

# **DELTA DOCUMENT**

**NPAC SMS**

**Interoperability Test Plan**

**Release 3.3.0**

**Draft #12**

**Supporting NANC IIS Version 3.3.0a**

**Junely 1729, 2005**

# Table of Contents

<b>1</b>	<b>MOC TEST CASES.....</b>	<b>1-1</b>
1.1	NANC 351.....	1-1
1.1.1	MOC.SOA.CAP.ACT.SWIM.lnpNetwork.lnpDownload	1-1
1.1.2	MOC.SOA.CAP.ACT.SWIM.lnpNotificationRecovery	1-2
1.1.3	MOC.SOA.INV.ACT.SWIM.ID.lnpNotificationRecovery	1-3
1.1.4	MOC.SOA.INV.ACT.SWIM.NORM.lnpNetwork.lnpDownload	1-4
1.1.5	MOC.SOA.INV.ACT.SWIM.NORM.lnpNotificationRecovery	1-4
	MOC.LSMS.CAP.ACT.SWIM.lnpNetwork.lnpDownload	1-5
1.1.6	MOC.LSMS.CAP.ACT.SWIM.lnpNotificationRecovery	1-6
1.1.7	MOC.LSMS.CAP.ACT.SWIM.lnpSubscriptions.lnpDownload	1-7
1.1.8	MOC.LSMS.INV.ACT.SWIM.lnpSubscriptions.lnpDownload	1-8
1.1.9	MOC.LSMS.INV.ACT.SWIM.ID.lnpSubscriptions.lnpDownload	1-9
1.1.10	MOC.LSMS.INV.ACT.SWIM.NORM.lnpNetwork.lnpDownload	1-9
1.1.11	MOC.LSMS.INV.ACT.SWIM.NORM.lnpNotificationRecovery	1-9
1.1.12	MOC.LSMS.INV.ACT.SWIM.NORM.lnpSubscriptions.lnpDownload	1-10
1.1.13	MOC.LSMS.VAL.SWIM.lnpDownload-NumberPoolBlock	1-10
1.1.14	MOC.LSMS.INV.ACT.SWIM.NORM.lnpDownload-NumberPoolBlock	1-10
1.2	NANC 388.....	1-11
1.2.1	MOC.SOA.CAP.ACT.UNDOCANPEND.subscriptionVersionModify	1-11
1.2.2		
	MOC.SOA.CAP.NOT.RANGE.UNDOCANPEND.subscriptionVersionRangeStatusAttributeValueChange	1-11
1.2.3		
	MOC.SOA.CAP.NOT.LIST.UNDOCANPEND.subscriptionVersionRangeStatusAttributeValueChange	1-12
1.2.4		
	MOC.SOA.INV.NOT.RANGE.UNDOCANPEND.subscriptionVersionRangeStatusAttributeValueChange	1-12
1.3	NANC 299.....	1-13
1.3.1	MOC.NPAC.CAP.OP.NOT.HEART.lnpNPAC-SMS	1-13
1.3.2	MOC.SOA.CAP.OP.NOT.HEART.lnpSOA	1-13
1.3.3	MOC.LSMS.CAP.OP.NOT.HEART.lnpLocalSMS	1-13
1.4	ILL 130.....	1-14
1.4.1	MOC.SOA.INV.ACT.lnpNotificationRecovery	1-14
1.4.2	MOC.SOA.INV.ACT.lnpRecoveryComplete	1-14
1.4.3	MOC.SOA.INV.ACT.LINK.CRIT.TOO.LARGE.lnpNotificationRecovery	1-14
1.4.4	MOC.SOA.INV.ACT.subscriptionVersionNewSP-Create	1-14
1.4.5	MOC.SOA.INV.ACT.subscriptionVersionOldSP-Create	1-14
1.4.6	MOC.SOA.INV.ACT.subscriptionVersionActivate	1-14
1.4.7	MOC.SOA.INV.ACT.subscriptionVersionModify	1-14
1.4.8	MOC.SOA.INV.ACT.subscriptionVersionCancel	1-14
1.4.9	MOC.SOA.INV.ACT.subscriptionVersionOldSP-CancellationAcknowledge	1-14
1.4.10	MOC.SOA.INV.ACT.subscriptionVersionNewSP-CancellationAcknowledge	1-14
1.4.11	MOC.SOA.INV.ACT.subscriptionVersionDisconnect	1-14
1.4.12	MOC.SOA.INV.ACT.subscriptionVersionRemoveFromConflict	1-15
1.4.13	MOC.SOA.INV.ACT.numberPoolBlockCreateAction	1-15
1.4.14	MOC.SOA.INV.GET.lnpNetwork	1-15
1.4.15	MOC.SOA.INV.ACT.lnpNetwork.lnpDownload	1-15
1.4.16	MOC.SOA.INV.ACT.LINK.CRIT.TOO.LARGE.lnpNetwork.lnpDownload	1-15

1.4.17	MOC.SOA.INV.SET.serviceProv	1-15
1.4.18	MOC.SOA.INV.GET.serviceProv	1-15
1.4.19	MOC.SOA.INV.CRE.subscriptionAudit	1-15
1.4.20	MOC.SOA.INV.DEL.subscriptionAudit	1-15
1.4.21	MOC.SOA.INV.CRE.serviceProvNPA-NXX	1-15
1.4.22	MOC.SOA.INV.DEL.serviceProvNPA-NXX	1-15
1.4.23	MOC.SOA.INV.CRE.serviceProvLRN	1-15
1.4.24	MOC.SOA.INV.DEL.serviceProvLRN	1-16
1.4.25	MOC.SOA.INV.GET.numberPoolBlockNPAC	1-16
1.4.26	MOC.SOA.INV.SET.numberPoolBlockNPAC	1-16
1.4.27	MOC.SOA.INV.GET.SCOP.numberPoolBlockNPAC	1-16
1.4.28	MOC.SOA.INV.GET.serviceProvNPA-NXX-X	1-16
1.4.29	MOC.SOA.INV.GET.SCOP.serviceProvNPA-NXX-X	1-16
1.4.30	MOC.LSMS.INV.GET.InpNPAC-SMS	1-16
1.4.31	MOC.LSMS.INV.ACT.InpNotificationRecovery	1-16
1.4.32	MOC.LSMS.INV.ACT.LINK.CRIT.TOO.LARGE.InpNotificationRecovery	1-16
1.4.33	MOC.LSMS.INV.GET.InpServiceProvs	1-16
1.4.34	MOC.LSMS.INV.ACT.LINK.InpSubscriptions.InpDownload	1-16
1.4.35	MOC.LSMS.INV.ACT.LINK.CRIT.TOO.LARGE.InpSubscriptions.InpDownload	1-16
1.4.36	MOC.LSMS.INV.GET.InpNetwork	1-17
1.4.37	MOC.LSMS.INV.ACT.InpNetwork.InpDownload	1-17
1.4.38	MOC.LSMS.INV.ACT.LINK.CRIT.TOO.LARGE.InpNetwork.InpDownload	1-17
1.4.39	MOC.LSMS.INV.DEL.lsmsFilterNPA-NXX	1-17
1.4.40	MOC.LSMS.INV.GET.serviceProvNetwork	1-17
1.4.41	MOC.LSMS.INV.DEL.serviceProvNPA-NXX	1-17
1.4.42	MOC.LSMS.INV.CRE.LATA.serviceProvNPA-NXX	1-17
1.4.43	MOC.LSMS.INV.DEL.serviceProvLRN	1-17
1.4.44	MOC.LSMS.INV.CRE.LATA.serviceProvLRN	1-17
1.4.45	MOC.LSMS.INV.GET.numberPoolBlockNPAC	1-17
1.4.46	MOC.LSMS.INV.GET.SCOP.numberPoolBlockNPAC	1-17
1.4.47	MOC.LSMS.INV.GET.serviceProvNPA-NXX-X	1-17
1.4.48	MOC.LSMS.INV.GET.SCOP.serviceProvNPA-NXX-X	1-18
1.4.49	MOC.NPAC.INV.NOT.InpSubscriptions	1-18
1.5	NANC 352.....	1-18
1.5.1	MOC.SOA.CAP.ACT.SWIM.InpNetwork.InpDownload	1-18
1.5.2	MOC.SOA.INV.ACT.SWIM.NORM.InpNetwork.InpDownload	1-18
1.5.3	MOC.LSMS.CAP.ACT.SWIM.InpNetwork.InpDownload	1-18
1.5.4	MOC.LSMS.INV.ACT.SWIM.NORM.InpNetwork.InpDownload	1-18
1.5.5	MOC.SOA.CAP.ACT.InpNetwork.InpDownload	1-18
1.5.6	MOC.SOA.CAP.ACT.LINK.InpNetwork.InpDownload	1-18
1.5.7	MOC.LSMS.CAP.ACT.InpNetwork.InpDownload	1-18
1.5.8	MOC.LSMS.CAP.ACT.LINK.InpNetwork.InpDownload	1-19
1.6	NANC 151.....	1-19
1.6.1	MOC.SOA.CAP.ACT.InpNotificationRecovery	1-19
1.6.2	MOC.SOA.CAP.ACT.subscriptionVersionNewSP-Create-Second	1-19
1.6.3	MOC.SOA.CAP.ACT.subscriptionVersionOldSP-Create-Second	1-19
1.6.4	MOC.SOA.CAP.ACT.subscriptionVersionCancel	1-19
1.6.5	MOC.SOA.CAP.ACT.subscriptionVersionOldSP-CancellationAcknowledge	1-19
1.6.6	MOC.SOA.CAP.ACT.subscriptionVersionNewSP-CancellationAcknowledge	1-19
1.6.7	MOC.SOA.CAP.ACT.subscriptionVersionDisconnect	1-19
1.6.8	MOC.SOA.CAP.ACT.subscriptionVersionRemoveFromConflict	1-19
1.6.9	MOC.SOA.CAP.ACT.CONFLICT.subscriptionVersionOldSP-Create-Second	1-20
1.6.10	MOC.SOA.CAP.NOT.subscriptionVersionOldSP-ConcurrenceRequest	1-20
1.6.11	MOC.SOA.CAP.NOT.subscriptionVersionOldSP-FinalConcurrenceWindowExpiration	1-20
1.6.12	MOC.SOA.CAP.NOT.subscriptionVersionNewSP-CreateRequest	1-20
1.6.13	MOC.SOA.CAP.NOT.subscriptionVersionCancellationAcknowledgeRequest	1-20

1.6.14	MOC.SOA.CAP.NOT.subscriptionVersionDonorSP-CustomerDisconnectDate	1-20
1.6.15	MOC.SOA.VAL.NOT.subscriptionVersionStatusAttributeValueChange	1-20
1.6.16	MOC.SOA.CAP.NOT.subscriptionVersionNewSP-FinalConcurrenceWindowExpiration	1-20
1.6.17	MOC.SOA.CAP.NOT.numberPoolBlockAttributeValueChange	1-20
1.6.18	MOC.SOA.CAP.NOT.numberPoolBlockStatusAttributeValueChange	1-20
1.7	NANC 357.....	1-21
1.7.1	MOC.NPAC.SOA.CAP.OP.GET.SPT.serviceProvNetwork	1-21
1.7.2	MOC.NPAC.SOA.CAP.OP.SET.SPT.serviceProvNetwork	1-21
1.7.3	MOC.NPAC.CAP.OP.GET.SPT.serviceProvNetwork	1-21
1.7.4	MOC.NPAC.CAP.OP.SET.SPT.serviceProvNetwork	1-22
1.7.5	MOC.NPAC.SOA.CAP.OP.CRE.serviceProvNetwork	1-22
1.7.6	MOC.SOA.CAP.ACT.InpNetwork.InpDownload	1-22
1.7.7	MOC.SOA.CAP.ACT.LINK.InpNetwork.InpDownload	1-22
1.7.8	MOC.NPAC.CAP.OP.CRE.serviceProvNetwork	1-22
1.7.9	MOC.LSMS.CAP.ACT.InpNetwork.InpDownload	1-22
1.7.10	MOC.LSMS.CAP.ACT.LINK.InpNetwork.InpDownload	1-22
1.8	NANC 285.....	1-23
1.8.1	MOC.SOA.CAP.OP.GET.MAX.InpSubscription	1-23
1.8.2	MOC.LSMS.CAP.OP.GET.MAX.InpSubscription	1-23
<b>2</b>	<b>ASSOCIATION MANAGEMENT TEST CASES.....</b>	<b>2-25</b>
2.1	NANC 386.....	2-25
2.1.1	AMG.SOA.NEW.BIND and AMG.LSMS.NEW.BIND	2-25
<b>3</b>	<b>A2A TEST CASES.....</b>	<b>3-26</b>
3.1	NANC 351.....	3-26
3.1.1	A2A.SOA.VAL.MISC.ACTION.SWIM.resync	3-26
3.1.2	A2A.SOA.VAL.MISC.ACTION.SWIM.ASSOCSP.resync	3-27
3.1.3	A2A.LSMS.VAL.MISC.ACTION.SWIM.resync	3-28
3.2	NANC 388.....	3-30
3.2.1	A2A.SOA.VAL.MODIFY.UNDOCANPEND.SubscriptionVersion	3-30
3.2.2	A2A.SOA.INV.MODIFY.UNDOCANPEND.SubscriptionVersion	3-30
3.2.3	A2A.SOA.VAL.MODIFY.TN-RANGE.UNDOCANPEND.SubscriptionVersion	3-31
3.2.4	A2A.SOA.INV.MODIFY.TN-RANGE.UNDOCANPEND.SubscriptionVersion	3-31
3.2.5	A2A.SOA.VAL.MODIFY.ASSOCSP.UNDOCANPEND.SubscriptionVersion	3-32
3.2.6	A2A.SOA.INV.MODIFY.ASSOCSP.UNDOCANPEND.SubscriptionVersion	3-32
3.3	NANC 299.....	3-33
3.3.1	A2A.NPAC.INV.HEART.NO.RESP.InpNPAC-SMS	3-33
3.4	NANC 352.....	3-33
3.4.1	A2A.SOA.VAL.MISC.ACTION.SWIM.resync	3-33
3.4.2	A2A.SOA.VAL.MISC.ACTION.SWIM.ASSOCSP.resync	3-33
3.4.3	A2A.LSMS.VAL.MISC.ACTION.SWIM.resync	3-33
3.5	NANC 151.....	3-33
3.5.1	A2A.OSOA.VAL.NOCONC.ACTIVATE.SubscriptionVersion	3-34
3.5.2	A2A.OSOA.VAL.NOCONC.NOACTIVATE.SubscriptionVersion	3-34
3.5.3	A2A.OSOA.VAL.CREATE.CONFLICT.SubscriptionVersion	3-34
3.5.4	A2A.DSOA.VAL.PORT-TO-ORIG.SubscriptionVersion	3-34
3.5.5	A2A.NSOA.INV.MISS.INITIAL.CONC.SubscriptionVersion	3-34
3.5.6	A2A.NSOA.INV.STATE-TRANS.PEND-ACTIVE.SubscriptionVersion	3-34
3.5.7	A2A.NSOA.INV.STATE-TRANS.PEND-OLD.SubscriptionVersion	3-34
3.5.8	A2A.OSOA.INV.STATE-TRANS.PEND-OLD.SubscriptionVersion	3-34
3.5.9	A2A.OSOA.INV.STATE-TRANS.PEND-FAILED.SubscriptionVersion	3-34
3.5.10	A2A.DONORSOA.VAL.PORT-TO-ORIG.PTOLISP.SubscriptionVersion	3-34
3.5.11	A2A.SOA.VAL.PORT-TO-ORIG.ASSOCSP.PTOLISP.SubscriptionVersion	3-34
3.5.12	A2A.NSOA.VAL.ACTIVATE.BYNPAC.SubscriptionVersion	3-34
3.5.13	A2A.NSOA.VAL.ACTIVATE.SubscriptionVersion	3-35

3.5.14	A2A.NSOA.VAL.ACTIVATE.FAIL.SubscriptionVersion	3-35
3.5.15	A2A.NSOA.VAL.ACTIVATE.PARTFAIL.SubscriptionVersion	3-35
3.5.16	A2A.OSOA.VAL.ACTIVATE.SubscriptionVersion	3-35
3.5.17	A2A.OSOA.VAL.ACTIVATE.FAIL.SubscriptionVersion	3-35
3.5.18	A2A.OSOA.VAL.ACTIVATE.PARTFAIL.SubscriptionVersion	3-35
3.5.19	A2A.NSOA.ACTIVATE.ACTNOTMISS.SubscriptionVersion	3-35
3.5.20	A2A.NSOA.INV.ACTIVATE.PARTFAIL.SubscriptionVersion	3-35
3.5.21	A2A.OSOA.INV.ACTIVATE.PARTFAIL.SubscriptionVersion	3-35
3.5.22	A2A.NSOA.VAL.ACTIVATE.TN-RANGE.SubscriptionVersion	3-35
3.5.23	A2A.NSOA.VAL.MODIFY.PEND.SubscriptionVersion	3-35
3.5.24	A2A.OSOA.VAL.MODIFY.PEND.SubscriptionVersion	3-35
3.5.25	A2A.SOA.VAL.MODIFY.ACTIVE.SubscriptionVersion	3-36
3.5.26	A2A.SOA.VAL.MODIFY.ACTIVE.TN-RANGE.SubscriptionVersion	3-36
3.5.27	A2A.SOA.VAL.MODIFY.BYNPAC.ACTIVE.SubscriptionVersion	3-36
3.5.28	A2A.SOA.VAL.MODIFY.PARTFAIL.SubscriptionVersion	3-36
3.5.29	A2A.SOA.VAL.MODIFY.FAIL.SubscriptionVersion	3-36
3.5.30	A2A.SOA.INV.MODIFY.PARTFAIL.NOSPLIST.SubscriptionVersion	3-36
3.5.31	A2A.SOA.INV.MODIFY.ACTIVE.NOTMISS.SubscriptionVersion	3-36
3.5.32	A2A.SOA.INV.MODIFY.ATTRSAME.NOTMISS.SubscriptionVersion	3-36
3.5.33	A2A.SOA.VAL.MODIFY.PEND.TN-RANGE.SubscriptionVersion	3-36
3.5.34	A2A.SOA.VAL.MODIFY.DISCONPEND.SubscriptionVersion	3-36
3.5.35	A2A.SOA.VAL.MODIFY.TN-RANGE.DISCONPEND.SubscriptionVersion	3-36
3.5.36	A2A.SOA.VAL.MODIFY.ASSOCSP.DISCONPEND.SubscriptionVersion	3-36
3.5.37	A2A.SOA.VAL.CANCEL.SubscriptionVersion	3-37
3.5.38	A2A.NSOA.VAL.CANCEL.BYOSOA.SubscriptionVersion	3-37
3.5.39	A2A.NSOA.VAL.CANCEL.TN-RANGE.SubscriptionVersion	3-37
3.5.40	A2A.OSOA.VAL.CANCEL.SubscriptionVersion	3-37
3.5.41	A2A.OSOA.VAL.CANCEL.BYNSOA.SubscriptionVersion	3-37
3.5.42	A2A.OSOA.VAL.CANCEL.TN-RANGE.SubscriptionVersion	3-37
3.5.43	A2A.OSOA.VAL.CANCEL.NOCONC.SubscriptionVersion	3-37
3.5.44	A2A.NSOA.VAL.CANCEL.BYNPAC.SubscriptionVersion	3-37
3.5.45	A2A.OSOA.VAL.CANCEL.BYNPAC.SubscriptionVersion	3-37
3.5.46	A2A.NSOA.VAL.CANCEL.ACKREQ.SubscriptionVersion	3-37
3.5.47	A2A.OSOA.VAL.CANCEL.ACKREQ.SubscriptionVersion	3-37
3.5.48	A2A.NSOA.INV.CANCEL.CONFLICT.SubscriptionVersion	3-37
3.5.49	A2A.NSOA.VAL.CANCEL.CANCELED.SubscriptionVersion	3-38
3.5.50	A2A.OSOA.VAL.CANCEL.CONFLICT.SubscriptionVersion	3-38
3.5.51	A2A.NSOA.INV.CANCEL.PEND.SubscriptionVersion	3-38
3.5.52	A2A.OSOA.INV.CANCEL.CONFLICT.SubscriptionVersion	3-38
3.5.53	A2A.NSOA.INV.CANCEL.ACTIVE.SubscriptionVersion	3-38
3.5.54	A2A.SOA.VAL.IMMDISC.SubscriptionVersion	3-38
3.5.55	A2A.SOA.VAL.DEFDISC.SubscriptionVersion	3-38
3.5.56	A2A.SOA.VAL.IMMDISC.BYNPAC.SubscriptionVersion	3-38
3.5.57	A2A.SOA.VAL.IMMDISC.FAIL.SubscriptionVersion	3-38
3.5.58	A2A.SOA.VAL.IMMDISC.PARTFAIL.SubscriptionVersion	3-38
3.5.59	A2A.SOA.VAL.IMMDISC.TN-RANGE.SubscriptionVersion	3-38
3.5.60	A2A.SOA.INV.IMMDISC.ACT.OLD.SubscriptionVersion	3-38
3.5.61	A2A.SOA.INV.IMMDISC.OLD.SubscriptionVersion	3-39
3.5.62	A2A.SOA.INV.IMMDISC.FAILED.SubscriptionVersion	3-39
3.5.63	A2A.SOA.INV.IMMDISC.OLD.FAILService Provider.SubscriptionVersion	3-39
3.5.64	A2A.SOA.VAL.CANCEL.DISCPEND.SubscriptionVersion	3-39
3.5.65	A2A.NSOA.VAL.CONFLICT.RESOLV.SubscriptionVersion	3-39
3.5.66	A2A.NSOA.VAL.CONFLICT.RESOLV.BYNSOA.SubscriptionVersion	3-39
3.5.67	A2A.OSOA.VAL.CONFLICT.RESOLV.SubscriptionVersion	3-39
3.5.68	A2A.OSOA.VAL.CONFLICT.RESOLV.BYOSOA.SubscriptionVersion	3-39
3.5.69	A2A.NSOA.VAL.CONFLICT.RESOLV.TN-RANGE.BYNSOA.SubscriptionVersion	3-39

3.6	NANC 357.....	3-39
3.6.1	A2A.SOA.CAP.OP.SET.ASSOCSP.serviceProv	3-39
3.6.2	A2A.SOA.CAP.OP.GET.ASSOCSP.serviceProv	3-40
<b>APPENDIX B TEST CASE NOMENCLATURE.....</b>		<b>1</b>
Appendix E Release 3.3 Test Case Checklist.....		1

# 1 MOC Test Cases

## 1.1 NANC 351

For each test case related to NANC 351, SWIM Indicator should be set to TRUE, and Linked Replies Blocking Factor should be set to the maximum allowable number to verify that all systems are capable of supporting the maximum amount.

### 1.1.1 MOC.SOA.CAP.ACT.SWIM.InpNetwork.InpDownload

<b>Purpose</b>	<p>To test the SOA's ability to download the serviceProvNPA-NXX, serviceProvNPA-NXX-X (optional data recovered by EDR Service Providers), and serviceProvLRN objects instantiated on the NPAC SMS Simulator and receive them <i>using both SWIM and linked replies</i>. This will be accomplished by the SOA issuing the confirmed M-ACTION request for <i>SWIM-based</i> InpDownload via the InpNetwork object and subsequently handling the NPAC SMS Simulator M-ACTION response(s).</p> <p>This test case must be executed five times, once for no objects (a no data selected response will be returned), once where the number of objects is less than or equal to the associated Blocking Factor (a single non-linked response will be returned), once where the number of objects is greater than the associated Blocking Factor (two or more linked replies will be returned, followed by an empty non-linked response at the end), once where the number of objects is greater than the associated Linked Replies Maximum (the NPAC will provide the swim-more-data indicator), and once where the number of objects is greater than the SWIM maximum.</p>
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the SOA is to support network data recovery <i>using SWIM</i> .
<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.

<p><b>Procedure</b></p>	<ol style="list-style-type: none"> <li>1. SOA sends a <i>SWIM-based</i> InpDownload M-ACTION request with criteria as supported by the product.</li> <li>2. NPAC SMS Simulator responds with an InpDownload M-ACTION response <i>using a SWIM response</i>.</li> <li>3. In the case of no objects, the NPAC SMS Simulator responds with a no data selected response.</li> <li>4. In the case where the number of objects is less than or equal to the associated Blocking Factor, the NPAC SMS Simulator responds with a single non-linked response.</li> <li>5. In the case where the number of objects is greater than the associated Blocking Factor, the NPAC SMS Simulator responds with two or more linked replies (<a href="#">each with a status and action_id</a>), followed by an empty non-linked response.</li> <li>6. In the case where the number of objects is greater than the Linked Replies Maximum, <a href="#">but less than the SWIM maximum</a>, the NPAC SMS Simulator responds with the data using linked replies, plus the swim-more-data indicator <a href="#">and action_id in each reply</a>. The subsequent SOA request must include the <a href="#">action_id</a> from the previous response of the same data type. This is required in order to remove entries on the SWIM list.</li> <li>7. In the case where the number of objects is greater than the SWIM maximum (<a href="#">Linked Replies maximum less than SWIM maximum</a>), the NPAC SMS Simulator responds with the maximum data using linked replies.</li> <li>8. In response to all cases where data is sent from the NPAC SMS Simulator, upon completion of that data type, the SOA sends a swimProcessing-RecoveryResults M-EVENT-REPORT, and includes the <a href="#">action_id</a> from the previous response of the same type. This is required in order to remove entries from the SWIM list.</li> <li>9. NPAC SMS Simulator responds to the M-EVENT-REPORT. In the case where the SWIM maximum was exceeded, the NPAC SMS Simulator returns the error-code and stop-time in the response to the SOA.</li> </ol>
<p><b>Expected Results</b></p>	<p>The SOA sends a valid M-ACTION request and receives the NPAC SMS Simulator M-ACTION response properly <i>using SWIM-based linked replies</i>.</p>



1.1.2 MOC.SOA.CAP.ACT.SWIM.InpNotificationRecovery

<p><b>Purpose</b></p>	<p>Verify SOA can successfully process the InpNotificationRecovery action when the SOA supports <i>both SWIM and linked replies</i>.</p> <p>This test case must be executed five times, once for no objects (a no data selected response will be returned), once where the number of objects is less than or equal to the associated Blocking Factor (a single non-linked response will be returned), once where the number of objects is greater than the associated Blocking Factor (two or more linked replies will be returned, followed by an empty non-linked response at the end), once where the number of objects is greater than the associated Linked Replies Maximum (the NPAC will provide the swim-more-data indicator), and once where the number of objects is greater than the SWIM maximum.</p> <p>This test case must be executed an additional five times if a SOA is supporting both “individual” and “range/list” notifications.</p>
<p><b>Severity</b></p>	<p>C</p>
<p><b>Severity Explanation</b></p>	<p>This test case must be executed if the service provider SOA supports notification recovery <i>using SWIM</i>. The SOA will recover either the “individual” notifications or the “range/list” version of the notifications, and they will be sent <i>using SWIM-based linked replies</i>.</p>
<p><b>Prerequisites</b></p>	<p>Notifications exist for 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”. There are a number of notifications to be recovered for both “individual” subscription version notifications and “range/list” subscription version notifications. Blocking Factors should be set to the maximum allowable number to verify that all systems are capable of supporting the maximum amount.</p>

<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA sends a <i>SWIM-based</i> InpNotificationRecovery action to the NPAC SMS Simulator to start notification data download.</li> <li>2. NPAC SMS Simulator responds with an M-ACTION InpNotificationRecovery response <i>using a SWIM response</i>.</li> <li>3. In the case of no objects, the NPAC SMS Simulator responds with a no data selected response.</li> <li>4. In the case where the number of objects is less than or equal to the associated Blocking Factor, the NPAC SMS Simulator responds with a single non-linked response.</li> <li>5. In the case where the number of objects is greater than the associated Blocking Factor, the NPAC SMS Simulator responds with two or more linked replies (<a href="#">each with a status and action_id</a>), followed by an empty non-linked response.</li> <li>6. In the case where the number of objects is greater than the Linked Replies Maximum, <a href="#">but less than the SWIM maximum</a>, the NPAC SMS Simulator responds with the data using linked replies, plus the swim-more-data indicator <a href="#">and action_id in each reply</a>. The subsequent SOA request must include the <a href="#">action_id</a> from the previous response of the same data type. This is required in order to remove entries on the SWIM list.</li> <li>7. In the case where the number of objects is greater than the SWIM maximum (<a href="#">Linked Replies maximum less than SWIM maximum</a>), the NPAC SMS Simulator responds with the maximum data using linked replies.</li> <li>8. In response to all cases where data is sent from the NPAC SMS Simulator, upon completion of that data type, the SOA sends a swimProcessing-RecoveryResults M-EVENT-REPORT, and includes the <a href="#">action_id</a> from the previous response of the same type. This is required in order to remove entries from the SWIM list.</li> <li>9. NPAC SMS Simulator responds to the M-EVENT-REPORT. In the case where the SWIM maximum was exceeded, the NPAC SMS Simulator returns the error-code and stop-time in the response to the SOA.</li> </ol>
<b>Expected Results</b>	SOA sends the M-ACTION and receives action response <i>using SWIM-based linked replies</i> with the notification data.

### 1.1.3 MOC.SOA.INV.ACT.SWIM.ID.InpNotificationRecovery

<b>Purpose</b>	To test the SOA's ability to handle an error response for the InpNotificationRecovery action related to an invalid action ID, when the SOA supports <i>SWIM</i> .
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the SOA supports notification data recovery <i>using SWIM</i> .
<b>Prerequisites</b>	<a href="#">Notifications exist for each type of notification that can be recovered for the requesting service provider.</a>
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA sends a <i>SWIM-based</i> InpNotificationRecovery M-ACTION request for notification data with criteria as supported by the product, and includes an invalid <a href="#">action_id</a>.</li> <li>2. 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.

1.1.4 MOC.SOA.INV.ACT.SWIM.NORM.InpNetwork.InpDownload

<b>Purpose</b>	Verify SOA can successfully process an error response to the InpDownload action <a href="#">using SWIM</a> , when sent while SOA is associated in normal mode.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the service provider SOA supports network data recovery <a href="#">using SWIM</a> .
<b>Prerequisites</b>	SOA has a valid association to the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA sends <del>the</del> <a href="#">a SWIM-based</a> InpDownload action to the NPAC SMS Simulator to start network data download, while in normal mode.</li> <li>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.

1.1.5 MOC.SOA.INV.ACT.SWIM.NORM.InpNotificationRecovery

<b>Purpose</b>	Verify SOA can successfully process an error response to the InpNotificationRecovery action <a href="#">using SWIM</a> , when sent while SOA is associated in normal mode.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the service provider SOA supports notification data recovery <a href="#">using SWIM</a> .
<b>Prerequisites</b>	SOA has a valid association to the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA sends <del>the</del> <a href="#">a SWIM-based</a> InpNotificationRecovery action to the NPAC SMS Simulator to start notification data download, while in normal mode.</li> <li>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.

MOC.LSMS.CAP.ACT.SWIM.InpNetwork.InpDownload

<b>Purpose</b>	<p>To test the LSMS's ability to download the serviceProvNPA-NXX, serviceProvNPA-NXX-X (optional data recovered by EDR Service Providers), and serviceProvLRN objects instantiated on the NPAC SMS Simulator and receive them <i>using both SWIM and linked replies</i>. This will be accomplished by the LSMS issuing the confirmed M-ACTION request for <i>SWIM-based</i> InpDownload via the InpNetwork object and subsequently handling the NPAC SMS Simulator M-ACTION response(s).</p> <p>This test case must be executed five times, once for no objects (a no data selected response will be returned), once where the number of objects is less than or equal to the associated Blocking Factor (a single non-linked response will be returned), once where the number of objects is greater than the associated Blocking Factor (two or more linked replies will be returned, followed by an empty non-linked response at the end), once where the number of objects is greater than the associated Linked Replies Maximum (the NPAC will provide the swim-more-data indicator), and once where the number of objects is greater than the SWIM maximum.</p>
----------------	---

<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the LSMS is to support network data recovery <i>using SWIM</i> .
<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>1. LSMS sends a <i>SWIM-based</i> InpDownload M-ACTION request with criteria as supported by the product.</li> <li>2. NPAC SMS Simulator responds with an InpDownload M-ACTION response <i>using a SWIM response</i>.</li> <li>3. In the case of no objects, the NPAC SMS Simulator responds with a no data selected response.</li> <li>4. In the case where the number of objects is less than or equal to the associated Blocking Factor, the NPAC SMS Simulator responds with a single non-linked response.</li> <li>5. In the case where the number of objects is greater than the associated Blocking Factor, the NPAC SMS Simulator responds with two or more linked replies, followed by an empty non-linked response.</li> <li>6. In the case where the number of objects is greater than the Linked Replies Maximum, the NPAC SMS Simulator responds with the data using linked replies, plus the swim-more-data indicator. The subsequent LSMS request must include the action_id from the previous response of the same data type. This is required in order to remove entries on the SWIM list.</li> <li>7. In the case where the number of objects is greater than the SWIM maximum, the NPAC SMS Simulator responds with the maximum data using linked replies.</li> <li>8. In response to all cases where data is sent from the NPAC SMS Simulator, upon completion of that data type, the LSMS sends a swimProcessing-RecoveryResults M-EVENT-REPORT, and includes the action_id from the previous response of the same type. This is required in order to remove entries from the SWIM list.</li> <li>9. NPAC SMS Simulator responds to the M-EVENT-REPORT. In the case where the SWIM maximum was exceeded, the NPAC SMS Simulator returns the error-code and stop-time in the response to the LSMS.</li> </ol>
<b>Expected Results</b>	The LSMS sends a valid M-ACTION request and receives the NPAC SMS Simulator M-ACTION response properly <i>using SWIM-based linked replies</i> .

### 1.1.6 MOC.LSMS.CAP.ACT.SWIM.InpNotificationRecovery

<b>Purpose</b>	<p>Verify LSMS can successfully process the InpNotificationRecovery action when the LSMS supports <i>both SWIM and linked replies</i>.</p> <p>This test case must be executed three times, once for no objects (a no data selected response will be returned), once where the number of objects is less than or equal to the associated Blocking Factor (a single non-linked response will be returned), and once where the number of objects is greater than the associated Blocking Factor (two or more linked replies will be returned, followed by an empty non-linked response at the end).</p>
<b>Severity</b>	C

<b>Severity Explanation</b>	This test case must be executed if the service provider LSMS supports notification recovery <i>using SWIM</i> .
<b>Prerequisites</b>	Notifications exist for each type of notification that can be recovered for the requesting service provider. Blocking Factors should be set to the maximum allowable number to verify that all systems are capable of supporting the maximum amount.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a <i>SWIM-based</i> InpNotificationRecovery action to the NPAC SMS Simulator to start notification data download.</li> <li>2. NPAC SMS Simulator responds with an M-ACTION InpNotificationRecovery response <i>using a SWIM response</i>.</li> <li>3. In the case of no objects, the NPAC SMS Simulator responds with a no data selected response.</li> <li>4. In the case where the number of objects is less than or equal to the associated Blocking Factor, the NPAC SMS Simulator responds with a single non-linked response.</li> <li>5. In the case where the number of objects is greater than the associated Blocking Factor, the NPAC SMS Simulator responds with two or more linked replies, followed by an empty non-linked response.</li> <li>6. In response to all cases where data is sent from the NPAC SMS Simulator, upon completion of that data type, the LSMS sends a swimProcessing-RecoveryResults M-EVENT-REPORT, and includes the action_id from the previous response of the same type. This is required in order to remove entries from the SWIM list.</li> <li>7. NPAC SMS Simulator responds to the M-EVENT-REPORT.</li> </ol>
<b>Expected Results</b>	LSMS sends the M-ACTION and receives action response <i>using SWIM-based linked replies</i> with the notification data.

### 1.1.7 MOC.LSMS.CAP.ACT.SWIM.InpSubscriptions.InpDownload

<b>Purpose</b>	<p>To test the LSMS's ability to download the subscriptionVersionNPAC objects instantiated on the NPAC SMS Simulator when the LSMS supports <i>SWIM recovery</i>. This will be accomplished by the LSMS issuing the confirmed M-ACTION request for SWIM-based InpDownload via the InpSubscription object and subsequently handling the NPAC SMS Simulator M-ACTION response(s).</p> <p>This test case must be executed five times, once for no objects (a no data selected response will be returned), once where the number of objects is less than or equal to the associated Blocking Factor (a single non-linked response will be returned), once where the number of objects is greater than the associated Blocking Factor (two or more linked replies will be returned, followed by an empty non-linked response at the end), once where the number of objects is greater than the associated Linked Replies Maximum (the NPAC will provide the swim-more-data indicator), and once where the number of objects is greater than the SWIM maximum.</p>
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the LSMS supports subscription data recovery <i>using SWIM</i> .

<b>Prerequisites</b>	An InpSubscriptions managed object instance has been inherently created. Blocking Factors should be set to the maximum allowable number to verify that all systems are capable of supporting the maximum amount.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a valid <i>SWIM-based</i> InpDownload M-ACTION request.</li> <li>2. NPAC SMS Simulator responds with a successful M-ACTION response <i>using a SWIM response</i>.</li> <li>3. In the case of no objects, the NPAC SMS Simulator responds with a no data selected response.</li> <li>4. In the case where the number of objects is less than or equal to the associated Blocking Factor, the NPAC SMS Simulator responds with a single non-linked response.</li> <li>5. In the case where the number of objects is greater than the associated Blocking Factor, the NPAC SMS Simulator responds with two or more linked replies, followed by an empty non-linked response.</li> <li>6. In the case where the number of objects is greater than the Linked Replies Maximum, the NPAC SMS Simulator responds with the data using linked replies, plus the swim-more-data indicator. The subsequent LSMS request must include the action_id from the previous response of the same data type. This is required in order to remove entries on the SWIM list.</li> <li>7. In the case where the number of objects is greater than the SWIM maximum, the NPAC SMS Simulator responds with the maximum data using linked replies.</li> <li>8. In response to all cases where data is sent from the NPAC SMS Simulator, upon completion of that data type, the LSMS sends a swimProcessing-RecoveryResults M-EVENT-REPORT, and includes the action_id from the previous response of the same type. This is required in order to remove entries from the SWIM list.</li> <li>9. NPAC SMS Simulator responds to the M-EVENT-REPORT. In the case where the SWIM maximum was exceeded, the NPAC SMS Simulator returns the error-code and stop-time in the response to the LSMS.</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 <i>using SWIM-based linked replies</i> .

1.1.8 MOC.LSMS.INV.ACT.SWIM.InpSubscriptions.InpDownload

<b>Purpose</b>	To test the LSMS's ability to handle an error response for the InpDownload action when the LSMS supports <i>both SWIM and linked replies</i> .
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the LSMS supports subscription data recovery <i>using SWIM</i> .
<b>Prerequisites</b>	<a href="#">InpSubscriptions managed object instances exist</a> .
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a <i>SWIM-based</i> InpDownload M-ACTION request for subscription data with criteria as supported by the product.</li> <li>2. NPAC SMS Simulator responds with error status 'failed'.</li> </ol>
<b>Expected Results</b>	The LSMS will correctly handle the error response received from the NPAC SMS Simulator.

1.1.9 MOC.LSMS.INV.ACT.SWIM.ID.InpSubscriptions.InpDownload

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

1.1.10 MOC.LSMS.INV.ACT.SWIM.NORM.InpNetwork.InpDownload

<b>Purpose</b>	Verify LSMS can successfully process an error response to the InpDownload action <i>using SWIM</i> , when sent while LSMS is associated in normal mode.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the service provider LSMS supports network data recovery <i>using SWIM</i> .
<b>Prerequisites</b>	LSMS has a valid association to the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>LSMS sends <del>the</del> a <i>SWIM-based</i> InpDownload action to the NPAC SMS Simulator to start network data download, while in normal mode.</li> <li>NPAC SMS Simulator responds with error status 'failed'.</li> </ol>
<b>Expected Results</b>	LSMS sends the M-ACTION request and receives the action response with the error successfully.

1.1.11 MOC.LSMS.INV.ACT.SWIM.NORM.InpNotificationRecovery

<b>Purpose</b>	Verify LSMS can successfully process an error response to the InpNotificationRecovery action <i>using SWIM</i> , when sent while LSMS is associated in normal mode.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the service provider LSMS supports notification data recovery <i>using SWIM</i> .
<b>Prerequisites</b>	LSMS has a valid association to the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>LSMS sends <del>the</del> a <i>SWIM-based</i> InpNotificationRecovery action to the NPAC SMS Simulator to start notification data download, while in normal mode.</li> <li>NPAC SMS Simulator responds with error status 'failed'.</li> </ol>
<b>Expected Results</b>	LSMS sends the M-ACTION request and receives the action response with the error successfully.

1.1.12 MOC.LSMS.INV.ACT.SWIM.NORM.InpSubscriptions.InpDownload

<b>Purpose</b>	Verify LSMS can successfully process an error response to the InpDownload action <a href="#">using SWIM</a> , when sent while <a href="#">SOA-LSMS</a> is associated in normal mode.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the service provider <a href="#">SOA-LSMS</a> supports subscription data recovery <a href="#">using SWIM</a> .
<b>Prerequisites</b>	LSMS has a valid association to the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends <del>the</del> a <i>SWIM-based</i> InpDownload action to the NPAC SMS Simulator to start subscription data download, while in normal mode.</li> <li>2. NPAC SMS Simulator responds with error status 'failed'.</li> </ol>
<b>Expected Results</b>	LSMS sends the M-ACTION request and receives the action response with the error successfully.

1.1.13 MOC.LSMS.VAL.SWIM.InpDownload-NumberPoolBlock

<b>Purpose</b>	Verify the LSMS's ability to issue the InpDownload action for <i>SWIM-based</i> numberPoolBlock data.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if LSMS will be supporting numberPoolBlock data (i.e., EDR LSMS) <a href="#">using SWIM</a> .
<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 <i>SWIM-based</i> InpDownload M-ACTION request for all numberPoolBlock objects in the LSMS's SWIM list.</li> <li>2. NPAC SMS Simulator responds with a successful M-ACTION response containing the requested data <a href="#">using a SWIM response</a>.</li> </ol>
<b>Expected Results</b>	LSMS issues a valid M-ACTION request and retrieves the data successfully from the NPAC SMS Simulator <a href="#">using a SWIM-based response</a> .

1.1.14 MOC.LSMS.INV.ACT.SWIM.NORM.InpDownload-NumberPoolBlock

<b>Purpose</b>	Verify LSMS can successfully process an error response to the InpDownload action <a href="#">using SWIM</a> , when sent while SOA is associated in normal mode.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the service provider <a href="#">SOA-LSMS</a> supports number pool block data recovery <a href="#">using SWIM</a> .
<b>Prerequisites</b>	LSMS has a valid association to the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends <del>the</del> a <i>SWIM-based</i> InpDownload action to the NPAC SMS Simulator to start number pool block data download, while in normal mode.</li> <li>2. NPAC SMS Simulator responds with error status 'failed'.</li> </ol>
<b>Expected Results</b>	LSMS sends the M-ACTION request and receives the action response with the error successfully.



## 1.2 NANC 388

### 1.2.1 MOC.SOA.CAP.ACT.UNDOCANPEND.subscriptionVersionModify

<b>Purpose</b>	To test the SOA's ability to modify a cancel-pending subscription version, by changing the status from cancel-pending back to pending.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the service provider SOA supports an SV modify that changes the status from cancel-pending back to pending.
<b>Prerequisites</b>	One or more 'cancel-pending' subscription versions exist on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>SOA issues a valid subscriptionVersionModify M-ACTION, specifies either the subscriptionVersionId or subscriptionVersionTN or TN-Range with subscriptionVersionStatus, in order to update the subscriptionVersionStatus attribute. The SOA specifies the new-version-status set to PENDING.</li> <li>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.

### 1.2.2 MOC.SOA.CAP.NOT.RANGE.UNDOCANPEND.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>	One or more 'cancel-pending' subscription versions exist on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>NPAC SMS Simulator issues the subscriptionVersionRangeStatusAttributeValueChange M-EVENT-REPORT, simulating the New or Old Service Provider modify, for one or more 'cancel-pending' subscription versions with consecutive TNs and subscription version Ids, where the status is changed to pending.</li> <li>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.

### 1.2.3 MOC.SOA.CAP.NOT.LIST.UNDOCANPEND.subscriptionVersionRangeStatusAttributeValueChange

<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 subscriptionVersionRangeStatusAttributeValueChange M-EVENT-REPORT.
<b>Prerequisites</b>	Two or more subscription versions exist on the NPAC SMS Simulator with a subscriptionVersionStatus of 'cancel-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 subscriptionVersionRangeStatusAttributeValueChange specifying the subscriptionVersionStatus as 'cancel-pending' for two or more subscription versions with consecutive TNs and non-consecutive subscription version Ids, where the status is changed to pending.</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.

1.2.4 MOC.SOA.INV.NOT.RANGE.UNDOCANPEND.subscriptionVersionRangeStatusAttributeValueChange

<b>Purpose</b>	To test the SOA's ability to accept a subscriptionVersionRangeStatusAttributeValueChange M-EVENT-REPORT with invalid value for the subscriptionVersionStatus attribute.
<b>Severity</b>	O
<b>Severity Explanation</b>	One or more 'cancel-pending' subscription versions exist on the NPAC SMS Simulator.
<b>Prerequisites</b>	One or more 'cancel-pending' subscription versions exist 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 'conflict'.</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.

### 1.3 NANC 299

#### 1.3.1 MOC.NPAC.CAP.OP.NOT.HEART.InpNPAC-SMS

<b>Purpose</b>	Verifies the SOA/LSMS capability to correctly respond to an InpNPAC-SMS MO class M-EVENT-REPORT request for the Heartbeat Notification.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if the SOA/LSMS is supporting the Heartbeat Notification.
<b>Prerequisites</b>	An InpNPAC-SMS instance has been inherently created on the NPAC SMS.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC sends a Heartbeat M-EVENT-REPORT request for InpNPAC-SMS (Heartbeat Notification).</li> <li>2. SOA/LSMS responds with M-EVENT-REPORT confirmation.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives an M-EVENT-REPORT confirmation from the SOA/LSMS.

#### 1.3.2 MOC.SOA.CAP.OP.NOT.HEART.InpSOA

<b>Purpose</b>	Verifies the SOA capability to correctly send an InpSOA MO class M-EVENT-REPORT request for the Heartbeat Notification.
<b>Severity</b>	O
<b>Severity Explanation</b>	Needed for NPAC SMS Simulator to verify correct initiation by SOA for the Heartbeat Notification.
<b>Prerequisites</b>	An InpSOA instance has been inherently created on the SOA.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA sends a Heartbeat M-EVENT-REPORT request for InpSOA (Heartbeat Notification).</li> <li>2. NPAC SMS responds with M-EVENT-REPORT confirmation.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives an M-EVENT-REPORT request from the SOA.

#### 1.3.3 MOC.LSMS.CAP.OP.NOT.HEART.InpLocalSMS

<b>Purpose</b>	Verifies the LSMS capability to correctly send an InpLocalSMS MO class M-EVENT-REPORT request for the Heartbeat Notification.
<b>Severity</b>	O
<b>Severity Explanation</b>	Needed for NPAC SMS Simulator to verify correct initiation by LSMS for the Heartbeat Notification.
<b>Prerequisites</b>	An InpLocalSMS instance has been inherently created on the LSMS.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. LSMS sends a Heartbeat M-EVENT-REPORT request for InpLocalSMS (Heartbeat Notification).</li> <li>2. NPAC SMS responds with M-EVENT-REPORT confirmation.</li> </ol>
<b>Expected Results</b>	The NPAC SMS Simulator receives an M-EVENT-REPORT request from the LSMS.

## **1.4 ILL 130**

Perform the following Test Cases for Error Code details when SOA Application Error Indicator is set to TRUE. Each test should verify that an application specific error code is returned instead of the normal CMIP error code response. For ACTIONS, the SOA must send the new ACTION requests in order to receive the error code in the ACTION response.

### **1.4.1 MOC.SOA.INV.ACT.InpNotificationRecovery**

Existing Test Case, verify a specific error code is sent in M-ACTION Response.

### **1.4.2 MOC.SOA.INV.ACT.InpRecoveryComplete**

Existing Test Case, verify a specific error code is sent in M-ACTION Response.

### **1.4.3 MOC.SOA.INV.ACT.LINK.CRIT.TOO.LARGE.InpNotificationRecovery**

Existing Test Case, verify a specific error code is sent in M-ACTION Response.

### **1.4.4 MOC.SOA.INV.ACT.subscriptionVersionNewSP-Create**

Existing Test Case, verify a specific error code is sent in M-ACTION Response.

### **1.4.5 MOC.SOA.INV.ACT.subscriptionVersionOldSP-Create**

Existing Test Case, verify a specific error code is sent in M-ACTION Response.

### **1.4.6 MOC.SOA.INV.ACT.subscriptionVersionActivate**

Existing Test Case, verify a specific error code is sent in M-ACTION Response, [when SOA sends M-ACTION Request of subscriptionVersionActivateWithErrorCode](#).

### **1.4.7 MOC.SOA.INV.ACT.subscriptionVersionModify**

Existing Test Case, verify a specific error code is sent in M-ACTION Response.

### **1.4.8 MOC.SOA.INV.ACT.subscriptionVersionCancel**

Existing Test Case, verify a specific error code is sent in M-ACTION Response, [when SOA sends M-ACTION Request of subscriptionVersionCancelWithErrorCode](#).

### **1.4.9 MOC.SOA.INV.ACT.subscriptionVersionOldSP-CancellationAcknowledge**

Existing Test Case, verify a specific error code is sent in M-ACTION Response, [when SOA sends M-ACTION Request of subscriptionVersionOldSP-CancellationAcknowledgeWithErrorCode](#).

### **1.4.10 MOC.SOA.INV.ACT.subscriptionVersionNewSP-CancellationAcknowledge**

Existing Test Case, verify a specific error code is sent in M-ACTION Response, [when SOA sends M-ACTION Request of subscriptionVersionNewSP-CancellationAcknowledgeWithErrorCode](#).

#### 1.4.11 MOC.SOA.INV.ACT.subscriptionVersionDisconnect

Existing Test Case, verify a specific error code is sent in M-ACTION Response.

#### 1.4.12 MOC.SOA.INV.ACT.subscriptionVersionRemoveFromConflict

Existing Test Case, verify a specific error code is sent in M-ACTION Response, [when SOA sends M-ACTION Request of subscriptionVersionRemoveFromConflictWithErrorCode](#).

#### 1.4.13 MOC.SOA.INV.ACT.numberPoolBlockCreateAction

Existing Test Case, verify a specific error code is sent in M-ACTION Response.

#### 1.4.14 MOC.SOA.INV.GET.InpNetwork

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### 1.4.15 MOC.SOA.INV.ACT.InpNetwork.InpDownload

Existing Test Case, verify a specific error code is sent in the M-ACTION response.

#### 1.4.16 MOC.SOA.INV.ACT.LINK.CRIT.TOO.LARGE.InpNetwork.InpDownload

Existing Test Case, verify a specific error code is sent in M-ACTION Response.

#### 1.4.17 MOC.SOA.INV.SET.serviceProv

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### 1.4.18 MOC.SOA.INV.GET.serviceProv

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### 1.4.19 MOC.SOA.INV.CRE.subscriptionAudit

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### 1.4.20 MOC.SOA.INV.DEL.subscriptionAudit

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### 1.4.21 MOC.SOA.INV.CRE.serviceProvNPA-NXX

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### 1.4.22 MOC.SOA.INV.DEL.serviceProvNPA-NXX

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### **1.4.23 MOC.SOA.INV.CRE.serviceProvLRN**

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### **1.4.24 MOC.SOA.INV.DEL.serviceProvLRN**

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### **1.4.25 MOC.SOA.INV.GET.numberPoolBlockNPAC**

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### **1.4.26 MOC.SOA.INV.SET.numberPoolBlockNPAC**

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### **1.4.27 MOC.SOA.INV.GET.SCOP.numberPoolBlockNPAC**

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### **1.4.28 MOC.SOA.INV.GET.serviceProvNPA-NXX-X**

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### **1.4.29 MOC.SOA.INV.GET.SCOP.serviceProvNPA-NXX-X**

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### **1.4.30 MOC.LSMS.INV.GET.InpNPAC-SMS**

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### **1.4.31 MOC.LSMS.INV.ACT.InpNotificationRecovery**

Existing Test Case, verify a specific error code is sent in M-ACTION Response.

#### **1.4.32 MOC.LSMS.INV.ACT.LINK.CRIT.TOO.LARGE.InpNotificationRecovery**

Existing Test Case, verify a specific error code is sent in M-ACTION Response.

#### **1.4.33 MOC.LSMS.INV.GET.InpServiceProvs**

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### **1.4.34 MOC.LSMS.INV.ACT.LINK.InpSubscriptions.InpDownload**

Existing Test Case, verify a specific error code is sent in M-ACTION Response.

#### 1.4.35 MOC.LSMS.INV.ACT.LINK.CRIT.TOO.LARGE.InpSubscriptions.InpDownload

Existing Test Case, verify a specific error code is sent in M-ACTION Response.

#### 1.4.36 MOC.LSMS.INV.GET.InpNetwork

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### 1.4.37 MOC.LSMS.INV.ACT.InpNetwork.InpDownload

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### 1.4.38 MOC.LSMS.INV.ACT.LINK.CRIT.TOO.LARGE.InpNetwork.InpDownload

Existing Test Case, verify a specific error code is sent in M-ACTION Response.

#### 1.4.39 MOC.LSMS.INV.DEL.IsmsFilterNPA-NXX

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### 1.4.40 [MOC.LSMS.INV.SET.serviceProv](#)

[Existing Test Case, verify a specific error code is sent in the processingFailure response.](#)

#### 1.4.41 MOC.LSMS.INV.GET.serviceProvNetwork

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### 1.4.42 MOC.LSMS.INV.DEL.serviceProvNPA-NXX

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### 1.4.43 MOC.LSMS.INV.CRE.LATA.serviceProvNPA-NXX

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### 1.4.44 MOC.LSMS.INV.DEL.serviceProvLRN

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### 1.4.45 MOC.LSMS.INV.CRE.LATA.serviceProvLRN

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### 1.4.46 MOC.LSMS.INV.GET.numberPoolBlockNPAC

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### 1.4.47 MOC.LSMS.INV.GET.SCOP.numberPoolBlockNPAC

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### 1.4.48 MOC.LSMS.INV.GET.serviceProvNPA-NXX-X

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### 1.4.49 MOC.LSMS.INV.GET.SCOP.serviceProvNPA-NXX-X

Existing Test Case, verify a specific error code is sent in the processingFailure response.

#### 1.4.50 MOC.NPAC.INV.NOT.InpSubscriptions

Existing Test Case, verify a specific error code is sent in the processingFailure response.

### 1.5 **NANC 352**

#### 1.5.1 MOC.SOA.CAP.ACT.SWIM.InpNetwork.InpDownload

Perform NANC 351, TC 1.1.1, for Service Provider data, when both SOA SP Data Indicator is set to TRUE and SOA SWIM Indicator is set to TRUE. Test case should be executed multiple times to test time range and SPID selection.

#### 1.5.2 MOC.SOA.INV.ACT.SWIM.NORM.InpNetwork.InpDownload

Perform NANC 351, TC 1.1.4, for Service Provider data, when both SOA SP Data Indicator is set to TRUE and SOA SWIM Indicator is set to TRUE.

#### 1.5.3 MOC.LSMS.CAP.ACT.SWIM.InpNetwork.InpDownload

Perform NANC 351, TC 1.1.6, for Service Provider data, when both LSMS SP Data Indicator is set to TRUE and LSMS SWIM Indicator is set to TRUE. Test case should be executed multiple times to test time range and SPID selection.

#### 1.5.4 MOC.LSMS.INV.ACT.SWIM.NORM.InpNetwork.InpDownload

Perform NANC 351, TC 1.1.10, for Service Provider data, when both SOA SP Data Indicator is set to TRUE and SOA SWIM Indicator is set to TRUE.

Perform the following Test Cases for Service Provider recovery data when SOA SP Data Indicator is set to TRUE. Each test should verify that SP data is returned in the recovery response.

#### 1.5.5 MOC.SOA.CAP.ACT.InpNetwork.InpDownload

Existing Test Case, verify Service Provider data is requested and recovered in the M-ACTION Request/Response.



### 1.5.6 MOC.SOA.CAP.ACT.LINK.InpNetwork.InpDownload

Existing Test Case, verify Service Provider data is requested and recovered in the M-ACTION Request/Response.

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

Existing Test Case, verify Service Provider data is requested and recovered in the M-ACTION Request/Response.

### 1.5.8 MOC.LSMS.CAP.ACT.LINK.InpNetwork.InpDownload

Existing Test Case, verify Service Provider data is requested and recovered in the M-ACTION Request/Response.

## 1.6 **NANC 151**

Perform the following Test Cases for TN attribute when SOA TN Attribute Indicator is set to TRUE. Each test should verify that the TN/TN-Range/NPA-NXX-X attribute is provided.

### 1.6.1 MOC.SOA.CAP.ACT.InpNotificationRecovery

Existing Test Case, verify that the TN/TN-Range/NPA-NXX-X attribute is provided in the recovery of SV and NPB AVC and SAVC notifications.

### 1.6.2 MOC.SOA.CAP.ACT.subscriptionVersionNewSP-Create-Second

Existing Test Case, verify that the TN attribute is provided in the notification.

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

Existing Test Case, verify that the TN attribute is provided in the notification.

### 1.6.4 MOC.SOA.CAP.ACT.subscriptionVersionCancel

Existing Test Case, verify that the TN attribute is provided in the notification.

### 1.6.5 MOC.SOA.CAP.ACT.subscriptionVersionOldSP-CancellationAcknowledge

Existing Test Case, verify that the TN attribute is provided in the notification.

### 1.6.6 MOC.SOA.CAP.ACT.subscriptionVersionNewSP-CancellationAcknowledge

Existing Test Case, verify that the TN attribute is provided in the notification.

### 1.6.7 MOC.SOA.CAP.ACT.subscriptionVersionDisconnect

Existing Test Case, verify that the TN attribute is provided in the notification.

**1.6.8 MOC.SOA.CAP.ACT.subscriptionVersionRemoveFromConflict**

Existing Test Case, verify that the TN attribute is provided in the notification.

**1.6.9 MOC.SOA.CAP.ACT.CONFLICT.subscriptionVersionOldSP-Create-Second**

Existing Test Case, verify that the TN attribute is provided in the notification.

**1.6.10 ~~MOC.SOA.CAP.NOT.subscriptionVersionOldSP-ConcurrenceRequest~~**

~~Existing Test Case, verify that the TN attribute is provided in the notification.~~

**1.6.11 ~~MOC.SOA.CAP.NOT.subscriptionVersionOldSP-FinalConcurrenceWindowExpiration~~**

~~Existing Test Case, verify that the TN attribute is provided in the notification.~~

**1.6.12 ~~MOC.SOA.CAP.NOT.subscriptionVersionNewSP-CreateRequest~~**

~~Existing Test Case, verify that the TN attribute is provided in the notification.~~

**1.6.13 ~~MOC.SOA.CAP.NOT.subscriptionVersionCancellationAcknowledgeRequest~~**

~~Existing Test Case, verify that the TN attribute is provided in the notification.~~

**1.6.14 ~~MOC.SOA.CAP.NOT.subscriptionVersionDonorSP-CustomerDisconnectDate~~**

~~Existing Test Case, verify that the TN attribute is provided in the notification.~~

**1.6.15 ~~MOC.SOA.VAL.NOT.subscriptionVersionStatusAttributeValueChange~~**

~~Existing Test Case, verify that the TN attribute is provided in the notification.~~

**1.6.16 ~~MOC.SOA.CAP.NOT.subscriptionVersionNewSP-FinalConcurrenceWindowExpiration~~**

~~Existing Test Case, verify that the TN attribute is provided in the notification.~~

**1.6.17 MOC.SOA.CAP.NOT.numberPoolBlockAttributeValueChange**

Existing Test Case, verify that the NPA-NXX-X attribute is provided in the notification.

**1.6.18 MOC.SOA.CAP.NOT.numberPoolBlockStatusAttributeValueChange**

Existing Test Case, verify that the NPA-NXX-X attribute is provided in the notification.

## 1.7 NANC 357

### 1.7.1 MOC.NPAC.SOA.CAP.OP.GET.SPT.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, including the SP Type attribute.
<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, and the optional SP Type attribute.
<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, including the optional SP Type attribute.

### 1.7.2 MOC.NPAC.SOA.CAP.OP.SET.SPT.serviceProvNetwork

<b>Purpose</b>	Tests the capability of the SOA to support the M-SET of the SP Type 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, and the optional SP Type attribute.
<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 SP Type 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.

### 1.7.3 MOC.NPAC.CAP.OP.GET.SPT.serviceProvNetwork

<b>Purpose</b>	Tests the capability of the LSMS 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, including the SP Type attribute.
<b>Severity</b>	O
<b>Severity Explanation</b>	Needed for the NPAC SMS Simulator to verify correct instantiation by LSMS if the SOA is supporting network data download, and the optional SP Type attribute.
<b>Prerequisites</b>	A serviceProvNetwork instance exists on LSMS.
<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. LSMS responds with getResult.</li> </ol>

<b>Expected Results</b>	The NPAC SMS Simulator receives a getResult with all attribute values, including the optional SP Type attribute.
-------------------------	--

#### 1.7.4 MOC.NPAC.CAP.OP.SET.SPT.serviceProvNetwork

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

Perform the following Test Cases for SP Type attribute when SOA SP Type Attribute Indicator is set to TRUE. Each test should verify that the SP Type attribute is provided.

#### 1.7.5 MOC.NPAC.SOA.CAP.OP.CRE.serviceProvNetwork

Existing Test Case, verify that the SP Type attribute is provided in the M-CREATE Request.

#### 1.7.6 MOC.SOA.CAP.ACT.InpNetwork.InpDownload

Existing Test Case, verify that the SP Type attribute is provided in the M-ACTION Response.

#### 1.7.7 MOC.SOA.CAP.ACT.LINK.InpNetwork.InpDownload

Existing Test Case, verify that the SP Type attribute is provided in the M-ACTION Response.

#### 1.7.8 MOC.NPAC.CAP.OP.CRE.serviceProvNetwork

Existing Test Case, verify that the SP Type attribute is provided in the M-CREATE Request.

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

Existing Test Case, verify that the SP Type attribute is provided in the M-ACTION Response.

#### 1.7.10 MOC.LSMS.CAP.ACT.LINK.InpNetwork.InpDownload

Existing Test Case, verify that the SP Type attribute is provided in the M-ACTION Response.

## 1.8 NANC 285

### 1.8.1 MOC.SOA.CAP.OP.GET.MAX.InpSubscription

<b>Purpose</b>	To test the SOA's ability to GET all the attributes of the InpSubscriptions managed object instance, when the amount of data exceeds the maximum query size, and the SOA supports enhanced query capability (SOA SV Query Indicator).
<b>Severity</b>	O
<b>Severity Explanation</b>	Does not impact ability to provide LNP service. SOA may perform to validate InpSubscriptions object.
<b>Prerequisites</b>	An InpSubscriptions 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 multiple InpSubscriptions objects.</li> <li>2. NPAC SMS Simulator responds with a successful M-GET result containing all the attributes, of the number of objects equal to the SV Query Maximum.</li> <li>3. SOA sends a SECOND valid M-GET request to retrieve all attributes of multiple InpSubscriptions objects, greater than the last InpSubscriptions object returned from the first M-GET request.</li> <li>4. NPAC SMS Simulator responds with a successful M-GET result containing all the attributes, of the subsequent InpSubscriptions objects.</li> <li>5. SOA continues to send a valid M-GET request to retrieve all attributes of multiple InpSubscriptions objects, greater than the last InpSubscriptions object returned from the previous M-GET request.</li> <li>6. NPAC SMS Simulator continues to respond with a successful M-GET result containing all the attributes, of the subsequent InpSubscriptions objects. Once all data has been provided, the last response will contain no additional data.</li> <li>7. SOA receives an M-GET result with no data. This is the indication that all data has been successfully delivered from the NPAC SMS Simulator.</li> </ol>
<b>Expected Results</b>	The SOA issues a valid M-GET request, retrieves the attributes successfully from the NPAC SMS Simulator and correctly handles the response. The SOA uses the last object of the first response to determine the starting point for the second M-GET request. The response is successfully handled. This continues until the SOA receives an empty GET response, indicating all data has been delivered.

### 1.8.2 MOC.LSMS.CAP.OP.GET.MAX.InpSubscription

<b>Purpose</b>	To test the LSMS's ability to GET all the attributes of the InpSubscriptions managed object instance, when the amount of data exceeds the maximum query size, and the LSMS supports enhanced query capability (LSMS SV Query Indicator).
<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>	An InpSubscriptions 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 to retrieve all attributes of multiple InpSubscriptions objects.</li> <li>2. NPAC SMS Simulator responds with a successful M-GET result containing all the attributes, of the number of objects equal to the SV Query Maximum.</li> <li>3. LSMS sends a SECOND valid M-GET request to retrieve all attributes of multiple InpSubscriptions objects, greater than the last InpSubscriptions object returned from the first M-GET request.</li> <li>4. NPAC SMS Simulator responds with a successful M-GET result containing all the attributes, of the subsequent InpSubscriptions objects.</li> <li>5. LSMS continues to send a valid M-GET request to retrieve all attributes of multiple InpSubscriptions objects, greater than the last InpSubscriptions object returned from the previous M-GET request.</li> <li>6. NPAC SMS Simulator continues to respond with a successful M-GET result containing all the attributes, of the subsequent InpSubscriptions objects. Once all data has been provided, the last response will contain no additional data.</li> <li>7. LSMS receives an M-GET result with no data. This is the indication that all data has been successfully delivered from the NPAC SMS Simulator.</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. The LSMS uses the last object of the first response to determine the starting point for the second M-GET request. The response is successfully handled. This continues until the LSMS receives an empty GET response, indicating all data has been delivered.

## 2 Association Management Test Cases

### 2.1 NANC 386

#### 2.1.1 AMG.SOA.NEW.BIND and AMG.LSMS.NEW.BIND

<b>Purpose</b>	To verify that the SOA/LSMS handles an association abort error message when a second association bind request is received, and the first association is still active.
<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>• System clocks are synchronized.</li> </ul>
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. SOA/LSMS issues another association request (AARQ).</li> <li>2. NPAC SMS Simulator accepts association request and issues an association response with errorCode = new-bind-received.</li> <li>3. SOA/LSMS receives and correctly handles response.</li> </ol>
<b>Expected Results</b>	First association is aborted, and second association is successfully established with NPAC SMS Simulator.

### 3 A2A Test Cases

#### 3.1 NANC 351

For each test case related to NANC 351, SWIM Indicator should be set to TRUE, and Linked Replies Blocking Factor should be set to the maximum allowable number to verify that all systems are capable of supporting the maximum amount.

##### 3.1.1 A2A.SOA.VAL.MISC.ACTION.SWIM.resync

<b>Purpose</b>	Verify SOA can process resynchronization updates <i>using SWIM</i> 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 SP, network and/or notification data recovery <i>using SWIM</i> .
<b>Prerequisites</b>	SP, 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 SP data recovery, sends the <i>SWIM-based</i> InpDownload action request to NPAC SMS Simulator to start SP data download.</li> <li>3. NPAC SMS Simulator responds with SP data updates <i>using a SWIM response</i>.</li> <li>4. SOA, if supporting network data recovery, sends the <i>SWIM-based</i> InpDownload action request to NPAC SMS Simulator to start network data download.</li> <li>5. NPAC SMS Simulator responds with network data updates <i>using a SWIM response</i>.</li> <li>6. SOA, if supporting notification data recovery, sends the <i>SWIM-based</i> InpNotificationRecovery action request to NPAC SMS Simulator to start notification data download.</li> <li>7. NPAC SMS Simulator responds with notification updates <i>using a SWIM response</i>.</li> <li>8. Upon completion for each type of data, the SOA sends a swimProcessing-RecoveryResults M-EVENT-REPORT, and includes the action_id from the previous response of the same data type. This is required in order to remove entries from the SWIM list.</li> <li>9. NPAC SMS Simulator responds to the M-EVENT-REPORT. In the case where the SWIM maximum is exceeded, the NPAC SMS Simulator returns the error-code and stop-time in the response to the SOA.</li> <li>10. If the NPAC SMS Simulator responds with a stop-time for any of the responses, the SOA will perform normal recovery for that type of data, using the SWIM stop-time as the normal recovery start time.</li> <li>11. SOA sends InpRecoveryComplete action request to NPAC SMS Simulator to set the resynchronization flag off.</li> <li>12. NPAC SMS Simulator sends the action response.</li> </ol>



<b>Expected Results</b>	SOA associates in recovery mode, issues SWIM-based data download and/or notification recovery actions, and receives action responses containing SP, network and/or notification data updates.
-------------------------	---

3.1.2 A2A.SOA.VAL.MISC.ACTION.SWIM.ASSOCSP.resync

<b>Purpose</b>	Verify SOA can process resynchronization updates <i>using SWIM</i> 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 SP, network and/or notification data recovery <i>using SWIM</i> for an associated service provider.
<b>Prerequisites</b>	SP, 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 SP data recovery, sends the SWIM-based InpDownload action request to NPAC SMS Simulator to start SP data download at a specified period of time for the associated service provider <i>using a SWIM response</i>.</li> <li>3. NPAC SMS Simulator responds with SP data updates <i>using a SWIM response</i>.</li> <li>4. SOA, if supporting network data recovery, sends the SWIM-based InpDownload action request to NPAC SMS Simulator to start network data download at a specified period of time for the associated service provider <i>using a SWIM response</i>.</li> <li>5. NPAC SMS Simulator responds with network data updates <i>using a SWIM response</i>.</li> <li>6. SOA, if supporting notification data recovery, sends the SWIM-based 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>7. NPAC SMS Simulator responds with notification updates <i>using a SWIM response</i>.</li> <li>8. Upon completion for each type of data, the SOA sends a swimProcessing-RecoveryResults M-EVENT-REPORT, and includes the action_id from the previous response of the same data type. This is required in order to remove entries from the SWIM list.</li> <li>9. NPAC SMS Simulator responds to the M-EVENT-REPORT. In the case where the SWIM maximum is exceeded, the NPAC SMS Simulator returns the error-code and stop-time in the response to the SOA.</li> <li>10. If the NPAC SMS Simulator responds with a stop-time for any of the responses, the SOA will perform normal recovery for that type of data, using the SWIM stop-time as the normal recovery start time.</li> <li>11. SOA sends the InpRecoveryComplete action request to the NPAC SMS Simulator to set the resynchronization flag off.</li> <li>12. NPAC SMS Simulator responds to the action.</li> </ol>
<b>Expected Results</b>	SOA associates in recovery mode, issues SWIM-based data download and notification recovery actions, and receives action responses containing SP, network and notification data updates time for the associated service.

### 3.1.3 A2A.LSMS.VAL.MISC.ACTION.SWIM.resync

<b>Purpose</b>	Verify LSMS can process resynchronization updates <i>using SWIM</i> from NPAC SMS Simulator.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if an LSMS is to support SP, network, subscription, number pool block, and/or notification data recovery <i>using SWIM</i> .
<b>Prerequisites</b>	SP, network, subscription, number pool block and notification data exist to recover.

<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 supporting SP data recovery, sends the SWIM-based InpDownload action request to NPAC SMS Simulator to start SP data download for a specified period of time.</li> <li>3. NPAC SMS Simulator responds with SP data updates <i>using a SWIM response</i>.</li> <li>4. LSMS, if supporting network data recovery, sends the SWIM-based InpDownload action request to NPAC SMS Simulator to start network data download for a specified period of time.</li> <li>5. NPAC SMS Simulator responds with network data updates <i>using a SWIM response</i>.</li> <li>6. LSMS, if supporting subscription data recovery, sends the SWIM-based InpDownload action request to NPAC SMS Simulator to start subscription data download for a specified period of time.</li> <li>7. NPAC SMS Simulator responds with subscription data updates <i>using a SWIM response</i>.</li> <li>8. LSMS, if supporting number pool block data recovery, sends the SWIM-based InpDownload action request to NPAC SMS Simulator to start number pool block data download for a specified period of time.</li> <li>9. NPAC SMS Simulator responds with number pool block data updates <i>using a SWIM response</i>.</li> <li>10. LSMS, if supporting notification data recovery, sends the SWIM-based InpNotificationRecovery action request to NPAC SMS Simulator to start notification data download for a specified period of time.</li> <li>11. NPAC SMS Simulator responds with notification updates <i>using a SWIM response</i>.</li> <li>12. Upon completion for each type of data, the LSMS sends a swimProcessing-RecoveryResults M-EVENT-REPORT, and includes the action_id from the previous response of the same data type. This is required in order to remove entries from the SWIM list.</li> <li>13. NPAC SMS Simulator responds to the M-EVENT-REPORT. In the case where the SWIM maximum is exceeded, the NPAC SMS Simulator returns the error-code and stop-time in the response to the LSMS.</li> <li>14. If the NPAC SMS Simulator responds with a stop-time for any of the responses, the LSMS will perform normal recovery for that type of data, using the SWIM stop-time as the normal recovery start time.</li> <li>15. LSMS sends InpRecoveryComplete action request to NPAC SMS Simulator to set the resynchronization flag off.</li> <li>16. NPAC SMS Simulator sends the action response.</li> </ol>
<p><b>Expected Results</b></p>	<p>LSMS associates in recovery mode, issues SWIM-based data download and/or notification recovery actions, and receives action responses containing SP, network, subscription, number pool block and/or notification data updates.</p>

### 3.2 NANC 388

#### 3.2.1 A2A.SOA.VAL.MODIFY.UNDOCANPEND.SubscriptionVersion

<b>Purpose</b>	Verify a SOA can modify a cancel-pending subscription version on the NPAC SMS Simulator, by changing the status from cancel-pending back to pending.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the service provider SOA supports an SV modify that changes the status from cancel-pending back to pending.
<b>Prerequisites</b>	One or more 'cancel-pending' subscription versions exist on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues an M-ACTION subscriptionVersionModify to NPAC SMS Simulator, in order to update the subscriptionVersionStatus attribute.</li> <li>2. The NPAC SMS Simulator locally sets the subscriptionVersionNPAC attribute value for the instance to be modified and responds to the M-ACTION.</li> <li>3. NPAC SMS Simulator issues the subscriptionVersionStatusAttributeValueChange notification containing the updated attribute.</li> <li>4. 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 attribute for the instance is modified.

#### 3.2.2 A2A.SOA.INV.MODIFY.UNDOCANPEND.SubscriptionVersion

<b>Purpose</b>	To verify a SOA can handle the error condition for the modification of a cancel-pending subscription version, and the subscription version is not updated in response to a modify cancel-pending version scenario.
<b>Severity</b>	C
<b>Severity Explanation</b>	This test case must be executed if the service provider SOA supports an SV modify that changes the status from cancel-pending back to pending.
<b>Prerequisites</b>	Two or more 'cancel-pending' subscription versions exist on the NPAC SMS Simulator.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. The SOA issues an M-ACTION subscriptionVersionModify to NPAC SMS Simulator, in order to update the subscriptionVersionStatus attribute.</li> <li>2. The 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.

3.2.3 A2A.SOA.VAL.MODIFY.TN-RANGE.UNDOCANPEND.SubscriptionVersion

<b>Purpose</b>	Verify a SOA can modify cancel-pending subscription versions on the NPAC SMS Simulator using a TN range. 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>	Two or more ‘cancel-pending’ subscription versions exist on the NPAC SMS Simulator.
<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 status attribute values for the instances to be modified from cancel-pending back to pending, and responds to the M-ACTION.</li> <li>3. The NPAC SMS Simulator sends the 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 status attribute for all the instances in the TN range are modified from cancel-pending back to pending.

3.2.4 A2A.SOA.INV.MODIFY.TN-RANGE.UNDOCANPEND.SubscriptionVersion

<b>Purpose</b>	To verify a SOA can handle the error condition for the modification of cancel-pending subscription versions using a TN range, and the subscription versions are not updated in response to a modify cancel-pending version scenario. 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 subscription version modification of cancel-pending status using a TN range.
<b>Prerequisites</b>	Two or more ‘cancel-pending’ subscription versions exist on the NPAC SMS Simulator.
<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, in order to update the subscriptionVersionStatus attribute.</li> <li>2. 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.

3.2.5 A2A.SOA.VAL.MODIFY.ASSOCSP.UNDOCANPEND.SubscriptionVersion

<b>Purpose</b>	To test that a SOA, for an associated service provider, can modify a cancel-pending 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 modify of cancel-pending status.
<b>Prerequisites</b>	A cancel-pending 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 the status from cancel-pending back to pending when 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.</li> <li>3. The NPAC SMS Simulator sends a subscriptionVersionStatusAttributeValueChange or subscriptionVersionRangeStatusAttributeValueChange.</li> <li>4. The SOA, for an associated service provider, handles the notification sent by the NPAC SMS Simulator for the cancel-pending status, and responds with confirmation.</li> </ol>
<b>Expected Results</b>	The SOA, acting for an associated service provider, successfully initiates the subscriptionVersionModify M-ACTION and handles the subsequent interactions with the NPAC SMS Simulator.

3.2.6 A2A.SOA.INV.MODIFY.ASSOCSP.UNDOCANPEND.SubscriptionVersion

<b>Purpose</b>	To verify a SOA can handle the error condition for the modification of a cancel-pending subscription, for an associated service provider, and the subscription version is not updated in response to a modify cancel-pending version scenario. 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 of cancel-pending status.
<b>Prerequisites</b>	A cancel-pending 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 a cancel-pending status back to pending when 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. NPAC SMS Simulator responds with error status ‘failed’.</li> </ol>
<b>Expected Results</b>	The SOA, for an associated service provider, will correctly handle the error response received from the NPAC SMS Simulator.

### 3.3 **NANC 299**

#### 3.3.1 A2A.NPAC.INV.HEART.NO.RESP.InpNPAC-SMS

<b>Purpose</b>	Verifies the SOA/LSMS capability to correctly handle an abort, when no response to an InpNPAC-SMS MO class M-EVENT-REPORT request for the Heartbeat Notification is received by the NPAC SMS Simulator.
<b>Severity</b>	C
<b>Severity Explanation</b>	Required if the SOA/LSMS is supporting the Heartbeat Notification.
<b>Prerequisites</b>	An InpNPAC-SMS instance has been inherently created on the NPAC SMS.
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. NPAC sends a Heartbeat M-EVENT-REPORT request for InpNPAC-SMS (Heartbeat Notification).</li> <li>2. SOA/LSMS does NOT respond.</li> <li>3. After waiting for the timeout period, NPAC aborts the association.</li> <li>4. SOA/LSMS proceeds to re-establish an association with the NPAC SMS Simulator.</li> </ol>
<b>Expected Results</b>	Lack of Heartbeat confirmation from the SOA/LSMS causes abort. SOA/LSMS re-establishes association.

### 3.4 **NANC 352**

#### 3.4.1 A2A.SOA.VAL.MISC.ACTION.SWIM.resync

Perform NANC 351, TC 3.1.1, for Service Provider data, when SOA SP Data Indicator is set to TRUE.

#### 3.4.2 A2A.SOA.VAL.MISC.ACTION.SWIM.ASSOCSP.resync

Perform NANC 351, TC 3.1.2, for Service Provider data, when SOA SP Data Indicator is set to TRUE.

#### 3.4.3 A2A.LSMS.VAL.MISC.ACTION.SWIM.resync

Perform NANC 351, TC 3.1.3, for Service Provider data, when LSMS SP Data Indicator is set to TRUE.

### 3.5 **NANC 151**

Perform the following Test Cases for TN attribute when SOA TN Attribute Indicator is set to TRUE. Each test should verify that the TN attribute is provided.

#### 3.5.1 ~~A2A.OSOA.VAL.NOCONC.ACTIVATE.SubscriptionVersion~~

~~Existing Test Case, verify that the TN attribute is provided in the notification.~~

**3.5.2 ~~A2A.OSOA.VAL.NOCONC.NOACTIVATE.SubscriptionVersion~~**

~~Existing Test Case, verify that the TN attribute is provided in the notification.~~

**3.5.3 A2A.OSOA.VAL.CREATE.CONFLICT.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.4 A2A.DSOA.VAL.PORT-TO-ORIG.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.5 ~~A2A.NSOA.INV.MISS.INITIAL.CONC.SubscriptionVersion~~**

~~Existing Test Case, verify that the TN attribute is provided in the notification.~~

**3.5.6 A2A.NSOA.INV.STATE-TRANS.PEND-ACTIVE.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.7 A2A.NSOA.INV.STATE-TRANS.PEND-OLD.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.8 A2A.OSOA.INV.STATE-TRANS.PEND-OLD.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.9 A2A.OSOA.INV.STATE-TRANS.PEND-FAILED.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.10 A2A.DONORSOA.VAL.PORT-TO-ORIG.PTOLISP.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.11 A2A.SOA.VAL.PORT-TO-ORIG.ASSOCSP.PTOLISP.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.12 A2A.NSOA.VAL.ACTIVATE.BYNPAC.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.13 A2A.NSOA.VAL.ACTIVATE.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.



**3.5.14 A2A.NSOA.VAL.ACTIVATE.FAIL.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.15 A2A.NSOA.VAL.ACTIVATE.PARTFAIL.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.16 A2A.OSOA.VAL.ACTIVATE.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.17 A2A.OSOA.VAL.ACTIVATE.FAIL.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.18 A2A.OSOA.VAL.ACTIVATE.PARTFAIL.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.19 A2A.NSOA.ACTIVATE.ACTNOTMISS.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.20 A2A.NSOA.INV.ACTIVATE.PARTFAIL.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.21 A2A.OSOA.INV.ACTIVATE.PARTFAIL.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.22 A2A.NSOA.VAL.ACTIVATE.TN-RANGE.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.23 A2A.NSOA.VAL.MODIFY.PEND.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.24 A2A.OSOA.VAL.MODIFY.PEND.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.25 A2A.SOA.VAL.MODIFY.ACTIVE.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.26 A2A.SOA.VAL.MODIFY.ACTIVE.TN-RANGE.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.27 A2A.SOA.VAL.MODIFY.BYNPAC.ACTIVE.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.28 A2A.SOA.VAL.MODIFY.PARTFAIL.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.29 A2A.SOA.VAL.MODIFY.FAIL.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.30 A2A.SOA.INV.MODIFY.PARTFAIL.NOSPLIST.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.31 A2A.SOA.INV.MODIFY.ACTIVE.NOTMISS.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.32 A2A.SOA.INV.MODIFY.ATTRSAME.NOTMISS.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.33 A2A.SOA.VAL.MODIFY.PEND.TN-RANGE.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.34 A2A.SOA.VAL.MODIFY.DISCONPEND.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.35 A2A.SOA.VAL.MODIFY.TN-RANGE.DISCONPEND.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.36 A2A.SOA.VAL.MODIFY.ASSOCSP.DISCONPEND.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.37 A2A.SOA.VAL.CANCEL.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.38 A2A.NSOA.VAL.CANCEL.BYOSOA.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.39 A2A.NSOA.VAL.CANCEL.TN-RANGE.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.40 A2A.OSOA.VAL.CANCEL.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.41 A2A.OSOA.VAL.CANCEL.BYNSOA.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.42 A2A.OSOA.VAL.CANCEL.TN-RANGE.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.43 A2A.OSOA.VAL.CANCEL.NOCONC.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.44 A2A.NSOA.VAL.CANCEL.BYNPAC.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.45 A2A.OSOA.VAL.CANCEL.BYNPAC.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.46 A2A.NSOA.VAL.CANCEL.ACKREQ.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.47 A2A.OSOA.VAL.CANCEL.ACKREQ.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.48 A2A.NSOA.INV.CANCEL.CONFLICT.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.49 A2A.NSOA.VAL.CANCEL.CANCELED.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.50 A2A.OSOA.VAL.CANCEL.CONFLICT.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.51 A2A.NSOA.INV.CANCEL.PEND.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.52 A2A.OSOA.INV.CANCEL.CONFLICT.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.53 A2A.NSOA.INV.CANCEL.ACTIVE.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.54 A2A.SOA.VAL.IMMDISC.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.55 A2A.SOA.VAL.DEFDISC.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.56 A2A.SOA.VAL.IMMDISC.BYNPAC.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.57 A2A.SOA.VAL.IMMDISC.FAIL.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.58 A2A.SOA.VAL.IMMDISC.PARTFAIL.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.59 A2A.SOA.VAL.IMMDISC.TN-RANGE.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.60 A2A.SOA.INV.IMMDISC.ACT.OLD.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

**3.5.61 A2A.SOA.INV.IMMDISC.OLD.SubscriptionVersion**

Existing Test Case, verify that the TN attribute is provided in the notification.

### 3.5.62 A2A.SOA.INV.IMMDISC.FAILED.SubscriptionVersion

Existing Test Case, verify that the TN attribute is provided in the notification.

### 3.5.63 A2A.SOA.INV.IMMDISC.OLD.FAILService Provider.SubscriptionVersion

Existing Test Case, verify that the TN attribute is provided in the notification.

### 3.5.64 A2A.SOA.VAL.CANCEL.DISCPEND.SubscriptionVersion

Existing Test Case, verify that the TN attribute is provided in the notification.

### 3.5.65 A2A.NSOA.VAL.CONFLICT.RESOLV.SubscriptionVersion

Existing Test Case, verify that the TN attribute is provided in the notification.

### 3.5.66 A2A.NSOA.VAL.CONFLICT.RESOLV.BYNSOA.SubscriptionVersion

Existing Test Case, verify that the TN attribute is provided in the notification.

### 3.5.67 A2A.OSOA.VAL.CONFLICT.RESOLV.SubscriptionVersion

Existing Test Case, verify that the TN attribute is provided in the notification.

### 3.5.68 A2A.OSOA.VAL.CONFLICT.RESOLV.BYOSOA.SubscriptionVersion

Existing Test Case, verify that the TN attribute is provided in the notification.

### 3.5.69 A2A.NSOA.VAL.CONFLICT.RESOLV.TN- RANGE.BYNSOA.SubscriptionVersion

Existing Test Case, verify that the TN attribute is provided in the notification.

## 3.6 **NANC 357**

Perform the following Test Cases for SP Type attribute when SOA SP Type Attribute Indicator is set to TRUE. Each test should verify that the SP Type attribute is provided.

### 3.6.1 A2A.SOA.CAP.OP.SET.ASSOCSP.serviceProv

Existing Test Case, verify that the SP Type attribute is provided in the M-SET Request.

### 3.6.2 A2A.SOA.CAP.OP.GET.ASSOCSP.serviceProv

Existing Test Case, verify that the SP Type attribute is provided in the M-ACTION Response.

## 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

