6	MOC.SOA.VAL.CRE.AUTO.subscriptionAudit	O		
7	11.7.7	MOC.SOA.VAL.GET.SCOPE.FILT.subscriptionAudit	O		
8	11.7.8	MOC.SOA.VAL.DEL.SCOPE.subscriptionAudit	O		
9	11.7.9	MOC.SOA.INV.CRE.subscriptionAudit	C		
10	11.7.10	MOC.SOA.INV.GET.subscriptionAudit	C		
11	11.7.11	MOC.SOA.INV.DEL.subscriptionAudit	C		
12	11.7.12	MOC.SOA.INV.NOT.subscriptionAuditResults	O		
13	11.7.13	MOC.SOA.INV.NOT.subscriptionAudit- DiscrepancyReport	O		
14	11.7.14	MOC.SOA.INV.CAP.OP.CRE.subscriptionAudit	O		
<b>MOC subscriptionVersionNPAC (SOA to NPAC SMS)</b>					
1	11.8.1	MOC.SOA.CAP.OP.SET.OldSP.subscriptionVersion NPAC	O		
2	11.8.2	MOC.SOA.CAP.OP.SET.NewSP.subscriptionVersion NPAC	O		
3	11.8.3	MOC.SOA.CAP.OP.GET.subscriptionVersionNPAC	O		
4	11.8.4	MOC.SOA.CAP.NOT.subscriptionVersionOldSP- ConcurrenceRequest	R		
5	11.8.5	MOC.SOA.CAP.NOT.subscriptionVersionOldSP- FinalConcurrenceWindowExpiration	R		



Test Case Number and Name			Sev	Date	Result
6	11.8.6	MOC.SOA.CAP.NOT.subscriptionVersionNewSP-CreateRequest	R		
7	11.8.7	MOC.SOA.CAP.NOT.subscriptionVersionCancellationAcknowledgeRequest	R		
8	11.8.8	MOC.SOA.CAP.NOT.subscriptionVersionDonorSP-CustomerDisconnectDate	R		
9	11.8.9	MOC.SOA.VAL.SET.SING.subscriptionVersionNPAC	O		
10	11.8.10	MOC.SOA.VAL.SET.MULT.subscriptionVersionNPAC	O		
11	11.8.11	MOC.SOA.VAL.GET.SCOPE.subscriptionVersionNPAC	O		
12	11.8.12	MOC.SOA.VAL.NOT.subscriptionVersionNewNPA-NXX	R		
13	11.8.13	MOC.SOA.VAL.NOT.subscriptionVersionStatusAttributeValueChange	R		
14	11.8.14	MOC.SOA.INV.SET.SING.subscriptionVersionNPAC	C		
15	11.8.15	MOC.SOA.INV.GET.subscriptionVersionNPAC	C		
16	11.8.16	MOC.SOA.INV.NOT.subscriptionVersionOldSp-ConcurrenceRequest	O		
17	11.8.17	MOC.SOA.INV.NOT.subscriptionVersionNewSP-CreateRequest	O		
18	11.8.18	MOC.SOA.INV.NOT.subscriptionVersionCancellationAcknowledgeRequest	O		
19	11.8.19	MOC.SOA.INV.NOT.subscriptionVersionDonorSP-CustomerDisconnectDate	O		
20	11.8.20	MOC.SOA.INV.NOT.subscriptionVersionStatusAttributeValueChange	O		
21	11.8.21	MOC.SOA.INV.NOT.attributeValueChange.subscriptionVersion	O		
22	11.8.22	MOC.SOA.INV.NOT.subscriptionVersionNewNPA-NXX	O		
23	11.8.23	MOC.SOA.BND.GET.MAXQ.subscriptionVersionNPAC	R		
24	11.8.24	MOC.SOA.INV.QUERY.SCOPE.subscriptionVersion	C		
25	11.8.25	MOC.SOA.CAP.NOT.subscriptionVersionNewSP-FinalCreateWindowExpiration	C		
26	11.8.26	MOC.SOA.INV.NOT.subscriptionVersionNewSP-FinalCreateWindowExpiration	O		
<b>MOC serviceProvNetwork (SOA to NPAC SMS)</b>					
1	11.9.1	MOC.SOA.CAP.OP.GET.serviceProvNetwork	O		
2	11.9.2	MOC.SOA.INV.GET.serviceProvNetwork	O		
<b>MOC serviceProvNPA-NXX (SOA to NPAC SMS)</b>					
1	11.10.1	MOC.SOA.CAP.OP.GET.serviceProvNPA-NXX	O		
2	11.10.2	MOC.SOA.CAP.OP.DEL.serviceProvNPA-NXX	C		
3	11.10.3	MOC.SOA.VAL.CRE.AUTO.serviceProvNPA-NXX	C		
4	11.10.4	MOC.SOA.VAL.GET.SCOPE.FILT.serviceProvNPA-NXX	O		
5	11.10.5	MOC.SOA.VAL.DEL.SCOPE.FILT.serviceProvNPA-NXX	O		

Test Case Number and Name			Sev	Date	Result
6	11.10.6	MOC.SOA.INV.CRE.serviceProvNPA-NXX	C		
7	11.10.7	MOC.SOA.INV.GET.serviceProvNPA-NXX	C		
8	11.10.8	MOC.SOA.INV.DEL.serviceProvNPA-NXX	C		
<b>MOC serviceProvLRN (SOA to NPAC SMS)</b>					
1	11.11.1	MOC.SOA.CAP.OP.GET.serviceProvLRN	O		
2	11.11.2	MOC.SOA.CAP.OP.DEL.serviceProvLRN	C		
3	11.11.3	MOC.SOA.VAL.CRE.AUTO.serviceProvLRN	C		
4	11.11.4	MOC.SOA.VAL.GET.SCOPE.FILT.serviceProvLRN	O		
5	11.11.5	MOC.SOA.VAL.DEL.SCOPE.FILT.serviceProvLRN	O		
6	11.11.6	MOC.SOA.INV.CRE.serviceProvLRN	C		
7	11.11.7	MOC.SOA.INV.GET.serviceProvLRN	C		
8	11.11.8	MOC.SOA.INV.DEL.serviceProvLRN	C		
<b>MOC numberPoolBlockNPAC (SOA to NPAC SMS)</b>					
1	11.12.1	MOC.SOA.CAP.OP.GET.numberPoolBlockNPAC	O		
2	11.12.2	MOC.SOA.CAP.OP.SET.numberPoolBlockNPAC	C		
3	11.12.3	MOC.SOA.VAL.GET.SCOPE.numberPoolBlockNPAC	O		
4	11.12.4	MOC.SOA.INV.GET.numberPoolBlockNPAC	O		
5	11.12.5	MOC.SOA.INV.SET.numberPoolBlockNPAC	C		
6	11.12.6	MOC.SOA.INV.GET.SCOPE.numberPoolBlockNPAC	O		
<b>MOC numberPoolBlockNPAC (SOA to NPAC SMS)</b>					
1	11.13.1	MOC.SOA.CAP.OP.GET.serviceProvNPA-NXX-X	O		
2	11.13.2	MOC.SOA.VAL.GET.SCOPE.serviceProvNPANXX-X	O		
3	11.13.3	MOC.SOA.INV.GET.serviceProvNPANXX-X	O		
4	11.13.4	MOC.SOA.INV.GET.SCOPE.serviceProvNPANXX-X	O		
<b>MOC InpSOA (NPAC SMS to SOA)</b>					
1	12.1.1	MOC.NPAC.CAP.OP.GET.InpSOA	O		
2	12.1.2	MOC.NPAC.INV.CRE.INH.InpSOA	O		
3	12.1.3	MOC.NPAC.INV.SET.InpSOA	O		
4	12.1.4	MOC.NPAC.INV.DEL.InpSOA	O		
<b>MOC InpNetwork (NPAC SMS to SOA)</b>					
1	12.2.1	MOC.NPAC.SOA.CAP.OP.GET.InpNetwork	O		
2	12.2.2	MOC.NPAC.SOA.INV.CRE.INH.InpNetwork	O		
3	12.2.3	MOC.NPAC.SOA.INV.SET.InpNetwork	O		
4	12.2.4	MOC.NPAC.SOA.INV.ACT.InpNetwork	O		
5	12.2.5	MOC.NPAC.SOA.INV.DEL.InpNetwork	O		
<b>MOC serviceProvNetwork (NPAC SMS to SOA)</b>					
1	12.3.1	MOC.NPAC.SOA.CAP.OP.CRE.serviceProvNetwork	C		
2	12.3.2	MOC.NPAC.SOA.CAP.OP.GET.serviceProvNetwork	O		
3	12.3.3	MOC.NPAC.SOA.CAP.OP.SET.serviceProvNetwork	C		
4	12.3.4	MOC.NPAC.SOA.CAP.OP.DEL.serviceProvNetwork	C		
5	12.3.5	MOC.NPAC.SOA.INV.CRE.DUP.serviceProvNetwork	O		
6	12.3.6	MOC.NPAC.SOA.INV.SET.RO.serviceProvNetwork	O		

Test Case Number and Name			Sev	Date	Result
7	12.3.7	MOC.NPAC.SOA.INV.SET.SYN.serviceProvNetwork	O		
8	12.3.8	MOC.NPAC.SOA.INV.SET.serviceProvNetwork	O		
9	12.3.9	MOC.NPAC.SOA.INV.GET.serviceProvNetwork	O		
10	12.3.10	MOC.NPAC.SOA.INV.DEL.serviceProvNetwork	C		
11	12.3.11	MOC.NPAC.SOA.INV.DEL.CO.serviceProvNetwork	C		
12	12.3.12	MOC.NPAC.SOA.BND.SET.MIN.serviceProvNetwork	C		
13	12.3.13	MOC.NPAC.SOA.BND.SET.MAX.serviceProvNetwork	C		
<b>MOC serviceProvNPA-NXX (NPAC SMS to SOA)</b>					
1	12.4.1	MOC.NPAC.SOA.CAP.OP.CRE.serviceProvNPA-NXX	C		
2	12.4.2	MOC.NPAC.SOA.CAP.OP.DEL.serviceProvNPA-NXX	C		
3	12.4.3	MOC.NPAC.SOA.INV.CRE.DUP.serviceProvNPA-NXX	C		
4	12.4.4	MOC.NPAC.SOA.INV.SET.serviceProvNPA-NXX	O		
5	12.4.5	MOC.NPAC.SOA.INV.DEL.serviceProvNPA-NXX	C		
<b>MOC serviceProvLRN (NPAC SMS to SOA)</b>					
1	12.5.1	MOC.NPAC.SOA.CAP.OP.CRE.serviceProvLRN	C		
2	12.5.2	MOC.NPAC.SOA.CAP.OP.DEL.serviceProvLRN	C		
3	12.5.3	MOC.NPAC.SOA.INV.CRE.DUP.serviceProvLRN	C		
4	12.5.4	MOC.NPAC.SOA.INV.SET.serviceProvLRN	O		
5	12.5.5	MOC.NPAC.SOA.INV.DEL.serviceProvLRN	C		
<b>MOC numberPoolBlock (NPAC SMS to SOA)</b>					
1	12.6.1	MOC.SOA.CAP.NOT.numberPoolBlockAttributeValueChange	C		
2	12.6.2	MOC.SOA.CAP.NOT.numberPoolBlockStatusAttributeValueChange	C		
<b>MOC serviceProvNPA-NXX-X (NPAC SMS to SOA)</b>					
1	12.7.1	MOC.NPAC.SOA.CAP.OP.CRE.serviceProvNPA-NXX-X	C		
2	12.7.2	MOC.NPAC.SOA.CAP.OP.SET.serviceProvNPA-NXX-X	C		
3	12.7.3	MOC.NPAC.SOA.CAP.OP.DEL.serviceProvNPA-NXX-X	C		
4	12.7.4	MOC.NPAC.SOA.INV.CRE.DUP.serviceProvNPA-NXX-X	C		
5	12.7.5	MOC.NPAC.SOA.INV.SET.serviceProvNPA-NXX-X	O		
6	12.7.6	MOC.NPAC.SOA.INV.DEL.serviceProvNPA-NXX-X	C		
<b>MOC InpNPAC-SMS (LSMS to NPAC SMS)</b>					
1	13.1.1	MOC.LSMS.CAP.OP.GET.InpNPAC-SMS	O		
2	13.1.2	MOC.LSMS.CAP.OP.ACT.InpRecoveryComplete	R		
3	13.1.3	MOC.LSMS.CAP.NOT.InpNPAC-SMS-Operational-Information	R		
4	13.1.4	MOC.LSMS.INV.GET.InpNPAC-SMS	O		

Test Case Number and Name			Sev	Date	Result
5	13.1.5	MOC.LSMS.INV.ACT.InpRecoveryComplete	R		
6	13.1.6	MOC.LSMS.INV.NOT.InpNPAC-SMS-Operational-Information	O		
7	13.1.7	MOC.LSMS.CAP.NOT.subscriptionVersionNewNPA-NXX	R		
8	13.1.8	MOC.LSMS.INV.NOT.subscriptionVersionNewNPA-NXX	O		
9	13.1.9	MOC.LSMS.CAP.ACT.InpNotificationRecovery	C		
10	13.1.10	MOC.LSMS.INV.ACT.InpNotificationRecovery	C		
<b>MOC InpServiceProv (LSMS to NPAC SMS)</b>					
1	13.2.1	MOC.LSMS.CAP.OP.GET.InpServiceProvs	O		
2	13.2.2	MOC.LSMS.INV.GET.InpServiceProvs	O		
<b>MOC InpSubscription (LSMS to NPAC SMS)</b>					
1	13.3.1	MOC.LSMS.CAP.OP.GET.InpSubscriptions	O		
2	13.3.2	MOC.LSMS.CAP.ACT.InpSubscriptions.InpDownload	R		
3	13.3.3	MOC.LSMS.INV.GET.InpSubscriptions	O		
4	13.3.4	MOC.LSMS.INV.ACT.InpSubscriptions	R		
5	13.3.5	MOC.LSMS.VAL.InpDownload-NumberPoolBlock	C		
<b>MOC InpNetwork (LSMS to NPAC SMS)</b>					
1	13.4.1	MOC.LSMS.CAP.OP.GET.InpNetwork	O		
2	13.4.2	MOC.LSMS.CAP.ACT.InpNetwork.InpDownload	R		
3	13.4.3	MOC.LSMS.INV.GET.InpNetwork	O		
4	13.4.4	MOC.LSMS.INV.ACT.InpNetwork	R		
5	13.4.5	MOC.LSMS.VAL.InpDownload-NPA-NXX-X	C		
<b>MOC serviceProv (LSMS to NPAC SMS)</b>					
1	13.5.1	MOC.LSMS.CAP.OP.SET.serviceProv	C		
2	13.5.2	MOC.LSMS.CAP.OP.GET.serviceProv	O		
3	13.5.3	MOC.LSMS.VAL.SET.SING.serviceProv	C		
4	13.5.4	MOC.LSMS.VAL.SET.SING.COND.serviceProv	C		
5	13.5.5	MOC.LSMS.VAL.SET.MULT.serviceProv	C		
6	13.5.6	MOC.LSMS.INV.SET.serviceProv	C		
7	13.5.7	MOC.LSMS.INV.GET.serviceProv	C		
8	13.5.8	MOC.LSMS.BND.MIN.SET.serviceProv	C		
9	13.5.9	MOC.LSMS.BND.MAX.SET.serviceProv	C		
<b>MOC IsmsFilterNPA-NXX (LSMS to NPAC SMS)</b>					
1	13.6.1	MOC.LSMS.CAP.OP.CRE.IsmsFilterNPA-NXX	C		
2	13.6.2	MOC.LSMS.CAP.OP.GET.IsmsFilterNPA-NXX	O		
3	13.6.3	MOC.LSMS.CAP.OP.DEL.IsmsFilterNPA-NXX	C		
4	13.6.4	MOC.LSMS.VAL.CRE.AUTO.IsmsFilterNPA-NXX	C		
5	13.6.5	MOC.LSMS.VAL.GET.SCOP.FILT.IsmsFilterNPA-NXX	O		
6	13.6.6	MOC.LSMS.VAL.DEL.SCOP.FILT.IsmsFilterNPA-NXX	O		
7	13.6.7	MOC.LSMS.INV.CRE.IsmsFilterNPA-NXX	C		
8	13.6.8	MOC.LSMS.INV.GET.IsmsFilterNPA-NXX	C		
9	13.6.9	MOC.LSMS.INV.DEL.IsmsFilterNPA-NXX	C		
<b>MOC subscriptionVersionNPAC (LSMS to NPAC SMS)</b>					

Test Case Number and Name			Sev	Date	Result
1	13.7.1	MOC.LSMS.CAP.OP.GET.subscriptionVersionNPAC	O		
2	13.7.2	MOC.LSMS.CAP.NOT.subscriptionVersionNewNPA-NXX	R		
3	13.7.3	MOC.LSMS.VAL.GET.SCOP.subscriptionVersionNPAC	O		
4	13.7.4	MOC.LSMS.INV.GET.subscriptionVersionNPAC	O		
5	13.7.5	MOC.LSMS.INV.NOT.subscriptionVersionNPAC	O		
6	13.7.6	MOC.LSMS.BND.GET.MAXQ.subscriptionVersionNPAC	C		
7	13.7.7	MOC.LSMS.INV.QUERY.SCOPE.D.subscriptionVersion	C		
<b>MOC serviceProvNetwork (LSMS to NPAC SMS)</b>					
1	13.8.1	MOC.LSMS.CAP.OP.GET.serviceProvNetwork	O		
2	13.8.2	MOC.LSMS.INV.GET.serviceProvNetwork	O		
<b>MOC serviceProvNPA-NXX (LSMS to NPAC SMS)</b>					
1	13.9.1	MOC.LSMS.CAP.OP.GET.serviceProvNPA-NXX	O		
2	13.9.2	MOC.LSMS.CAP.OP.DEL.serviceProvNPA-NXX	C		
3	13.9.3	MOC.LSMS.VAL.CRE.AUTO.serviceProvNPA-NXX	C		
4	13.9.4	MOC.LSMS.VAL.GET.SCOP.FILT.serviceProvNPA-NXX	O		
5	13.9.5	MOC.LSMS.VAL.DEL.SCOP.FILT.serviceProvNPA-NXX	O		
6	13.9.6	MOC.LSMS.INV.CRE.serviceProvNPA-NXX	C		
7	13.9.7	MOC.LSMS.INV.GET.serviceProvNPA-NXX	O		
8	13.9.8	MOC.LSMS.INV.DEL.serviceProvNPA-NXX	C		
<b>MOC serviceProvLRN(LSMS to NPAC SMS)</b>					
1	13.10.1	MOC.LSMS.CAP.OP.GET.serviceProvLRN	O		
2	13.10.2	MOC.LSMS.CAP.OP.DEL.serviceProvLRN	C		
3	13.10.3	MOC.LSMS.VAL.CRE.AUTO.serviceProvLRN	C		
4	13.10.4	MOC.LSMS.VAL.GET.SCOP.FILT.serviceProvLRN	O		
5	13.10.5	MOC.LSMS.VAL.DEL.SCOP.FILT.serviceProvLRN	O		
6	13.10.6	MOC.LSMS.INV.CRE.serviceProvLRN	C		
7	13.10.7	MOC.LSMS.INV.GET.serviceProvLRN	O		
8	13.10.8	MOC.LSMS.INV.DEL.serviceProvLRN	C		
<b>MOC numberPoolBlockNPAC (LSMS to NPAC SMS)</b>					
1	13.11.1	MOC.LSMS.CAP.OP.GET.numberPoolBlockNPAC	O		
2	13.11.2	MOC.LSMS.VAL.GET.SCOP.numberPoolBlockNPAC	O		
3	13.11.3	MOC.LSMS.INV.GET.numberPoolBlockNPAC	O		
4	13.11.4	MOC.LSMS.INV.GET.SCOP.numberPoolBlockNPAC	O		
<b>MOC serviceProvNPA-NXX-X (LSMS to NPAC SMS)</b>					
1	13.12.1	MOC.LSMS.CAP.OP.GET.serviceProvNPA-NXX-X	O		
2	13.12.2	MOC.LSMS.VAL.GET.SCOP.serviceProvNPA-NXX-X	O		
3	13.12.3	MOC.LSMS.INV.GET.serviceProvNPA-NXX-X	O		
4	13.12.4	MOC.LSMS.INV.GET.SCOP.serviceProvNPA-	O		

Test Case Number and Name			Sev	Date	Result
		NXX-X			
<b>MOC InpLocalSMS (NPAC SMS to LSMS)</b>					
1	14.1.1	MOC.NPAC.CAP.OP.GET.InpLocalSMS	O		
2	14.1.2	MOC.NPAC.INV.CRE.INH.InpLocalSMS	O		
3	14.1.3	MOC.NPAC.INV.SET.InpLocalSMS	O		
4	14.1.4	MOC.NPAC.INV.DEL.InpLocalSMS	O		
5	14.1.5	MOC.LSMS.CAP.NOT.InpNPAC-SMS-Operational-Information	R		
<b>MOC InpSubscriptions (NPAC SMS to LSMS)</b>					
1	14.2.1	MOC.NPAC.CAP.OP.GET.InpSubscriptions	O		
2	14.2.2	MOC.NPAC.CAP.OP.ACT.InpSubscriptions	R		
3	14.2.3	MOC.NPAC.CAP.OP.NOT.InpSubscriptions	R		
4	14.2.4	MOC.NPAC.INV.CRE.INH.InpSubscriptions	O		
5	14.2.5	MOC.NPAC.INV.SET.InpSubscriptions	O		
6	14.2.6	MOC.NPAC.INV.ACT.SYN.ID.InpSubscriptions	O		
7	14.2.7	MOC.NPAC.INV.ACT.SYN.CLS.InpSubscriptions	O		
8	14.2.8	MOC.NPAC.INV.ACT.InpSubscriptions	O		
9	14.2.9	MOC.NPAC.INV.NOT.InpSubscriptions	R		
10	14.2.10	MOC.NPAC.INV.DEL.InpSubscriptions	O		
<b>MOC InpNetwork (NPAC SMS to LSMS)</b>					
1	14.3.1	MOC.NPAC.CAP.OP.GET.InpNetwork	O		
2	14.3.2	MOC.NPAC.INV.CRE.INH.InpNetwork	O		
3	14.3.3	MOC.NPAC.INV.SET.InpNetwork	O		
4	14.3.4	MOC.NPAC.INV.ACT.InpNetwork	O		
5	14.3.5	MOC.NPAC.INV.DEL.InpNetwork	O		
<b>MOC subscriptionVersion (NPAC SMS to LSMS)</b>					
1	14.4.1	MOC.NPAC.CAP.OP.CRE.subscriptionVersion	R		
2	14.4.2	MOC.NPAC.CAP.OP.SET.subscriptionVersion	R		
3	14.4.3	MOC.NPAC.CAP.OP.GET.subscriptionVersion	R		
4	14.4.4	MOC.NPAC.CAP.OP.DEL.subscriptionVersion	R		
5	14.4.5	MOC.NPAC.VAL.SET.SING.subscriptionVersion	R		
6	14.4.6	MOC.NPAC.VAL.SET.MULT.subscriptionVersion	R		
7	14.4.7	MOC.NPAC.VAL.SET.SCOP.FILT.subscriptionVersion	R		
8	14.4.8	MOC.NPAC.VAL.GET.SCOP.FILT.subscriptionVersion	R		
9	14.4.9	MOC.NPAC.VAL.DEL.SCOP.FILT.subscriptionVersion	R		
10	14.4.10	MOC.NPAC.INV.CRE.subscriptionVersion	O		
11	14.4.11	MOC.NPAC.INV.SET.RO.subscriptionVersion	O		
12	14.4.12	MOC.NPAC.INV.SET.MULT.subscriptionVersion	O		
13	14.4.13	MOC.NPAC.INV.SET.SYN.subscriptionVersion	O		
14	14.4.14	MOC.NPAC.INV.SET.SCOP.subscriptionVersion	R		
15	14.4.15	MOC.NPAC.INV.DEL.SCOP.subscriptionVersion	R		
16	14.4.16	MOC.NPAC.BND.SET.MIN.subscriptionVersion	R		
17	14.4.17	MOC.NPAC.BND.SET.MAX.subscriptionVersion	R		
<b>MOC serviceProvNetwork (NPAC SMS to LSMS)</b>					

Test Case Number and Name			Sev	Date	Result
1	14.5.1	MOC.NPAC.CAP.OP.CRE.serviceProvNetwork	R		
2	14.5.2	MOC.NPAC.CAP.OP.GET.serviceProvNetwork	O		
3	14.5.3	MOC.NPAC.CAP.OP.SET.serviceProvNetwork	R		
4	14.5.4	MOC.NPAC.CAP.OP.DEL.serviceProvNetwork	R		
5	14.5.5	MOC.NPAC.INV.CRE.DUP.serviceProvNetwork	R		
6	14.5.6	MOC.NPAC.INV.SET.RO.serviceProvNetwork	O		
7	14.5.7	MOC.NPAC.INV.SET.SYN.serviceProvNetwork	O		
8	14.5.8	MOC.NPAC.INV.SET.serviceProvNetwork	O		
9	14.5.9	MOC.NPAC.INV.GET.serviceProvNetwork	O		
10	14.5.10	MOC.NPAC.INV.DEL.serviceProvNetwork	R		
11	14.5.11	MOC.NPAC.INV.DEL.CO.serviceProvNetwork	R		
12	14.5.12	MOC.NPAC.BND.SET.MIN.serviceProvNetwork	R		
13	14.5.13	MOC.NPAC.BND.SET.MAX.serviceProvNetwork	R		
<b>MOC serviceProvNPA-NXX (NPAC SMS to LSMS)</b>					
1	14.6.1	MOC.NPAC.CAP.OP.CRE.serviceProvNPA-NXX	R		
2	14.6.2	MOC.NPAC.CAP.OP.DEL.serviceProvNPA-NXX	R		
3	14.6.3	MOC.NPAC.INV.CRE.DUP.serviceProvNPA-NXX	R		
4	14.6.4	MOC.NPAC.INV.SET.serviceProvNPA-NXX	O		
5	14.6.5	MOC.NPAC.INV.DEL.serviceProvNPA-NXX	R		
<b>MOC serviceProvLRN (NPAC SMS to LSMS)</b>					
1	14.7.1	MOC.NPAC.CAP.OP.CRE.serviceProvLRN	R		
2	14.7.2	MOC.NPAC.CAP.OP.DEL.serviceProvLRN	R		
3	14.7.3	MOC.NPAC.INV.CRE.DUP.serviceProvLRN	R		
4	14.7.4	MOC.NPAC.INV.SET.serviceProvLRN	O		
5	14.7.5	MOC.NPAC.INV.DEL.serviceProvLRN	R		
<b>MOC numberPoolBlock (NPAC SMS to LSMS)</b>					
1	14.8.1	MOC.NPAC.CAP.OP.CRE.numberPoolBlock	C		
2	14.8.2	MOC.NPAC.CAP.OP.SET.numberPoolBlock	C		
3	14.8.3	MOC.NPAC.CAP.OP.GET.numberPoolBlock	C		
4	14.8.4	MOC.NPAC.CAP.OP.GET.MULTIPLE.numberPoolBlock	C		
5	14.8.5	MOC.NPAC.CAP.OP.DEL.numberPoolBlock	C		
6	14.8.6	MOC.NPAC.CAP.OP.SET.SING.numberPoolBlock	C		
7	14.8.7	MOC.NPAC.CAP.OP.SET.MULT.numberPoolBlock	C		
8	14.8.8	MOC.NPAC.INV.CRE.numberPoolBlock	O		
9	14.8.9	MOC.NPAC.INV.SET.numberPoolBlock	O		
10	14.8.10	MOC.NPAC.INV.DEL.numberPoolBlock	C		
<b>MOC serviceProvNPA-NXX-X (NPAC SMS to LSMS)</b>					
1	14.9.1	MOC.NPAC.CAP.OP.CRE.serviceProvNPA-NXX-X	C		
2	14.9.2	MOC.NPAC.CAP.OP.SET.serviceProvNPA-NXX-X	C		
3	14.9.3	MOC.NPAC.CAP.OP.DEL.serviceProvNPA-NXX-X	C		
4	14.9.4	MOC.NPAC.INV.CRE.DUP.serviceProvNPA-NXX-X	C		
5	14.9.5	MOC.NPAC.INV.SET.serviceProvNPA-NXX-X	O		
6	14.9.6	MOC.NPAC.INV.DEL.serviceProvNPA-NXX-X	C		
<b>Association Management</b>					



Test Case Number and Name			Sev	Date	Result
1	15.1.1	AMG.SOA.ASSOC.SAME and AMG.LSMS.ASSOC.SAME	R		
2	15.1.2	AMG.SOA.ASSOC.OTHER and AMG.LSMS.ASSOC.OTHER	R		
3	15.1.3	AMG.SOA.REQTMOT and AMG.LSMS.REQTMOT	O		
4	15.1.4	AMG.SOA.RETRY.CMIP and AMG.LSMS.RETRY.CMIP	O		
5	15.1.5	AMG.SOA.RETRY.ASSOC and AMG.LSMS.RETRY.ASSOC	O		
6	15.1.6	AMG.SOA.SECVIOL and AMG.LSMS.SECVIOL	R		
7	15.1.7	AMG.SOA.LOSS and AMG.LSMS.LOSS	R		
8	15.1.8	AMG.SOA.DOWN and AMG.LSMS.DOWN	R		
<b>Audit App-to-App</b>					
1	16.1.1	A2A.LSMS.VAL.MISSVER.subscriptionAudit	R		
2	16.1.2	A2A.LSMS.VAL.OBSVER.subscriptionAudit	R		
3	16.1.3	A2A.LSMS.VAL.ERRVER.subscriptionAudit	R		
4	16.1.4	A2A.SOA.VAL.NODIS.TN.subscriptionAudit	C		
5	16.1.5	A2A.SOA.VAL.NODIS.TNRNG.subscriptionAudit	O		
6	16.1.6	A2A.SOA.VAL.NODIS.ACTRNG.subscriptionAudit	O		
7	16.1.7	A2A.SOA.VAL.WITHDIS.TN.subscriptionAudit	R		
8	16.1.8	A2A.SOA.VAL.WITHDIS.TNRNG.subscriptionAudit	C		
9	16.1.9	A2A.SOA.VAL.WITHDIS.ACTRNG.subscriptionAudit	O		
10	16.1.10	A2A.SOA.VAL.NPACCNCLD.subscriptionAudit	C		
11	16.1.11	A2A.SOA.INV.CRENOT.TIMOUT.subscriptionAudit	O		
12	16.1.12	A2A.SOA.VAL.WITHDIS.WSMSC.RANGE.subscriptionAudit	C		
13	16.1.13	A2A.SOA.VAL.WITHDIS.WSMSC.SINGLE.subscriptionAudit	C		
14	16.1.14	A2A.SOA.VAL.WITHDIS.ASSOCSP.RANGE.subscriptionAudit	C		
15	16.1.15	A2A.SOA.VAL.WITHDIS.ASSOCSP.SINGLE.subscriptionAudit	C		
16	16.1.16	A2A.LSMS.VAL.MISSVER.subscriptionAudit.POOL	C		
<b>Service Provider and Network Data App-to-App</b>					
1	16.2.1	A2A.LSMS.VAL.CREND.serviceProviderNPA-NXX	C		
2	16.2.2	A2A.LSMS.VAL.DELND.serviceProviderNPA-NXX	C		
3	16.2.3	A2A.LSMS.VAL.CREND.serviceProviderLRN	C		
4	16.2.4	A2A.LSMS.VAL.DELND.serviceProviderLRN	C		
5	16.2.5	A2A.SOA.CAP.OP.SET.ASSOCSP.serviceProv	C		
6	16.2.6	A2A.SOA.CAP.OP.GET.ASSOCSP.serviceProv	O		
7	16.2.7	A2A.SOA.VAL.CREND.ASSOCSP.serviceProviderNPA-NXX	C		
8	16.2.8	A2A.SOA.VAL.DELND.ASSOCSP.serviceProvider	C		



Test Case Number and Name		Sev	Date	Result
	NPA-NXX			
9	16.2.9 A2A.SOA.VAL.CREND.ASSOCSP.serviceProviderLRN	C		
19	16.2.10 A2A.SOA.VAL.DELND.ASSOCSP.serviceProviderLRN	C		
<b>Subscription Version Create Data App-to-App</b>				
1	16.3.1* A2A.NSOA.VAL.CREATE.TN-RANGE.SubscriptionVersion	C		
2	16.3.2* A2A.NSOA.VAL.CREATE.CONFLICT.SubscriptionVersion	R		
3	16.3.3* A2A.OSOA.VAL.CREATE.TN-RANGE.SubscriptionVersion	O		
4	16.3.4* A2A.OSOA.VAL.NOCONC.ACTIVATE.SubscriptionVersion	R		
5	16.3.5* A2A.OSOA.VAL.NOCONC.NOACTIVATE.SubscriptionVersion	R		
6	16.3.6* A2A.OSOA.VAL.CREATE.CONFLICT.SubscriptionVersion	R		
7	16.3.7* A2A.NSOA.VAL.CREATE.INTRA-SP-PORT.SubscriptionVersion	C		
8	16.3.8* A2A.DSOA.VAL.PORT-TO-ORIG.SubscriptionVersion	R		
9	16.3.9* A2A.NSOA.INV.MISS.INITIAL.CONC.SubscriptionVersion	O		
10	16.3.10* A2A.NSOA.INV.STATE-TRANS.PEND-ACTIVE.SubscriptionVersion	O		
11	16.3.11* A2A.NSOA.INV.STATE-TRANS.PEND-OLD.SubscriptionVersion	O		
12	16.3.12* A2A.OSOA.INV.STATE-TRANS.PEND-OLD.SubscriptionVersion	O		
13	16.3.13* A2A.OSOA.INV.STATE-TRANS.PEND-FAILED.SubscriptionVersion	O		
14	16.3.14* A2A.NSOA.INV.CREATE.ACTIVE.SubscriptionVersion	O		
15	16.3.15* A2A.OSOA.INV.CREATE.SENDING.SubscriptionVersion	O		
16	16.3.16* A2A.NSOA.INV.OBJCRE.NOTMISS.SubscriptionVersion	O		
17	16.3.17* A2A.OSOA.INV.OBJCRE.NOTMISS.SubscriptionVersion	O		
<b>Subscription Version Activate Data App-to-App</b>				
1	16.4.1* A2A.NSOA.VAL.ACTIVATE.BYNPAC.SubscriptionVersion	R		
2	16.4.2* A2A.NSOA.VAL.ACTIVATE.SubscriptionVersion	R		
3	16.4.3* A2A.NSOA.VAL.ACTIVATE.FAIL.SubscriptionVersion	R		
4	16.4.4* A2A.NSOA.VAL.ACTIVATE.PARTFAIL.SubscriptionVersion	R		
5	16.4.5* A2A.OSOA.VAL.ACTIVATE.SubscriptionVersion	R		
6	16.4.6* A2A.OSOA.VAL.ACTIVATE.FAIL.SubscriptionVersion	R		
7	16.4.7* A2A.OSOA.VAL.ACTIVATE.PARTFAIL.SubscriptionVersion	R		

Test Case Number and Name		Sev	Date	Result
	onVersion			
8	16.4.8* A2A.NSOA.ACTIVATE.ACTNOTMISS.SubscriptionVersion	O		
9	16.4.9* A2A.NSOA.INV.ACTIVATE.PARTFAIL.SubscriptionVersion	O		
10	16.4.10* A2A.OSOA.INV.ACTIVATE.PARTFAIL.SubscriptionVersion	O		
11	16.4.11* A2A.NSOA.VAL.ACTIVATE.TN-RANGE.SubscriptionVersion	C		
<b>Subscription Version Modify Data App-to-App</b>				
1	16.5.1* A2A.NSOA.VAL.MODIFY.PEND.SubscriptionVersion	R		
2	16.5.2* A2A.OSOA.VAL.MODIFY.PEND.SubscriptionVersion	R		
3	16.5.3* A2A.SOA.VAL.MODIFY.ACTIVE.SubscriptionVersion	R		
4	16.5.4* A2A.SOA.VAL.MODIFY.ACTIVE.TN-RANGE.SubscriptionVersion	C		
5	16.5.5* A2A.SOA.VAL.MODIFY.BYNPAC.ACTIVE.SubscriptionVersion	R		
6	16.5.6* A2A.SOA.VAL.MODIFY.PARTFAIL.SubscriptionVersion	R		
7	16.5.7* A2A.SOA.VAL.MODIFY.FAIL.SubscriptionVersion	R		
8	16.5.8* A2A.SOA.INV.MODIFY.PARTFAIL.NOSPLIST.SubscriptionVersion	O		
9	16.5.9 A2A.SOA.INV.MODIFY.ACTIVE.NOTMISS.SubscriptionVersion	O		
10	16.5.10* A2A.SOA.INV.MODIFY.ATTRCHNG.NOTMISS.SubscriptionVersion	O		
11	16.5.11* A2A.SOA.INV.MODIFY.ATTRSAME.NOTMISS.SubscriptionVersion	O		
12	16.5.12* A2A.SOA.VAL.MODIFY.PEND.TN-RANGE.SubscriptionVersion	C		
<b>Subscription Version Cancel Data App-to-App</b>				
1	16.6.1* A2A.SOA.VAL.CANCEL.SubscriptionVersion	R		
2	16.6.2* A2A.NSOA.VAL.CANCEL.BYOSOA.SubscriptionVersion	R		
3	16.6.3* A2A.NSOA.VAL.CANCEL.TN-RANGE.SubscriptionVersion	C		
4	16.6.4* A2A.OSOA.VAL.CANCEL.SubscriptionVersion	R		
5	16.6.5* A2A.OSOA.VAL.CANCEL.BYNSOA.SubscriptionVersion	R		
6	16.6.6* A2A.OSOA.VAL.CANCEL.TN-RANGE.SubscriptionVersion	C		
7	16.6.7* A2A.OSOA.VAL.CANCEL.NOCONC.SubscriptionVersion	R		
8	16.6.8* A2A.NSOA.VAL.CANCEL.BYNPAC.SubscriptionVersion	R		
9	16.6.9* A2A.OSOA.VAL.CANCEL.BYNPAC.SubscriptionVersion	R		
10	16.6.10* A2A.NSOA.VAL.CANCEL.ACKREQ.SubscriptionVersion	R		

Test Case Number and Name			Sev	Date	Result
11	16.6.11*	A2A.OSOA.VAL.CANCEL.ACKREQ.SubscriptionVersion	R		
12	16.6.12*	A2A.NSOA.INV.CANCEL.CONFLICT.SubscriptionVersion	R		
13	16.6.13*	A2A.NSOA.VAL.CANCEL.CANCELED.SubscriptionVersion	R		
14	16.6.14*	A2A.OSOA.VAL.CANCEL.CONFLICT.SubscriptionVersion	R		
15	16.6.15*	A2A.NSOA.INV.CANCEL.PEND.SubscriptionVersion	O		
16	16.6.16*	A2A.OSOA.INV.CANCEL.CONFLICT.SubscriptionVersion	O		
17	16.6.17*	A2A.NSOA.INV.CANCEL.ACTIVE.SubscriptionVersion	O		
<b>Subscription Version Disconnect Data App-to-App</b>					
1	16.7.1*	A2A.SOA.VAL.IMMDISC.SubscriptionVersion	C		
2	16.7.2*	A2A.SOA.VAL.DEFDISC.SubscriptionVersion	C		
3	16.7.3*	A2A.SOA.VAL.IMMDISC.BYNPAC.SubscriptionVersion	R		
4	16.7.4*	A2A.SOA.VAL.IMMDISC.FAIL.SubscriptionVersion	R		
5	16.7.5*	A2A.SOA.VAL.IMMDISC.PARTFAIL.SubscriptionVersion	R		
6	16.7.6*	A2A.SOA.VAL.IMMDISC.TN-RANGE.SubscriptionVersion	C		
7	16.7.7*	A2A.SOA.INV.IMMDISC.ACT.OLD.SubscriptionVersion	O		
8	16.7.8*	A2A.SOA.INV.IMMDISC.OLD.SubscriptionVersion	O		
9	16.7.9*	A2A.SOA.INV.IMMDISC.FAILED.SubscriptionVersion	O		
10	16.7.10*	A2A.SOA.INV.IMMDISC.OLD.FAILService Provider.SubscriptionVersion	O		
11	16.7.11*	A2A.SOA.VAL.CANCEL.DISPEND.SubscriptionVersion	C		
<b>Subscription Version Conflict Data App-to-App</b>					
1	16.8.1*	A2A.NSOA.VAL.CONFLICT.RESOLV.SubscriptionVersion	R		
2	16.8.2*	A2A.NSOA.VAL.CONFLICT.RESOLV.BYNNSOA.SubscriptionVersion	R		
3	16.8.3*	A2A.OSOA.VAL.CONFLICT.RESOLV.SubscriptionVersion	R		
4	16.8.4*	A2A.OSOA.VAL.CONFLICT.RESOLV.BYOSOA.SubscriptionVersion	C		
5	16.8.5*	A2A.NSOA.VAL.CONFLICT.RESOLVE.TN-RANGE.BYNNSOA.SubscriptionVersion	C		
<b>LSMS App-to-App</b>					
1	16.9.1	A2A.LSMS.VAL.ACTIVATE.BYNPAC.SubscriptionVersion	R		
2	16.9.2	A2A.LSMS.VAL.MODIFY.BYNPAC.ACTIVE.SubscriptionVersion	R		
3	16.9.3	A2A.LSMS.VAL.IMMDISC.BYNPAC.SubscriptionVersion	R		

Test Case Number and Name			Sev	Date	Result
4	16.9.4	A2A.LSMS.VAL.CREATE.MULT.SubscriptionVersion	R		
5	16.9.5	A2A.LSMS.INV.CREATE.MULT.SubscriptionVersion	O		
6	16.9.6	A2A.LSMS.INV.CREATE.UNKNOWN.NPA-NXX.SubscriptionVersion	O		
<b>SOA WSMSC Data App-to-App</b>					
1	16.10.1*	A2A.NSOA.VAL.CREATE.WSMSC.SubscriptionVersion	C		
2	16.10.2*	A2A.NSOA.VAL.MODIFY.WSMSC.SubscriptionVersion	C		
3	16.10.3	A2A.SOA.VAL.QUERY.WSMSC.SubscriptionVersion	O		
<b>LSMS WSMSC Data App-to-App</b>					
1	16.11.1	A2A.LSMS.VAL.CREATE.WSMSC.SubscriptionVersion	C		
2	16.11.2	A2A.LSMS.VAL.CREATE.MULT.WSMSC.SubscriptionVersion	C		
3	16.11.3	A2A.LSMS.VAL.QUERY.SCOPE.WSMSC.SubscriptionVersion	C		
4	16.11.4	A2A.LSMS.VAL.MODIFY.WSMSC.SubscriptionVersion	C		
<b>Subscription Timer and Business Timer App-to-App</b>					
1	16.12.1	A2A.SOA.VAL.QUERY.SUBTIMER.SubscriptionVersion	O		
2	16.12.2	A2A.SOA.VAL.QUERY.BUSTYPE.SubscriptionVersion	O		
3	16.12.3*	A2A.OSOA.VAL.NOT.subscriptionVersionOldSP-ConcurrenceRequest	C		
4	16.12.4*	A2A.OSOA.VAL.NOT.subscriptionVersionOldSPFinalConcurrenceWindowExpiration	C		
5	16.12.5*	A2A.NSOA.VAL.NOT.subscriptionVersionNewSP-CreateRequest	C		
<b>Missing Sending Notification App-to-App</b>					
1	16.13.1*	A2A.NSOA.VAL.ACTIVATE.NOTMISS.SubscriptionVersion	R		
2	16.13.2*	A2A.OSOA.VAL.ACTIVATE.NOTMISS.SubscriptionVersion	R		
3	16.13.3*	A2A.SOA.VAL.MODIFY.ACTIVE.NOTMISS.SubscriptionVersion	R		
4	16.13.4*	A2A.SOA.VAL.IMMDISC.NOTMISS.SubscriptionVersion	R		
<b>Associated Service Provider App-to-App</b>					
1	16.14.1*	A2A.NSOA.VAL.CREATE.FIRST.ASSOCSP.SubscriptionVersion	C		
2	16.14.2*	A2A.NSOA.VAL.CREATE.SECOND.ASSOCSP.SubscriptionVersion	C		
3	16.14.3*	A2A.OSOA.VAL.CREATE.FIRST.ASSOCSP.SubscriptionVersion	C		
4	16.14.4*	A2A.OSOA.VAL.CREATE.SECOND.ASSOCSP.SubscriptionVersion	C		
5	16.14.5*	A2A.OSOA.VAL.NOCONC.ACTIVATE.ASSOCSP	C		

Test Case Number and Name		Sev	Date	Result
	P.SubscriptionVersion			
6	16.14.6* A2A.NSOA.VAL.ACTIVATE.ASSOCSP.SubscriptionVersion	C		
7	16.14.7* A2A.NSOA.VAL.MODIFY.PEND.ASSOCSP.SubscriptionVersion	C		
8	16.14.8* A2A.OSOA.VAL.MODIFY.PEND.ASSOCSP.SubscriptionVersion	C		
9	16.14.9* A2A.SOA.VAL.MODIFY.ACTIVE.ASSOCSP.SubscriptionVersion	C		
10	16.14.10* A2A.NSOA.VAL.CANCEL.ASSOCSP.SubscriptionVersion	C		
11	16.14.11* A2A.OSOA.VAL.CANCEL.ASSOCSP.SubscriptionVersion	C		
12	16.14.12* A2A.NSOA.VAL.CANCEL.ACKREQ.ASSOCSP.SubscriptionVersion	C		
13	16.14.13* A2A.OSOA.VAL.CANCEL.ACKREQ.ASSOCSP.SubscriptionVersion	C		
14	16.14.14* A2A.SOA.VAL.IMMDISC.ASSOCSP.SubscriptionVersion	C		
15	16.14.15* A2A.SOA.VAL.DEFDISC.ASSOCSP.SubscriptionVersion	C		
16	16.14.16* A2A.NSOA.VAL.CONFLICT.RESOLV.ASSOCSP.SubscriptionVersion	C		
17	16.14.17* A2A.OSOA.VAL.CONFLICT.RESOLV.ASSOCSP.SubscriptionVersion	C		
18	16.14.18* A2A.SOA.VAL.PORT-TO-ORIG.ASSOCSP.SubscriptionVersion	C		
19	16.14.19 A2A.SOA.CAP.ACT.ASSOCSP.numberPoolBlockCreateAction	C		
20	16.14.20 A2A.SOA.CAP.OP.SET.ASSOCSP.numberPoolBlockNPAC	C		
<b>Miscellaneous Scenarios Test Cases</b>				
1	16.15.1* A2A.SOA.VAL.MISC.ACTION.resync	C		
2	16.15.2 A2A.SOA.INV.MISC.ACTION.resync	O		
3	16.15.3 A2A.SOA.VAL.MISC.ACTION.ASSOCSP.resync	C		
4	16.15.4 A2A.LSMS.VAL.MISC.ACTION.resync	C		
5	16.15.5 A2A.LSMS.INV.MISC.ACTION.resync	O		
6	<a href="#">16.15.6</a> <a href="#">A2A.SOA.VAL.MISC.ACTION.resync_3_1</a>	<a href="#">C</a>		
<b>Number Pooling – SOA to NPAC SMS</b>				
1	16.16.1 A2A.SOA.VAL.GET.SCOPE.subscriptionVersion.TN-LNPTYPE	O		
<b>Number Pooling – LSMS to NPAC SMS</b>				
1	16.17.1 A2A.LSMS.VAL.GET.SCOPE.subscriptionVersion.TN-LNPTYPE	O		
<b>Number Pooling –NPAC SMS to LSMS</b>				
1	16.18.1 A2A.LSMS.VAL.CREATE.BYNPAC.subscriptionVersion.POOL	C		
2	16.18.2 A2A.LSMS.VAL.CREATE.RANGE.BYNPAC.subscriptionVersion.POOL	C		
3	16.18.3 A2A.LSMS.VAL.GET.SCOPE.BYNPAC.subscriptionVersion.TN-LNPTYPE	C		

\* This test case must be executed twice if the SOA will be supporting both the “individual” and “range/list” versions of notifications.

## Appendix D Standard Regression Test Case Checklist

Test Case Name			Sev	Date	Result
<b>Service Provider</b>					
1	11.6.2	MOC.SOA.CAP.OP.GET.serviceProv	O		
2	13.5.2	MOC.LSMS.CAP.OP.GET.serviceProv	O		
3	11.6.3	MOC.SOA.VAL.SET.SING.serviceProv	C		
4	13.5.3	MOC.LSMS.VAL.SET.SING.serviceProv	C		
<b>Network Data</b>					
1	11.10.3	MOC.SOA.VAL.CRE.AUTO.serviceProvNPA-NXX	C		
2	13.9.3	MOC.LSMS.VAL.CRE.AUTO.serviceProvNPA-NXX	C		
3	11.10.2	MOC.SOA.CAP.OP.DEL.serviceProvNPA-NXX	C		
4	13.9.2	MOC.LSMS.CAP.OP.DEL.serviceProvNPA-NXX	C		
5	11.11.3	MOC.SOA.VAL.CRE.AUTO.serviceProvLRN	C		
6	13.10.3	MOC.LSMS.VAL.CRE.AUTO.serviceProvLRN	C		
7	11.11.2	MOC.SOA.CAP.OP.DEL.serviceProvLRN	C		
8	13.10.2	MOC.LSMS.CAP.OP.DEL.serviceProvLRN	C		
9	14.6.1	MOC.NPAC.CAP.OP.CRE.serviceProvNPA-NXX	R		
10	14.6.2	MOC.NPAC.CAP.OP.DEL.serviceProvNPA-NXX	R		
11	14.7.1	MOC.NPAC.CAP.OP.CRE.serviceProvLRN	R		
12	14.7.2	MOC.NPAC.CAP.OP.DEL.serviceProvLRN	R		
13	12.7.1	MOC.NPAC.SOA.CAP.OP.CRE.serviceProvNPA-NXX-X	C		
14	12.7.2	MOC.NPAC.SOA.CAP.OP.SET.serviceProvNPA-NXX-X	C		
15	12.7.3	MOC.NPAC.SOA.CAP.OP.DEL.serviceProvNPA-NXX-X	C		
16	14.9.1	MOC.NPAC.CAP.OP.CRE.serviceProvNPA-NXX-X	C		
17	14.9.2	MOC.NPAC.CAP.OP.SET.serviceProvNPA-NXX-X	C		
18	14.9.3	MOC.NPAC.CAP.OP.DEL.serviceProvNPA-NXX-X	C		
<b>Subscription Version</b>					
1	11.4.3	MOC.SOA.CAP.ACT.subscriptionVersionOldSP-Create-Initial	R		
2	11.4.2	MOC.SOA.CAP.ACT.subscriptionVersionNewSP-Create-Initial	R		
3	16.3.4	A2A.OSOA.VAL.NOCONC.ACTIVATE.subscriptionVersion	R		
4	16.3.5	A2A.OSOA.VAL.NOCONC.NOACTIVATE.subscriptionVersion	R		
5	11.4.5	MOC.SOA.CAP.ACT.subscriptionVersionOldSP-Create-Second	R		
6	11.4.4	MOC.SOA.CAP.ACT.subscriptionVersionNewSP-Create-Second	R		
7	16.5.1	A2A.NSOA.VAL.MODIFY.PEND.subscriptionVersion	R		
8	16.5.2	A2A.OSOA.VAL.MODIFY.PEND.subscriptionVersion	R		

Test Case Name			Sev	Date	Result
9	16.8.2	A2A.NSOA.VAL.CONFLICT.RESOLV.BYN SOA.s subscriptionVersion	R		
10	16.6.1	A2A.SOA.VAL.CANCEL.subscriptionVersion	R		
11	16.6.4	A2A.OSOA.VAL.CANCEL.subscriptionVersion	R		
12	16.6.2	A2A.NSOA.VAL.CANCEL.BYOSO A.subscription Version	R		
13	16.6.5	A2A.OSOA.VAL.CANCEL.BYN SOA.subscription Version	R		
14	16.6.13	A2A.NSOA.VAL.CANCEL.CANCELED.subscription Version	R		
15	16.6.14	A2A.OSOA.VAL.CANCEL.CONFLICT.subscription Version	R		
16	16.4.2	A2A.NSOA.VAL.ACTIVATE.subscriptionVersion	R		
17	16.3.7	A2A.NSOA.VAL.INTRA-SP- PORT.subscriptionVersion	C		
18	16.3.8	A2A.DSOA.VAL.PORT-TO- ORIG.subscriptionVersion	R		
19	16.4.4	A2A.NSOA.VAL.ACTIVATE.PARTFAIL.subscription Version	R		
20	16.7.1	A2A.SOA.VAL.IMMDISC.subscriptionVersion	C		
21	16.7.2	A2A.SOA.VAL.DEFDISC.subscriptionVersion	C		
22	16.3.1	A2A.NSOA.VAL.CREATE.TN- RANGE.subscriptionVersion	C		
23	16.5.4	A2A.SOA.VAL.MODIFY.ACTIVE.TN- RANGE.subscriptionVersion	C		
24	16.7.6	A2A.SOA.VAL.IMMDISC.TN- RANGE.subscriptionVersion	C		
25	11.8.3	MOC.SOA.CAP.OP.GET.subscriptionVersionNPAC	O		
26	11.8.24	MOC.SOA.INV.QUERY.SCOPE D.subscriptionVersi on	C		
27	16.14.1	A2A.NSOA.VAL.CREATE.FIRST.ASSOCSP.Subsc riptionVersion	C		
28	13.7.1	MOC.LSMS.CAP.OP.GET.subscriptionVersionNPA C	O		
29	14.4.1	MOC.NPAC.CAP.OP.CRE.subscriptionVersion	R		
30	14.2.2	MOC.NPAC.CAP.OP.ACT.InpSubscriptions	R		
31	14.2.3	MOC.NPAC.CAP.OP.NOT.InpSubscriptions	R		
32	14.4.2	MOC.NPAC.CAP.OP.SET subscriptionVersion	R		
33	14.4.7	MOC.NPAC.VAL.SET.SCOP.FILT.subscriptionVersi on	R		
34	14.4.8	MOC.NPAC.VAL.GET.SCOP.FILT.subscriptionVers ion	R		
35	14.4.4	MOC.NPAC.CAP.OP.DEL subscriptionVersion	R		
36	14.4.9	MOC.NPAC.VAL.DEL.SCOP.FILT.subscriptionVers ion	R		
37	13.7.7	MOC.LSMS.INV.QUERY.SCOPE D.subscriptionVer sion	C		
38	16.18.1	A2A.LSMS.VAL.CREATE.BYN PAC.subscriptionVe rsion.POOL	C		
39	16.18.2	A2A.LSMS.VAL.CREATE.RANGE.BYN PAC.subsc riptionVersion.POOL	C		



Test Case Name			Sev	Date	Result
40	16.3.3	A2A.OSOA.VAL.CREATE.TN-RANGE.SubscriptionVersion	C		
41	16.4.11	A2A.NSOA.VAL.ACTIVATE.TN-RANGE.SubscriptionVersion	C		
42	16.5.12	A2A.SOA.VAL.MODIFY.PEND.TN-RANGE.SubscriptionVersion	C		
43	16.6.3	A2A.NSOA.VAL.CANCEL.TN-RANGE.SubscriptionVersion	C		
44	16.6.6	A2A.OSOA.VAL.CANCEL.TN-RANGE.SubscriptionVersion	C		
45	16.8.5	A2A.NSOA.VAL.CONFLICT.RESOLVE.TN-RANGE.BYNSOA.SubscriptionVersion	C		
<b>Number Pool Block</b>					
1	11.4.25	MOC.SOA.CAP.ACT.numberPoolBlockCreateAction	C		
2	11.12.1	MOC.SOA.CAP.OP.GET.numberPoolBlockNPAC	O		
3	11.12.2	MOC.SOA.CAP.OP.SET.numberPoolBlockNPAC	C		
4	13.11.1	MOC.LSMS.CAP.OP.GET.numberPoolBlockNPAC	O		
5	14.8.1	MOC.NPAC.CAP.OP.CRE.numberPoolBlock	C		
6	14.8.2	MOC.NPAC.CAP.OP.SET.numberPoolBlock	C		
7	14.8.3	MOC.NPAC.CAP.OP.GET.numberPoolBlock	C		
8	14.8.4	MOC.NPAC.CAP.OP.GET.MULTIPLE.numberPoolBlock	C		
9	14.8.5	MOC.NPAC.CAP.OP.DEL.numberPoolBlock	C		
10	14.8.6	MOC.NPAC.CAP.OP.SET.SING.numberPoolBlock	C		
11	14.8.7	MOC.NPAC.CAP.OP.SET.MULT.numberPoolBlock	C		
<b>Audit</b>					
1	11.7.1	MOC.SOA.CAP.OP.CRE.subscriptionAudit	C		
2	16.1.5	A2A.SOA.VAL.NODIS.TNRNG.subscripitonAudit	O		
3	16.1.8	A2A.SOA.VAL.WITHDIS.TNRNG.subscripitonAudit	C		
<b>Recovery</b>					
1	16.15.1	A2A.SOA.VAL.MISC.ACTION.resync	C		
2	16.15.4	A2A.LSMS.VAL.MISC.ACTION.resync	C		

\* This test case must be executed twice if the SOA will be supporting both the “individual” and “range/list” versions of notifications.

## Appendix E Release 3.1 Test Case Checklist

Test Case Number and Name			Sev	Date	Result
<b>MOC InpNPAC-SMS – SOA to NPAC SMS</b>					
1	11.1.7*	MOC.SOA.CAP.ACT.InpNotificationRecovery	C		
<b>MOC InpSubscriptions – SOA to NPAC SMS</b>					
1	11.4.27	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRange StatusAttributeValueChange	C		
2	11.4.28	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRange AttributeValueChange	C		
3	11.4.29	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRange ObjectCreation	C		
4	11.4.30	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRange DonorSP-CustomerDisconnectDate	C		
5	11.4.31	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRange CancellationAcknowledgeRequest	C		
6	11.4.32	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRange NewSP-CreateRequest	C		
7	11.4.33	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRange OldSP-ConcurrenceRequest	C		
8	11.4.34	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRange OldSPFinalConcurrenceWindowExpiration	C		
9	11.4.35	MOC.SOA.CAP.NOT.RANGE.subscriptionVersionRange NewSP-FinalCreateWindowExpiration	C		
10	11.4.36	MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeStat usAttributeValueChange	C		
11	<a href="#">11.4.37</a>	<a href="#">MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeAttri buteValueChange</a>	<a href="#">C</a>		
<del>11</del> <del>12</del>	<del>11.4.373</del> <del>8</del>	<del>MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeObj ectCreation</del>	<del>C</del>		
<del>12</del> <del>13</del>	<del>11.4.383</del> <del>9</del>	<del>MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeDon orSP-CustomerDisconnectDate</del>	<del>C</del>		
<del>13</del> <del>14</del>	<del>11.4.394</del> <del>0</del>	<del>MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeCan cellationAcknowledgeRequest</del>	<del>C</del>		
<del>14</del> <del>15</del>	<del>11.4.404</del> <del>1</del>	<del>MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeNew SP-CreateRequest</del>	<del>C</del>		
<del>15</del> <del>16</del>	<del>11.4.414</del> <del>2</del>	<del>MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeOld SP-ConcurrenceRequest</del>	<del>C</del>		
<del>16</del> <del>17</del>	<del>11.4.424</del> <del>3</del>	<del>MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeOld SPFinalConcurrenceWindowExpiration</del>	<del>C</del>		
<del>17</del>	<del>11.4.43</del>	<del>MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeNew SP-FinalCreateWindowExpiration</del>	<del>C</del>		
18	11.4.44	MOC.SOA.CAP.NOT.LIST.subscriptionVersionRangeNew SP-FinalCreateWindowExpiration	C		
19	11.4.45	MOC.SOA.INV.NOT.RANGE.subscriptionVersionRangeS tatusAttributeValueChange	O		
20	11.4.46	MOC.SOA.INV.NOT.RANGE.subscriptionVersionRange AttributeValueChange	O		
21	11.4.47	MOC.SOA.INV.NOT.RANGE.subscriptionVersionRange ObjectCreation	O		
22	11.4.48	MOC.SOA.INV.NOT.RANGE.subscriptionVersionRange DonorSP-CustomerDisconnectDate	O		
23	11.4.49	MOC.SOA.INV.NOT.RANGE.subscriptionVersionRangeC ancellationAcknowledgeRequest	O		
24	11.4.50	MOC.SOA.INV.NOT.RANGE.subscriptionVersionRange	O		

Test Case Number and Name		Sev	Date	Result
	NewSP-CreateRequest			
25	11.4.51 MOC.SOA.INV.NOT.RANGE.subscriptionVersionRange OldSP-ConcurrenceRequest	O		
26	11.4.52 MOC.SOA.INV.NOT.RANGE.subscriptionVersionRange OldSPFinalConcurrenceWindowExpiration	O		
27	11.4.53 MOC.SOA.INV.NOT.RANGE.subscriptionVersionRange NewSP-FinalCreateWindowExpiration	O		
<b>MOC subscriptionVersion – SOA to NPAC SMS</b>				
1	11.8.25 MOC.SOA.CAP.NOT.subscriptionVersionNewSP- FinalCreateWindowExpiration	C		
2	11.8.26 MOC.SOA.INV.NOT.subscriptionVersionNewSP- FinalCreateWindowExpiration	O		
<b>App-to-App Miscellaneous Scenario</b>				
1	16.15.6 A2A.SOA.VAL.MISC.ACTION.resync_3_1	C		

\* This test case must be executed twice if the SOA will be supporting both the “individual” and “range/list” versions of notifications.